Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

From Twitter to detector: Real-time traffic incident detection using social media data

Yiming Gu [a], Zhen (Sean) Qian [a,*], Feng Chen [b]

[a] Department of Civil and Environmental Engineering and Heinz College, Carnegie Mellon University, Pittsburgh, PA 15213, United States
[b] Department of Computer Science, University of Albany SUNY, Albany, NY 12222, United States

ARTICLE INFO

Article history:
Received 2 July 2015
Received in revised form 18 February 2016
Accepted 21 February 2016
Available online 17 March 2016

Keywords:
Incident detection
Social media
Natural language processing
Geocoding
Data mining
Crowd-sourcing

ABSTRACT

The effectiveness of traditional incident detection is often limited by sparse sensor coverage, and reporting incidents to emergency response systems is labor-intensive. We propose to mine tweet texts to extract incident information on both highways and arterials as an efficient and cost-effective alternative to existing data sources. This paper presents a methodology to crawl, process and filter tweets that are accessible by the public for free. Tweets are acquired from Twitter using the REST API in real time. The process of adaptive data acquisition establishes a dictionary of important keywords and their combinations that can imply traffic incidents (TI). A tweet is then mapped into a high dimensional binary vector in a feature space formed by the dictionary, and classified into either TI related or not. All the TI tweets are then geocoded to determine their locations, and further classified into one of the five incident categories.

We apply the methodology in two regions, the Pittsburgh and Philadelphia Metropolitan Areas. Overall, mining tweets holds great potentials to complement existing traffic incident data in a very cheap way. A small sample of tweets acquired from the Twitter API cover most of the incidents reported in the existing data set, and additional incidents can be identified through analyzing tweets text. Twitter also provides ample additional information with a reasonable coverage on arterials. A tweet that is related to TI and geocodable accounts for approximately 5% of all the acquired tweets. Of those geocodable TI tweets, 60–70% are posted by influential users (IU), namely public Twitter accounts mostly owned by public agencies and media, while the rest is contributed by individual users. There is more incident information provided by Twitter on weekends than on weekdays. Within the same day, both individuals and IUs tend to report incidents more frequently during the day time than at night, especially during traffic peak hours. Individual tweets are more likely to report incidents near the center of a city, and the volume of information significantly decays outwards from the center.

1. Introduction

Traffic congestion is one big challenge for both travelers and infrastructure managers. It can be generally categorized into recurrent and non-recurrent congestion. Commuters' behaviors are naturally a day-to-day repetitive choice, which leads to recurrent flow patterns. On the other hand, non-recurrent congestion is induced by non-recurring causes, such as traffic

* Corresponding author.
E-mail address: seanqian@cmu.edu (Zhen (Sean) Qian).

http://dx.doi.org/10.1016/j.trc.2016.02.011

accidents, work zones, adverse weather events, and special events, which takes about half of the total congestion (Systematics, 2005). It is critical to detect those non-recurrent causes, namely incidents, in an efficient and timely manner, so that traffic managers can apply management strategies to mitigate the non-recurrent congestion. This research proposes an efficient, inexpensive and ubiquitous way to detect incidents in real time by leveraging the power of social media data.

For decades, research has been dedicated into establishing traffic incident detection systems to identify the time, locations, and types of traffic incidents in real time. It would be ideal to have human beings to report all incidents manually since human beings can provide detailed and accurate information regarding incidents. However, due to high capital/labor cost and significant delay in human-based reports, algorithms have been developed to automatically detect incidents. Implicitly embedded in the detection automation is the assumption that significant change in flow characteristics immediately following the incidents. Through mining the real-time traffic data collected by scattered sensors in transportation networks, incidents and their features may be identified. Algorithmic incident detection is, however, still not cheap. Incidents may occur in any location and any time period, and thus to achieve reasonable coverage and accuracy, sensing traffic flow in a wide spectrum of time and space is necessary. More importantly, algorithmic incident detection tends to work well on highways, but not on local arterials. The traffic flow on arterials is largely affected by random factors, such as non-motorized traffic, signal lights, and street parking. Given the current sensing coverage, it is notoriously difficult to accurately detect arterial incidents. Our general motivation is to discover an efficient yet cost-effective way to detect incidents on both highways and arterials in real time. Crowdsourcing seems a solution to this matter for its low cost, real-time capacity and reasonable accuracy.

WAZE (www.waze.com) is a good example of incidents crowdsourcing. It has successfully provided real-time incident information in the web-GIS based applications. However, its limitation is twofold. First, reporting incidents to WAZE requires logging in particular applications and creating reports that fall in one of the pre-determined incident categories. It is unknown that how the self-selection and penetration rate would affect the coverage and accuracy of reported incidents. It is possible that the majority of the active internet users do not limit themselves to a specific crowdsourcing application (such as WAZE). Would public data in the social media lead to a better coverage on incidents? Second, WAZE' data are proprietary and owned by private sector. It may be difficult for public agencies to obtain those data. Is there an inexpensive way to crowdsource incidents for the public agencies to use?.

Social media sites sharing short messages, such as Twitter, have become a powerful and inexpensive tool for extracting information of all kinds. It has a fairly large user pool, much more diverse than a specific incident crowdsourcing tool (such as WAZE). Also a significant portion of its data is shared by individuals to the public, which can be acquired using APIs. Twitter currently produces 340 million tweets per day from more than 140 million active users. Since transportation is part of everyone's daily lives, many active users post messages when they encounter incidents, or shortly after. This huge resource may potentially gather a valuable body of information regarding incidents that differ significantly by type, location, and time. Social media sites may be an inexpensive alternative to privately-owned crowdsourcing tool (such as WAZE). Numerous research has been devoted into mining social media data to detect events and breaking news. Current event detection methods based on social media streams can be classified into three categories: clustering-based, model-based, and signal-processing-based, see Bontcheva and Rout (2012) for an overview. A well-known signal-processing-based approach has shown that Twitter can be used to detect special events much faster than the traditional media (Sakaki et al., 2010). An in-depth analysis of how breaking news spread on Twitter is provided in Hu et al. (2012).

Nevertheless, social media data does not come without a price. The real-time detection of incidents based on Twitter is challenging. The state-of-the-art text mining techniques cannot be applied directly to mine tweets since the tweet language varies considerably from daily language. Twitter messages are short (140 characters at most) and can often contain typos, grammatical errors, and cryptic abbreviations. In 2009, a short-term study stated that 40% of the tweets can be considered as "pointless babble" (Analytics, 2009), making it difficult to separate useful information from plain noise.

This paper presents a methodology to crawl, process and filter public tweets in the real time. Those tweets are then analyzed to extract incident information through a simple procedure using Natural Language Processing techniques. We apply the methodology in two regions, Pittsburgh and Phildelphia metropolitan areas, to extensively examine both temporal and spatial coverage of detected incidents. The Twitter-based incidents are then compared to the real-time data currently being used by Pennsylvania Department of Transportation Traffic Management Centers (TMCs). Many incidents on the roadway are neither reported to TMCs nor 911, or reporting is significantly delayed because data has to flow from external systems (such as 911 system) to TMCs. We demonstrate that mining social media data (using Twitter as an example) holds great potentials to complement real-time incident reporting sources in a very cost-effective way.

The methodology presented in this paper is applicable for real-time incident detection. However, in order to fully examine the accuracy, volume and timeliness of incident information, we compare historical social media data and existing incident reports in the two regions for this study. One limitation is that due to lack of ground truth on the incident occurrence time(not reported time), we are unable to make definitive conclusions on its timeliness at this time. The timeliness can be fully evaluated in the field by emergency response officers.

This paper is organized as follows. We first review incident detection literature in Section 2 and discuss the benefits of mining tweets in details in Section 3. Section 4 describes the methodology of processing and analyzing tweets data, which is then applied in two case studies in Section 5. Conclusions are drawn and discussed in Section 6.

2. Literature review and Twitter-based incident detection

2.1. Literature review on traditional algorithms of incident detection methods

The core of a traditional incident detector consists of two components: data acquisition and data analytics. The traffic data fed to an incident detector can be classified into two categories:

(C1) Flow measurements at a fixed location: counts, occupancy and speeds measured by inductive loop sensors, magnetic sensors, microwave sensors, infra-red sensors, ultrasonic sensors, acoustic sensors, laser sensors, video image processors, etc.
(C2) Probe vehicle data: sampled vehicle trajectories measured by global position systems, cellular geolocation systems, and automatic vehicle identification.

C1 has been widely deployed among major cities and highways, and therefore this form of data has relatively high level of reliability. The drawback is that C1 only provides the information of traffic flow at sparse locations in the network. C2 data is limited by its low sampling rate, and sometimes may exhibit high measurement errors. Comparing to algorithms using C1 data, methods using C2 data consider the spatio-temporal properties of probe vehicles. Results showed that 2–3% penetration of location-enabled vehicles may be sufficient for monitoring traffic conditions (Herrera et al., 2010).

Dependent on the data acquisition system, different incident detection algorithms have been developed. Pattern-matching algorithms identify the traffic flow characteristics under traffic incidents using C1 data, mostly for highways only (Stephanedes and Chassiakos, 1993). Efforts on pattern matching algorithms can be traced back to "California Method" (Thancanamootoo and Bell, 1988). Its assumption is that the occurrence of an incident reduces both upstream and downstream flow rate and downstream occupancy, and oftentimes increases upstream occupancy. Studies conclude that this type of algorithm is only accurate when the incident occurs near those fixed detectors (Tigor and Payne, 1977; Payne and Tignor, 1978). The main drawback of this method is that it is difficult to set up a "threshold" (e.g., the difference of occupancy rate between upstream and downstream detectors) which classifies incident and non-incident flow patterns.

Some researchers turned to statistical discriminative algorithms to detect incidents (e.g., Dudek et al., 1974; Tsai and Case, 1979). This type of algorithm computes the Standard Normal Deviate (SND) of traffic flow to trace the sudden change induced by incidents. Modern statistical discriminative models include linear classification (Sethi et al., 1995) and Support Vector Machine (Yuan and Cheu, 2003). Sethi et al. (1995) used simulated travel time and occupancy data under different representative traffic incidents from the INTRAS simulation model. Khan and Ritchie (1998) and Ritchie and Cheu (1993) used Artificial Neural Network (ANN) to classify traffic patterns into "with incidents" and without. They collected the Inductive loop detector data and incident data from the network of Disnay Land, Anaheim Convention Center.

Time series algorithms such as Kalman Filter (Willsky et al., 1980), Sequential Probability Ratio Test (SPRT) (Abdulhai and Ritchie, 1999), autoregressive integrated moving-average (ARIMA) model (Ahmed and Cook, 1977; Ahmed and Cook, 1980; Ahmed and Cook, 1982), high occupancy (HIOCC) algorithm (Collins and Martin, 1979), wavelet models (Teng and Qi, 2003) and clustering (Anbaroglu et al., 2014) have been used to detect incidents by tracking flow and occupancy data in real time. For SPRT, Abdulhai and Ritchie (1999) proposed that the traffic flow can be treated as time-series signals and traffic accidents can be detected by comparing the likelihood ratio of the signal data before and after the accident. SPRT targets a specific false-positive rate. Teng and Qi (2003) used wavelet method directly to develop an incident-sensitive features in the wavelet space, and indicated a better result than Artificial Neural Network and the California Method.

While incident detection algorithms on highways is mature, arterial incidents are notoriously difficult to detect in real time due to various disruptions to arterial flow (Sermons and Koppelman, 1996; Ahmed and Hawas, 2015). Bayesian Network (BN) has been implemented in arterial traffic incident detection by incorporating both prior knowledge and monitored data (Zhang and Taylor, 2006). Simulated occupancy and flow data have been used to feed a Bayesian Network where the causation relationship was pre-determined by experts.

Recently individuals trajectories data (collected through GPS) have been used to directly infer incidents (Park and Haghani, 2015). A multi-level approach was developed by Kamran and Haas (2007), where a hierarchical analysis of the vehicles' GPS data was conducted to identify precise locations of incidents. Moreover, an mobile application, called "WreckWatch" (White et al., 2011), was developed for GPS-enabled smartphones. In terms of cellular-based incident detection, exploratory study has been done by Demissie et al. (2013).

2.2. A new approach: Twitter-based incident detection

Compared to C1 and C2, Twitter data has a relatively high true positive rate and relatively high sampling rate, but the texts are, in most circumstances, obscure and non-standardized. Therefore natural language processing (NLP) is required before using such data. There are very few studies utilizing NLP to mine text-based data for analyzing incidents. Xiang and Gretzel (2010) discussed the usage of social media in collecting travel information. Pereira et al. (2013) incorporated Topic Models to estimate the duration of incidents from two years of incident reports. Recently, Schulz et al. (2013), Sasaki et al. (2012), and Krstajic et al. (2012) focused on detecting small-scale events such as a specific type of incident.

Yates and Paquette (2011) and Gao et al. (2011) explored the possibility of building a knowledge manage system based in social media data in crisis management.

Comparing to traditional methods, the new Twitter-based method has the following advantages: (1) The cost of acquiring public tweets is minimal. The Twitter APIs can be called for free of charge. (2) The incident report from natural language, such as tweets, can be very specific about the time, location, and presence of an incident, comparing traditional automatic incident detection relying on flow data monitoring. (3) Processing natural language is efficient and the algorithms are scalable. (4) Twitter-based or crowdsourcing incident detection can report incidents in a very timely manner. (5) The methodology of tweets-based incident detection can be extended and applied to extracting other types of traffic information.

Adler et al. (2015) reported the trend that crowdsourced data from social media is being implemented and enhancing Traffic Management Center (TMC) operations. Also, Lee et al. (2015) explored the possibility to use Twitter data on validating travel demands. Moreover, Fu et al. (2015) mined tweets from four Twitter users in the Washington D.C. region, namely "WTOPtraffic", "VaDOT", "drgridlock" and 'DCPoliceDep'. Those four accounts are created by public agencies disseminating incident info to the public travelers. This study identified incidents by matching pre-determined key words, and verified that they are in the incident database. However, what is unknown is that what incident information can the tweets from general users (individuals and other public accounts) provide? Could we use tweets to identify incidents that are not currently reported to TMC?

In this paper, we propose a methodology to crawl, process and filter tweets that are posted by generic users and shared with the public. Because public tweets exhibit special linguistic features, such as acronym and short words, we will use sophisticated natural language process (NLP) algorithms to dynamically learn the Twitter language and identify geo-locations and incidents features. We also study the efficiency and effectiveness of Twitter-based incident detection. In particular, this paper addresses the following questions.

- Is there a way to identify tweets that report incidents? If so, what is the type of the incident being reported?
- How can we extract geo-location information by mining tweets' texts?
- Which type of Twitter users contribute the most incident-related information, individual users or influential users (namely public Twitter accounts)?
- Are there any particular time periods or areas where people tend to tweet more often to report incidents?
- Can social media enhance, or even replace traditional traffic incident report systems?

3. Tweets and their relevance to traffic incidents

There are over 400 million tweets posted everyday on Twitter all over the world, with over 100 million daily active users. It is a huge resource for people to share information they observe. Tweets are essentially short messages, limited to 140 characters. Twitter users consist of authoritative users (such as public agencies) and individuals.

Individual users can tweet to share firsthand (original) information regarding incidents. Some examples are,

- "95 SB crawling near Betsy Ross Bridge" with a picture attached.
- "per traffic, 676 is not the blue route. Accident is at Vine St Expy and the Schkuykill Expy".
- "Fast food workers arrested as union-backed protest blocks traffic in front of Penn Ave McDonald's".
- "Whoa! Bad accident at Tulip & Unruh NorthEast Philly – Courtesy of Tacony Town Watch".
- "ramp off 376 while they had to close the Carneige exit. At least that could spare some traffic!".
- "Avoid 76 west, approaching Vare from Walt, apparent ax, traffic is nuts, Ben is also backed up".

Some tweets with the original information may be sufficient to infer the time and location of a traffic incident. However it may be difficult to extract that information with an automatic process, because the linguistic expression of incidents can vary significantly among different users. For example, there are numerous ways to discuss "traffic congestion", such as "bumper-to-bumper traffic", "stuck there for 10 min", and "a similar situation Harry experienced yesterday". Most of the tweets language are very personalized. In addition, accurate location information of incidents, as part of tweets' attributes, is usually not available, but it may be inferred by mining tweets' texts. For privacy concerns and by default settings of iOS/Android systems, Global Position System is disabled and almost all tweets are posted without latitude/longitude information (99.9% of tweets we obtained from Twitter APIs for our case study). However, tweets' texts may contain partial location information. The expression of locations in tweets is highly irregular, such as "on Forbes in front of Hamburgh hall" or "halfway to Phily". It is nearly impossible for an individual to text locations in full details using smartphones in a short time period. Ways of solving these issues in extracting location information will be discussed in the following sections.

News and public agencies (namely, influential users or IU for short) also tweet to disseminate incident information. Examples of typical Influential Users and their tweets are "Turnpike Roadwork on Pennsylvania Turnpike I-476 northbound between Exit 31 – PA 63 and Exit 44 – PA 663 affecting the right lane" by Pennsylvania Turnpike Twitter account, and "Water Main Break in the Strip District on 28th Street between Railroad St & Smallman St" by the City of Pittsburgh. These tweets are highly standardized, with much detail, and therefore easy to mine and geocode. This type of tweets serves as a critical source for incident data.

From now on, we call tweets that are related to traffic incidents “TI tweets”, and those unrelated to traffic incidents “NTI tweets”. Specifically, a TI tweet is one contains information about a traffic incident, implies abnormality on the transportation infrastructure, or indicates a (potentially) significant disruption to the traffic. For example, “the roads are really slippery to drive on” is a TI tweet, because it contains the information (being slippery) about the current traffic infrastructure (the roads). Tweets describing recurrent traffic do not count as TI tweets. For instance, “Parkway East WB slow traffic” posted at 5:25 pm (namely the afternoon peak hours) indicating recurrent peak hours. Some tweets are used as an outlet for expressing individual emotions. For instance, “I’m so sick of crazy bike riders!” or “If you drive so slow, why get on a highway!!!”. Clearly they are classified as NTI tweets. We carefully examine tweets in the training set by manually labeling tweets that are related to the traffic incidents only. Note that useful tweets sharing firsthand information can be forwarded by followers. Texts of re-tweets in addition to firsthand tweets are not mined in this study since they simply resemble the same incident information.

4. Methodology: Twitter data acquisition, processing and analytics

We develop a methodology to translate tweets into traffic incident information. Fig. 1 illustrates the procedure conceptually. We conduct the initial data acquisition from Twitter servers, followed by an iterative process, namely the adaptive data acquisition where we crawl tweets and establish our dictionary of “words” relevant to traffic incidents. The process of adaptive data acquisition also selects the most important features to form a feature space that is informative and non-redundant. Next, tweets are mapped into this feature space and classified by a well-trained Semi-Naive-Bayes (SNB) classifier as either TI or NTI tweets. All the TI tweets are then pushed through a geo-parser and geocoder, to determine their locations. In addition, each TI tweet is further classified by a trained Supervised Latent Dirichlet Allocation classifier to identify the category of the incident it reports.

4.1. Twitter APIs and initial data crawling

Twitter offers two types of crawling APIs that allow users to query tweets by keywords, user IDs, and time/date. The first type, REST APIs, allows users to submit queries to retrieve recent or popular tweets. The REST query format includes a centroid (latitude, longitude), a radius (e.g., 0.5 mile), and a set of keywords with the support of operators, including AND, OR, and EXCLUDE (e.g., “traffic AND (accident OR collision)”). Tweets data consist of user information, text, time posted, times of re-tweets, latitude and longitude (if any), and the referred URL for figures/videos. Notice that the location data is only available for a small portion of collected tweets (less than 0.1% in our case studies) and therefore texts are used as a main source for textual ming and geocoding. User information includes userID, nationality, residence city, and the number of following/-followed users. IUs can be identified as those Twitter accounts that are created by public agencies or media for disseminating traffic information. In this study, IUs are collected manually by Google searching Twitter accounts of major news channels and public agencies.

Twitter REST APIs are 100% free of charge but has certain limitations. For example, API calls are limited to 350 queries every 15 min for one user account, or 3500 total tweets per REST query, whichever comes more restrictive. To fully utilize all the REST API calls, the developers use an OAuth 2.0 protocol to access data. Twitter also encourages third-party libraries for various programming languages including Python, PHP, Java, and Javascript. A popular Java library, Twitter4J, is used in this research.

The second type, Twitter streaming APIs, requires users to keep an uninterrupted HTTP connection to access the most recent (mostly within 1 day) Twitter data up to 1 percent of public tweets. The streaming query format includes a combination of a centroid (latitude, longitude) and a radius (e.g., 0.5 mile) or a set of keywords with operations similar to REST APIs. However, this APIs do not support the joint query of locations and keywords.

For the purpose of establishing a machine learning model based on historical tweets, it is clear that REST APIs are the most effective ways since it allows us to query with locations and keywords simultaneously. In addition, it is also critical to select

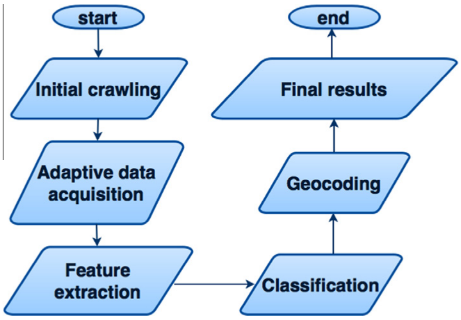


Fig. 1. The work flow of Twitter data acquisition, processing and analytics.

trustworthy IUs to access their timelines and historical tweets. To obtain incident information from tweets to the fullest extent, it is essential to develop a combination of keywords that leads to the best recall and reasonable precision. Recall and precision as basic scores of the data acquisition system, defined as follows:

$$Recall = \frac{A \cap B}{A} \tag{1}$$

$$Precision = \frac{A \cap B}{B} \tag{2}$$

where A is the set of all TI tweets in a time period, and B is the set of all acquired tweets in the same time period. The ideal goal is to achieve as much precision and recall as possible simultaneously, which means all acquired tweets are only TI tweets and all TI tweets in the pool are acquired.

However, it is usually impossible to achieve a 100% recall, or to precisely estimate the recall, given the limitations on the total number of tweets accessible through REST API. And there is hardly any ground truth of a full set of TI tweets. However, it can be estimated using the ratio of E over Q where Q is denoted by the number of TI tweets of all the tweets of randomly selected test users from Twitter, and Q is denoted by the number of tweets crawled based on REST API (with a specific set of query parameters) intersecting those test users' tweets. A 100% precision is also hard to achieve because there always exist some tweets matching query keywords but are not related to traffic incidents. Therefore, the goal of the data acquisition process is to achieve reasonably high recall and precision, such that we obtain as many TI tweets as possible through free-of-charge Twitter APIs.

4.2. Adaptive data acquisition

The REST APIs require a list of keywords to perform queries. We can come up with an initial set of keywords, namely an initial dictionary. However, the recall and precision is usually low because those keywords do not cover all the language indicating incidents. To ensure the best quality and maximum quantity of TI tweets that could be possibly acquired, we need to expand the dictionary to retrieve more TI tweets. One strategy is to learn from all words of the queried tweets that are not part of the dictionary and select those words relevant to incidents. We can then add those new words to the dictionary and perform new queries to obtain more TI tweets in a new iteration. This is also known as the adaptive data acquisition as this strategy of data acquisition can be implemented over time to adapt to the most recent Twitter language. In each iteration, we will manually label TI tweets, and count the frequencies of new words. It is believed that the more frequent a word (or a combination of words) is used in all TI tweets, the more likely it contains incident and geo-location information. The process of adaptive data acquisition is described in Fig. 2.

First, we collect initial words that are related to traffic incidents, namely "seed words", to assemble initial queries. The seed dictionary includes, but are not limited to, "traffic, accident, road, avenue, car, bike, truck, driver, injured, congestion, slow, I-, PA-, US-, exit, mile, stop, -crazy, -hate, -! ". The "-" symbol leading a word implies the exclusion of this word in a query. Note that a query with keyword "traffic" pulls out tweets containing "traffic", while a query with keywords "traffic -hate" extracts tweets that contain the word "traffic" and do not contain "hate". The dictionary of the initial "seed words" is notated as S_0.

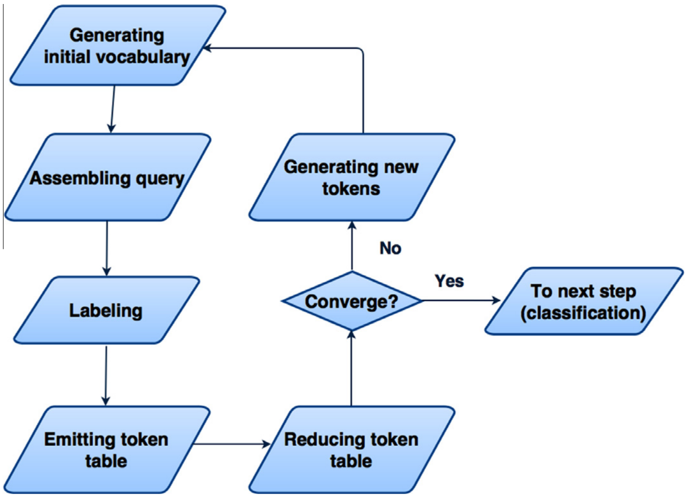


Fig. 2. Flow chart of adaptive data acquisition.

For the v-th iteration, with a dictionary S_v, an expansion of the dictionary is made to include the first two synonyms of each of the words in S_v. The synonyms are extracted from the open-source WordNet database. Therefore we have an expanded dictionary:

$$S'_v = E(S_v) \tag{3}$$

where E is the expansion operator and also notice that S'_v is a superset of S_v:

$$S'_v \supseteq S_v \tag{4}$$

In addition, a batch of queries are constructed by using “OR” criteria to contain each word and combinations of any two, three and four words from all the words in S'_v:

$$Q_v = C(S'_v) \tag{5}$$

where Q_v is the queries constructed from the dictionary S'_v with the operator C. The size of Q_v increases exponentially with respect to the size of S'_v. Note that the REST API here pulls out tweets randomly from the entire tweets pool with a total limit. Therefore, a query with one keyword “A” does not necessarily pulls out all tweets with the keyword. For the same reason, the resultant tweet set is not necessarily a superset of what we obtain from a query with two keyword “A” and “B”. Thus, the queries consisting of combinations of two, three and four words from the dictionary can acquire as many tweets as possible with a reasonable acquisition speed. The set of tweets acquired are denoted by,

$$W_v = F(Q_v) \tag{6}$$

where F is the operator where we send queries to REST API and receive a set of tweets.

From all the acquired tweets (W_v), we manually label the TI tweets and NTI tweets. a TI tweet is one contains information about a traffic incident, implies abnormality on the transportation infrastructure, or indicates a (potentially) significant disruption to the traffic. For example, two tweets are labeled in Table 1.

Next the tweets are processed through a specially designed tokenizer to retrieve incident information. The tokenizer sees each word of nouns, verbs, adverbs and adjectives as a token, and removes articles, prepositions, the verb “be” and other words that does not carry physical meanings. It also uses a dictionary of road names to identify tokens of road names. For example, the tokenizer will transform the word “I-376” to the token “*road-name*”. Regardless of what roads tweets imply, the road name and number are likely to indicate an incident. Using the example in Table 1, the tokenized tweet can be expressed as: [”cleared:” “multi” “vehicle” “accident” “*road-name*” “eastbound” “mile” “post:” “73.0.”]. Note that prepositions, such as “on” and “at” are neglected by the tokenizer.

Following the Map-Reduce framework, the combinations of tokens with the labels of the tweets are emitted. Table 2 explain the emission using the same example. Then, a “reducer” is applied to the entire acquired tweets T_0 to count the total number of positive and negative labels for each of the token combinations. Examples are given in Table 3. For instance, the token combination, “cleared:” and “multi”, has 12 positive counts and one negative counts, indicating that there are in all 13 occurrences of [”cleared:” “multi”] in the same tweets in this batch of queries, and 12 of them are TI tweets and 1 of them is an NTI tweet. After building this table, we can identify those token combinations with the maximum positive counts (namely with the greatest positive correlation with TI tweets) and the maximum negative counts (with the greatest negative correlation with TI tweets).

To limit the number of new words added to the dictionary in the next iteration, a total of K new token combinations are chosen. We set the chosen number of new positively (negatively) correlated token combinations proportional to the percentage of TI tweets (NTI tweets) in the all acquired tweets, denoted by N_p and N_n respectively,

$$N_p = K\frac{N_{TI}}{N} \tag{7}$$

$$N_n = K\frac{N_{NTI}}{N} \tag{8}$$

where N, N_{TI} and N_{NTI} are the number of total acquired tweets, TI tweets and NTI tweets respectively.

The final step of the v-th iteration is to add the chosen K new tokens into the dictionary,

$$S_{v+1} = S_v \cup W_v \tag{9}$$

where S_{v+1} is the set of words to assemble queries for the next iteration, and W_v is the set of K chosen tokens.

Table 1
An example of tweet labeling.

tweet.text	tweet.label
CLEARED: Multi vehicle accident on I-376 eastbound at Mile Post: 73.0.	1
Apple shows off iPad’s improved camera	−1

Table 2
An example of token emission.

Token combinations	Tweet label
cleared:	1
multi	1
.	.
.	.
“cleared:” “multi”	1
.	.
.	.
“cleared:” “multi” “vehicle”	1
.	.
.	.
“apple”	−1
.	.
.	.
“apple” “shows” “off” “ipad” “s” “improved” “camera”	−1

Table 3
An example of reducing labels.

Token combinations	Positive counts	Negative counts
cleared:	15	1
multi	12	4
.	.	.
.	.	.
“cleared:” “multi”	12	1
.	.	.
.	.	.
“cleared:” “multi” “vehicle”	4	0
.	.	.
.	.	.
“apple”	0	10
.	.	.
.	.	.
“apple” “shows” “off” “ipad” “s” “improved” “camera”	0	1

The iterations go on until we do not find new TI tweets or identifying new token is no longer cost-effective. The convergence criteria is therefore set to that the percentage of TI tweets in all newly acquired tweets in the v-th iteration, r_v, is very small, e.g., 1%. $r_v = 1\%$ indicates that we will need to label 100 tweets based on the pre-defined incident dictionary (roughly 30 min to 1 h in our experiments) in order to find one TI tweet, and the process is no longer cost-effective and should terminate.

Denote A_v the number of new tweets acquired in the v-th iteration, and K_v the number of tweets that are not yet acquired from the Twitter pool. Suppose r'_v is the percentage of TI tweets in those K_v tweets not yet acquired. In the $(v+1)$-th iteration, there are $K_v - A_v$ amount of new tweets left in the pool, and the percentage of TI tweets is,

$$r'_{v+1} = \frac{K_v r'_v - A_v r_v}{K_v - A_v} \tag{10}$$

Since we use a pre-define dictionary consisting of words related to incidents, the percentage of TI tweets in the new acquired tweets is generally greater than that of all tweets in the pool, namely $r_v > r'_v$. Therefore, r'_v decreases as the iteration goes on. When r'_v is sufficiently small, the percentage of TI tweets acquired in a new iteration can be smaller than a threshold (e.g., 1%) when the convergence criteria is met.

As a result of the adaptive data acquisition process, the dictionary contains single words and combinations of some words that are positively correlated with being a TI tweet, which serve as features. Those features are actually selected from all the words used in all acquired tweets, in order to form an informative feature space with minimal redundancy. We discuss how to project each tweet onto the feature space for the classifications.

4.3. Classification

4.3.1. Classification on TI/NTI tweets

As described in Section 3, tweets are acquired by two methods: queries on keywords and queries on IU accounts. Queries from both sources hold fundamentally different properties: queries on keywords need to ensure an adequate recall and therefore come with a low precision, whereas queries on IUs' timelines result in limited number of tweets, and have extremely high precision. Therefore, a filter is necessary to remove those acquired NTI tweets from the data set. We briefly describe the Semi-Naive-Bayes model adopted in this paper to filter out NTI tweets.

Suppose Y is the label whether or not a tweet is TI tweet, and X is the projection of a tweet onto a feature space, given by the vector $X = [X_1, X_2, X_3, \ldots, X_I]^T$, where I is the dimension of the feature space, namely the total number of words and word combinations in the dictionary that are positively correlated with incidents. To estimate the probability of a tweet X being a TI tweet, the Bayes formula is given by,

$$P(Y|X) = \frac{P(XY)}{P(X)} = \frac{P(Y)P(X|Y)}{P(X)} \propto P(Y)P(X|Y) \tag{11}$$

One issue is how to efficiently calculate the joint conditional probability $P(X|Y)$, because X is usually in very high dimensions. To simplify this calculation, Naive-Bayes makes a "Naive" assumption on the conditional independence of all features:

$$P(X|Y) = P(X_1|Y)P(X_2|Y)P(X_3|Y)\ldots P(X_k|Y) \tag{12}$$

However, conditional independence is not realistic in this study. For instance, suppose $X = [X_1, X_2]^T$ where X_1 is 1 if the word "bus" occurs, and 0 otherwise. X_2 is 1 if the word "stop" occurs and 0 otherwise. Then $P(X_1|Y = 1)$ and $P(X_2|Y = 1)$ is the probability of the occurrence of the word "work" and "zone" given a TI tweet, respectively. Clearly, $P(X_1X_2|Y) \neq P(X_1|Y)P(X_2|Y)$ because "work zone" is a commonly used term in a tweet to alert road closures, but not each of these two words alone. In other words, the occurrence of the words "work" and "zone" in a TI tweet is highly correlated.

The intention of using a Semi-Naive-Bayes is to take into account those correlated features whereas still holding a part of the "naive" assumption to avoid computation in high dimensions. The Semi-Naive-Bayes classification model differs from the Naive Bayes model by consolidating those correlated features,

$$P(X|Y) = \prod_{(i,j)\in\sigma} P(X_i, \ldots X_j|Y) \prod_{I}^{J} P(X_n|Y) \tag{13}$$

where all features are ordered by those correlated feature tuples first, followed by independent features. σ is the set of the positions in the order for the first and last feature in a correlated tuple, and the features with the position from J to I are all independent features. Fortunately, features have been selected in the adaptive data acquisition process with the consideration of word combinations. Note that those single words and word combinations with the highest frequencies are selected as part of the dictionary. Therefore, we can directly apply those words and combinations to form a feature space for the Semi-Naive-Bayes classification by assuming that each of those single words and word combinations can occur in TI tweets independently.

In the Semi-Bayes-Classifier, each probability term can be computed by,

$$P(X_i, \ldots X_j|Y) = \frac{\#\{X_i, \ldots X_j, Y\}}{\#\{Y\}} \tag{14}$$

where the notation $\#\{A\}$ means the number of label/word $\{A\}$ in the pool of all acquired tweets.

Given a feature vector X of a tweet, we classify this tweet by,

$$Y^* = \arg\max_Y P(Y|X) \tag{15}$$

where $Y^* = 1$ indicates this tweet is a TI tweet, and 0 otherwise.

For example, a tweet reads "Pkwy W delays begin before the top inbound, very slow outbound from Green Tree to work zone.", where we suppose the feature space is defined by ("pkwy","delay", "work zone", "crash"), then this tweet's coordinate in this feature space is (1,1,1,0). The posterior probability of Y is given by

$$\begin{aligned} P(Y = 1|X) \propto \frac{N_{TI}}{N} &\times \frac{\#\{\text{"}work + zone\text{"} = 1, Y = 1\}}{\#\{Y = 1\}} \times \frac{\#\{\text{"}pkwy\text{"} = 1, Y = 1\}}{\#\{Y = 1\}} \times \frac{\#\{\text{"}delay\text{"} = 1, Y = 1\}}{\#\{Y = 1\}} \\ &\times \frac{\#\{\text{"}crash\text{"} = 0, Y = 1\}}{\#\{Y = 1\}} \end{aligned} \tag{16}$$

$$\begin{aligned} P(Y = 0|X) \propto \frac{N_{NTI}}{N} &\times \frac{\#\{\text{"}work + zone\text{"} = 1, Y = 0\}}{\#\{Y = 0\}} \times \frac{\#\{\text{"}pkwy\text{"} = 1, Y = 0\}}{\#\{Y = 0\}} \times \frac{\#\{\text{"}delay\text{"} = 1, Y = 0\}}{\#\{Y = 0\}} \\ &\times \frac{\#\{\text{"}crash\text{"} = 0, Y = 0\}}{\#\{Y = 0\}} \end{aligned} \tag{17}$$

4.3.2. Classification on incident categories of TI tweets

Given a TI tweet, it is practically useful to know what category of incident it reports. In particular, we define five traffic incident categories for labeling and classifications:

1. Accidents: traffic accidents such as collision.
2. Road work: the scheduled or unplanned road work.
3. Hazards & Weather.
4. Events: special events such as Marathon.
5. Obstacle Vehicles.

A sophisticated classifier is built to assign a categorical label to those TI tweets. Specifically, a Supervised Latent Dirichlet Allocation (sLDA) (Mcauliffe and Blei, 2008) is used in our model. The benefits of using sLDA over other classifiers are as follows: (1) sLDA is a classical topic model, widely used in the domain of Natural Language Processing; (2) it has relatively fast training and prediction runtime using variational inference (Wainwright and Jordan, 2008); (3) it is able to identify not only the categorical label but also the underlying structure of how the words are generated; and (4) comparing to the basic topic model (Blei et al., 2003) or mixture model, sLDA is able to optimize both the likelihood and the deviation of predicted labels.

The conceptual process of sLDA is as follows (Blei et al., 2003). With the underlying K topics (categories) and for each tweet with the length N:

1. Draw topic proportions $\theta|\alpha \sim Dir(\alpha)$, where α is the parameter of the Dirichlet distribution.
2. Fore each word $n = \{1, 2, \ldots, N\}$ in a TI tweet:
   (a) Draw a topic of that word according to the multinomial distribution of the topic proportions of the TI tweet: $z_n|\theta \sim Multi(\theta)$.
   (b) Draw the word $w_n|z_n$ from the multinomial distribution of words over the topic: $w_n|z_n \sim Multi(\beta_{z_n})$, where β_{z_n} is the multinomial parameter of the topic z_n.
3. Draw the response variable $y \sim N(\eta \bar{z}_{1:N}, \sigma^2)$ over an normal distribution.

The inference of unknown latent parameters (α, β, η and σ^2) over the sLDA model follows the Expectation–Maximization (E–M) algorithm. More details can be found in Blei et al. (2003).

4.4. Geocoding

After the classification, we have identified all TI tweets in the acquired pool. Next we will extract their location information and geocode them in GIS.

The geographic location information carried by tweets is rich but very noisy. There are generally three types of location information. (1) A tiny portion of tweets carry latitude/longitude coordinates, and they are usually tweeted from geo-tagging enabled smart phones. In our experiments, this portion is never greater than 0.1%. These coordinates relate to the locations where users posted the TI tweets, but are not necessarily where those incidents are. (2) Some tweets are posted by accounts whose profiles are shared with the public, such as city, country, and sometimes finer-grained business names and street addresses of the business. Unfortunately, this type of location information generally does not imply incident locations. (3) Road names and points of interest may be referred in tweet texts. The main objective of our geocoding algorithm is to extract location information from the third type, namely tweet texts, and map each TI tweet to the GIS if possible.

The general idea is to first identify those words representing road names and/or point-of-interest (POI) names, followed by a geocoder that translates those names to latitude/longtitude coordinates of the incident. Some tweets, especially those tweeted by IUs, report incidents with highway exit numbers or mile markers. In that case, we build a GIS to geocode the exit numbers or mile markers into latitude/longtitude coordinates. The process of geocoding tweet texts is conceptually depicted in Fig. 3.

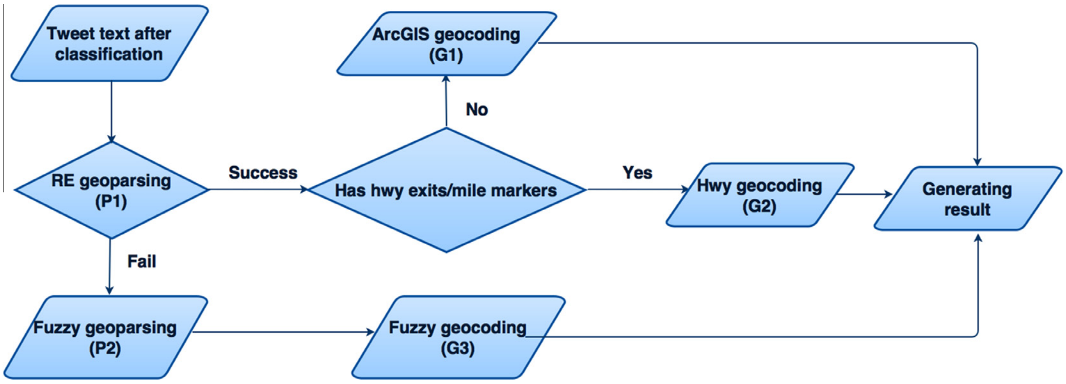


Fig. 3. Flow chart of tweets geocoding.

Table 4
The data structure of RE-based geoparser.

Keys	Meaning
road1:	The road name mentioned
road2:	The second road name mentioned
road3:	The third road name mentioned
hwy1:	The highway name mentioned
hwy2:	The second highway name mentioned
hwy3:	The third highway name mentioned
hwy1-mm1:	The starting mile-marker/exit number of the highway
hwy1-mm2:	The ending mile-marker/exit number of the highway
relational-word:	The relational word used like "near", "cross", "intersection", etc.
original-text:	The original tweet text

Table 5
An example of geo-parsing result.

Keys	Value
road1:	
road2:	
road3:	
hwy1:	I-376 WB
hwy2:	
hwy3:	
hwy1-mm1:	61.0
hwy1-mm2:	60.0
Relational-word:	"between"
Original-text:	Accident on I-376 westbound between Mile Post: 61.0 and Mile Post: 60.0...

A geo-parser is a machine that receives input of a string and produces a structured and segmented strings that contain only geographical information. As shown in Fig. 3, we use two geo-parsers. The first one (P1) is to carefully implement a large set of Regular Expressions (REs) to extract road names, intersection names, highway exit numbers, and highway mile markers. When the REs set is sufficiently large to cover all roads in a region, its geo-parser can work well to extract geographical information. However, it cannot process the names of point of interests commonly referred to in tweets, such as "Hamburg Hall" (a landmark building in Pittsburgh) and "Squirrel Hill" (a local neighborhood in Pittsburgh). Whenever P1 does not work, the secondary geo-parser (P2) developed by Gelernter and Balaji (2013) is adopted, where a fuzzy language matching algorithm is implemented to parse those words relevant to locations. Those fuzzy words are specified in a pre-defined dictionary. Comparing to P1, P2 can process point of interests but not road names and numbers. As shown in Fig. 3, the strategy is to apply P1 to a tweet, and whenever P1 fails, P2 is used instead.

P1 geo-parses a tweet following the structure shown in Table 4. It identifies either a segment of highway with starting and ending mile marker specified, a specified road, or intersections of up to three roads/highways. For instance, a tweet reads "Accident on I-376 westbound between Mile Post: 61.0 and Mile Post: 60.0. There is a lane restriction." Using P1, the parsing result is shown in Table 5.

The output of geo-parsers is then translated to latitude/longitude by geocoders. If a tweet is processed by the fuzzy geo-parser (P2), the output words are fed into a Gazetteer (Gelernter and Balaji, 2013) to identify the location. For the tweets processed by the RE geo-parser (P1), there are two possible geocoders. If a tweet has road names or intersection names without specifying highway mile markers (e.g., an arterial road), then the ArcGIS geocoder (G1) is used to generate latitude and longitude coordinates. However, a major drawback of G1 is that it cannot geocode those mile-markers or exit numbers of highways. If that is the case, a highway geocoder (G2) should be built. For the case study in this paper, we collect the GIS of all highways in Pennsylvania, and map the mileage of each of all highway junctions to a pair of latitude and longitude. G2 has some limitations and can be enhanced in the future research. It currently cannot compile vague relational words such as "to the north of" or "near". It does not correct misspelled road/highway names.

5. Case studies: Pittsburgh and Philadelphia Metropolitan Areas

In this paper, tweets from two regions, Pittsburgh and Philadelphia Metropolitan Areas, in September 2014 have been crawled and processed to retrieve incident information. Furthermore, in order to evaluate our Twitter-based incident detector in terms of timeliness, accuracy, and effectiveness, we performed extensive testing and validation summarized as follows:

(1) Testing on the classifiers and the geocoder: we test our classifier and geocoder by comparing the results with manual labels on all acquired tweets.
(2) Validation with existing incidents data: we compare the final output of the Twitter-based incident detector, namely the time, location, and category of incidents extracted from those geocoded TI-tweets, against a subset of "ground truth" data that include RCRS (Road Condition Report System) incident data maintained by PennDOT for all state-owned roads and 911 Call For Service (CFS) data provided by the City of Pittsburgh. This examines to what extent Twitter-based incident detector covers incidents reported by existing data sets, and the quantity of additional incidents Twitter can provide.
(3) Validation with HERE travel time data: we compare the travel time at the time and location of incidents extracted from Twitter against the historical travel time at the same location and the same time of day/week. For those incidents reported by Twitter but not by existing sources, this can provide evidence whether the Twitter-reported incidents are likely to be true.

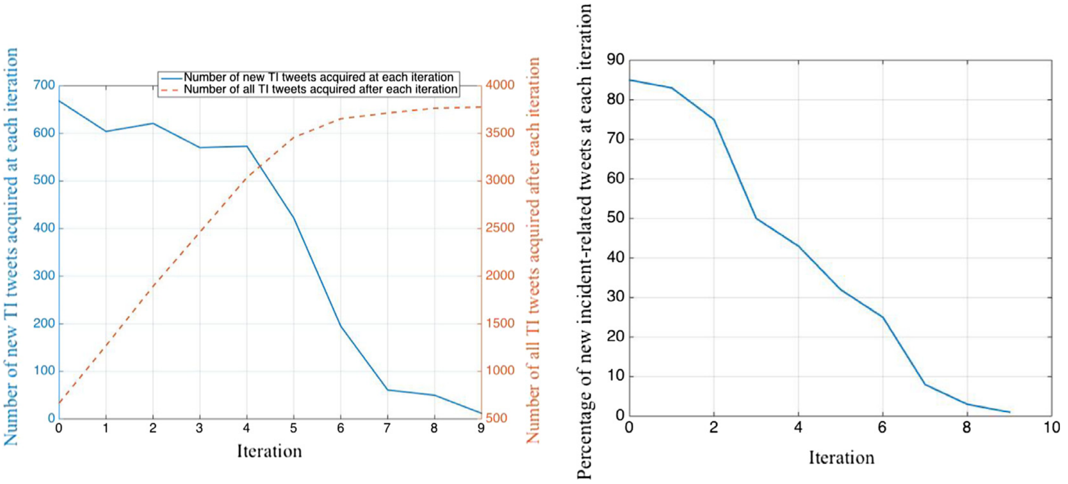
5.1. Twitter data acquisition

We first conduct the adaptive data acquisition following the aforementioned methodology. In particular, queries with keywords are made through REST API. In addition, we also query tweets posted by a list of in all 46 active Influential Users (IUs) such as Department of Transportation agencies (@511PAPhilly), News channels (@6abcBreaking), and other governmental agencies (e.g., @PGHtransit). We start with a dictionary with 50 "seed keywords" for the iterative data acquisition process. After 9 iterations, the acquisition process converges. As a result, we obtain a dictionary with 131 keywords (and some combinations) that are positively related to being a TI tweet, and 383 keywords that are negatively related. Using this dictionary, we acquired in all 10,542 and 11,658 tweets in the Pittsburgh and Philadelphia region in September 2014, respectively. As part of this data acquisition process, we have also manually labeled all tweets into TI tweets and NTI tweets as the "ground truth". There are 3776 and 4571 TI tweets in Pittsburgh and Philadelphia, respectively.

The convergence of the adaptive data acquisition process for Pittsburgh is shown in the Fig. 4. Fig. 4a shows that as iterations progress, the total number of new TI tweets acquired at each iteration drops quickly after 4 iterations. At the 9th iteration, the percentage of newly acquired TI tweets is less than 1% (Fig. 4b), implying that adding more words to the dictionary and manually labeling newly acquired tweets is no longer cost-effective to obtain additional TI tweets. The adaptive data acquisition process for the Philadelphia region follows a similar pattern. The total number of new TI tweets converges after 9 iterations.

5.2. Geocoding on manually labeled TI tweets

Since all tweets have been manually labeled, we know in fact whether they are TI tweets or not. We can therefore examine the effectiveness of our geocoding algorithms alone by geocoding those manually labeled TI tweets. This can eliminate



(a) The number of TI tweets

(b) The percentage of new TI tweets in each iteration

Fig. 4. Convergence of the iterative data acquisition process.

the error brought by the classifier when examining the geocoder. We will discuss the accuracy of the classifier and geocoder combined in the next subsection.

Without applying the classifier, the adaptive data acquisition and geocoding yield the results shown in Table 6. For Pittsburgh, among the total 10,542 acquired tweets, 3776 are TI tweets. 554 TI tweets directly or indirectly reported traffic incidents with meaningful time and location information. This result seems fairly impressive. In addition, tweets acquired from those 19 IUs in the Pittsburgh region account for merely 5.9% of all acquired TI tweets, but they contribute to 69.8% of those 554 TI tweets that are geocodable. This is not surprising since IUs' tweets tend to report traffic incidents with full details of the location. Most of them follow a clear linguistic structure, which allows accurate geocoding. For example, a typical tweet posted by an IU reads: "Turnpike Roadwork on Pennsylvania Turnpike I-476 northbound between Exit 31 – PA 63 and Exit 44 – PA 663 affecting the right lane", where the location information is clearly structured in this tweet.

As a comparison, a typical TI tweet posted by an individual reads: "@** – Daughter & Grandkids In Serious Car Crash http://dlvr.it/6tpycK @** @** all our prayers", with a picture of the accident attached (users' account is masked by **). This TI tweet contains vague descriptions about the accident, and it is hard to extract location information of this traffic accident. Though only 4.9% of TI tweets that are posted by individuals can be geocoded, they considerably complement the incident data, most of which are not necessarily reported by IUs. One example is "per traffic, 676 is not the blue route. Accident is at Vine St Expy and the Schkuykill Expy". By textual mining this tweet, we can identify the location of an accident at the intersection of Vine St Expy and the Schkuykill Expy. In Sep 2014, there are in all 156 and 175 instances of incidents reported by individuals in Pittsburgh and Philadelphia, respectively. Note that the free version of REST APIs only retrieves a small portion of tweets randomly. The incidents information from individual tweets may be more extensive if the full set of tweets data are available.

We plot the number of TI tweets and geocodable TI tweets by day of month for Pittsburgh in Fig. 5. It can be seen that there is clearly a pattern on the weekly basis: both numbers peak on weekends. On weekdays generally less incident information can be identified in Twitter. The daily number of geocodable TI tweets increases in an approximate portion to the daily number of acquired TI tweets, implying that a stable performance of the geocoders extracting location information from tweets.

Table 6
The result of data acquisition, manual labeling and geocoding.

	Pittsburgh	Philadelphia
TI tweets	3776	4571
NTI tweets	6766	7087
All tweets	10,542	11,658
IUs' tweets	621	554
Individual users' tweets	9921	11,104
TI tweets of IUs'	595	518
TI tweets of individual users'	3181	4053
Geocodable TI tweets	554	419
Geocodable TI tweets of IUs'	381	244
Geocodable TI tweets of individual users'	156	175
The portion of TI tweets in IUs' tweets	95.8%	93.5%
The portion of TI tweets in individual tweets	31.1%	37.5%
The portion of geocodable tweets in IUs' TI tweets	64.0%	47.1%
The portion of geocodable tweets in individual TI tweets	4.9%	4.3%
The portion of IUs' tweets in all acquired tweets	5.9%	4.8%
The portion of IUs' tweets in geocodable TI tweets	69.8%	58.2%

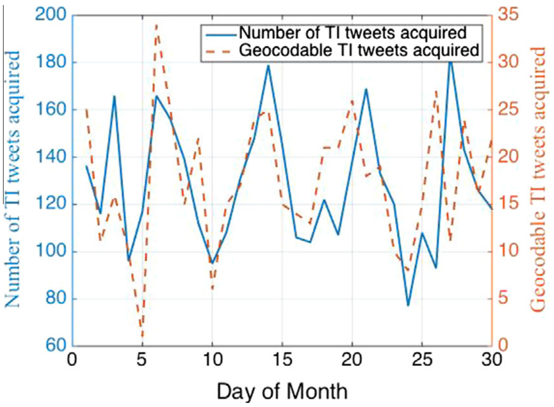


Fig. 5. The number of TI tweets by day of month.

5.3. Testing on the geocoder and classifiers

Next, a Semi-Naive-Bayes classifier is built based on the 131 words and combinations (namely features) that are positively related to being a TI tweet as a result of the data acquisition. The top ten features are, "*road-name*", "exit", "accident", "traffic", "roadwork", "lane", "PA", "mile", "cleared", and "post". For all Geocodable TI tweets, the sLDA categorical classifier is further applied to identify the category of the incident.

Note that all the acquired tweets have been manually labeled, and therefore we have the ground truth of TI tweets. In order to examine the effectiveness of the classifier and/or geocoding, we conduct four tests. Prior to the tests, we randomly select 5000 tweets from the acquired tweets of both cities as the test data set, and the remaining 17,200 tweets as the training data set.

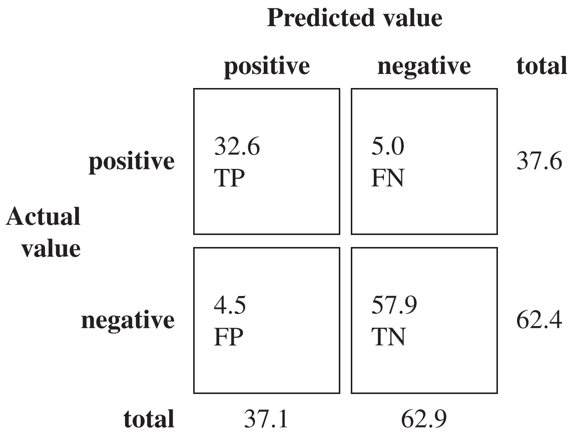
The first test is to test the accuracy of the Semi-Naive-Bayes classifier. The Semi-Naive-Bayes classifier is built with the training data set through a ten-fold cross validation approach, and applied to predict whether a tweet is TI for the test set. Second, we apply the geocoding methodology to the test data set directly without classifying them in prior. This examines how the geocoder works without a prior classification. Third, we follow our combined methodology where the test data set is Semi-Naive-Bayes classified and then processed by the geocoder. This allows us to investigate how effective the combined classifier and geocoder is.

In the first test, the resulting average confusion matrix is shown in the Table 7. The overall classification accuracy is 90.5% as a result of the cross validation. TP, FN, FP, and TN denotes True Positive, False Negative, False Positive, and True Negative, respectively. An interesting observation is that FN is approximately the same as FP. This indicates that this classifier identifies tweets as TI falsely, almost as much as it fails to identify actual TI tweets. Notice that this classification is based a "neutral" classification threshold of 50%, namely identify the label Y^* that yields the greater of $P(Y=1|X)$ and $P(Y=0|X)$. $Y^*=1$ indicates this tweet is a TI tweet, and 0 otherwise. In fact, this threshold can be adjusted in favor of users' experience. For example, if the user prefers to capture all the TI tweets as many as possible regardless of FP, one can set a small threshold <50%. A proper threshold value should be carefully chosen for the best results in practice.

The second test applies the geocoder only to the same test data set. The ground truth of a tweet being "geocodable" is determined by the criteria that whether or not a tweet contains point location information. For example, "along I-376" or "near downtown" is not sufficiently accurate for geocoding and they are not geocodable. However, "I-376 Exit 74" can be accurately geocoded in a map. Since we do not apply the classifier before geocoding, it is possible that a tweet is geocodable but not a TI tweet. An example is "FINALLY found them at the intersection of Dauphin street and 12th street". The confusion matrix is shown in Table 8. Among the 5.8% tweets that actually contain accurate point location information, the geocoder successfully identifies 82% of them, namely 4.8% of all acquired tweets.

In the third test, the test data set is first classified into TI tweets and NTI tweets using the Semi-Naive-Bayes classifier. Those tweets in the test set classified as TI tweets are then geocoded. A tweet is defined as "positive" if it is classified as a TI tweet and it is also geocodable. A confusion matrix is shown in Table 9. Note that since the percentage is too small, we use the actual numbers of the tweets instead of percentages. It can be seen from Table 9 that there are in all 253 geocodable TI-tweets in the test data set. The combined classifier and geocoder can successfully identify 202 of them. Notice that these 202 geocodable TI-tweets is obtained by running the geocoding algorithm on the TI tweets classified by the Semi-Naive-Bayes classifier. In contrast, if we run the geocoding algorithm the TI tweets manually labeled, we can obtain 219 geocodable tweets. This implies that the classifier misses 17 (219 − 202) useful tweets, while the other 34 (253 − 219) useful tweets cannot be picked up due to the limitations of this geocoder.

Table 7
Confusion matrix of the SNB classifier (all numbers are in percentages).



		Predicted value		
		positive	negative	total
Actual value	positive	32.6 TP	5.0 FN	37.6
	negative	4.5 FP	57.9 TN	62.4
	total	37.1	62.9	

Table 8
Confusion matrix of the geocoder (all numbers are in percentages).

		Predictive value		
		positive	**negative**	**total**
Actual value	**positive**	4.8 TP	1.0 FN	5.8
	negative	0.2 FP	94.0 TN	94.2
	total	5.0	95.0	

Table 9
Confusion matrix of the combined classifier and geocoder.

		Predictive value		
		positive	**negative**	**total**
Actual value	**positive**	202 TP	51 FN	253
	negative	2 FP	4745 TN	4747
	total	204	4796	

The fourth test is performed on the sLDA classifier. Following the same process as previous tests, we randomly select 873 of all the 973 geocodable TI tweets for both Pittsburgh and Philadelphia as a training set and the remaining 100 tweets as a test set. A sLDA classifier is built by the training set and is used to perform classifications on the testing set. Notice that the output of the sLDA classifier is a vector containing the probability of this tweet falling into each of the five categories. We choose the category with the highest probability as the classified label of the tweet. The testing true positive rate is 51%. This implies that 51% of the geocodable TI tweets can be correctly classified by the sLDA classifier.

It is worth noting that the two trained classifier and the geocoder is computationally efficient. In our case studies using a regular desktop (CPU i5-2500k +8 GB memory), all tweets acquired from the API can be classified by the SNB and sLDA in 3.1 ms. Those TI tweets can be geocoded within 87 ms. Therefore, the computational time cost by the algorithms can be negligible comparing to the time between an incident actually occurs and when it is reported by a user.

5.4. Validation with existing incidents data

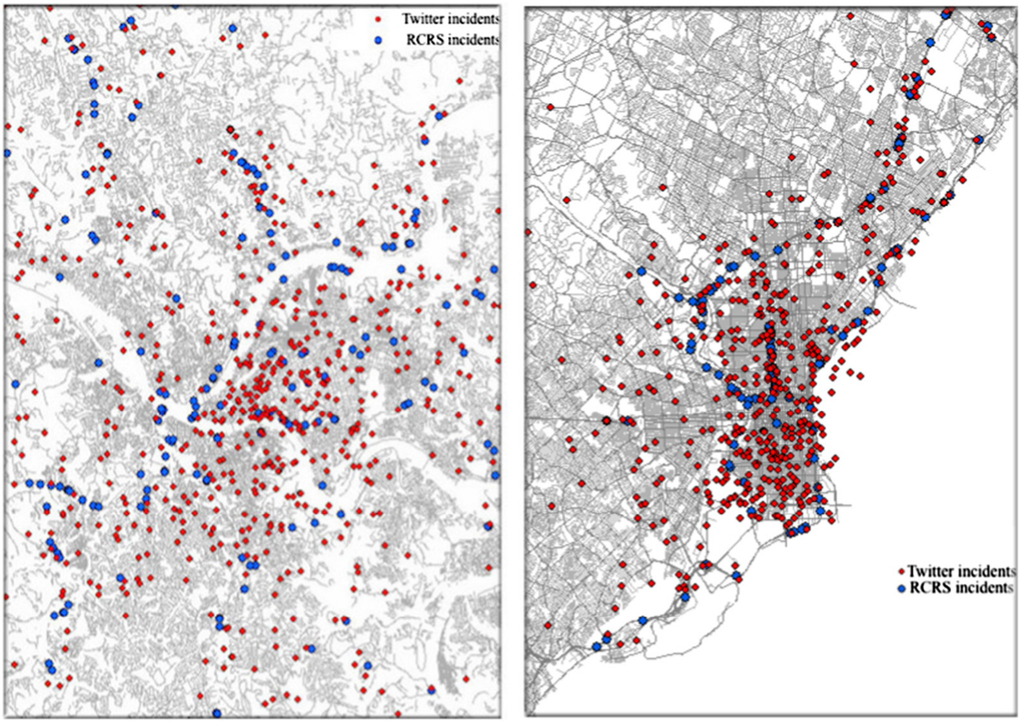
After geocoding the manually labeled TI tweets, we obtain 554 geocodable tweets for Pittsburgh and 419 for Philadelphia in Sep 2014. For both the Pittsburgh and Philadelphia regions, we obtained RCRS (Road Condition Report System) traffic incident data, maintained by the Department of Transportation Pennsylvania. RCRS covers all state-owned roads and is the first and only data stream fed to 511PA.org and Transportation Management Centers (TMC) in the real time. RCRS data contain incident details which include the location, reported time, as well as the category of the incident. In Sep 2014, RCRS reported 217 incidents in Pittsburgh and 105 in Philadelphia on state-owned roads. By representing each geocodable TI tweet with a stand-alone dot, scatter plots of incidents reported by RCRS and Twitter are shown in Fig. 6a and b for each region respectively. The Twitter-based incident detector seems to have much wider spatial coverage, especially on arterials.

For the Pittsburgh region, we also obtained 911 Call For Service (CFS) data for the Allegheny County. CFS data contain information of all the calls to 911 regarding traffic, such as "time of dispatch", "address of the incident", "disposition", and "call type". Unlike the RCRS data which contains the exact time and location of the reported traffic incidents, the CFS data needs to be pre-processed to extract location information. First, we choose the entries which has the "call type" relating to traffic incidents, such as "TRAFFIC-HIGH MECHANISM (AUTO-PEDESTRIAN)" and "TRAFFIC -WITH INJURIES". Second, we discard the entries which has the "Disposition" that appears unlikely to validate, such as "Unable to Locate" and "Gone on Arrival". Finally, we geo-code the "Address" field into the exact longitude and latitude using the aforementioned geocoder. After the pre-processing, we obtained 106 traffic incidents reported by the CFS for the Allegheny County.

In addition, note that in order to compare incidents reported from Twitter, CFS and RCRS most effectively, those Twitter incidents that were originated from official PennDOT Twitter accounts and Pittsburgh Bureau of Police (PBP) accounts are left out. We do not intend to use Twitter to repeat incident information that is known to both agencies. Rather, the intension is to provide additional information. After excluding those Twitter incidents reported by PennDOT and PBP, we have in all 698 Twitter incidents for the two regions.

To compare Twitter with existing traffic incident data sources, we treat the union of RCRS incidents and CFS incidents as the "ground truth" (we use "RCRS + CFS incidents" for short). First, we explore the percentage of RCRS + CFS incidents covered by Twitter. To quantify this coverage, we measure how "close" the RCRS + CFS incidents are to those Twitter-reported incidents both temporally and spatially. The time and location of an incident reported by tweets and RCRS may not precisely overlap, possible due to reporting time discrepancies and/or incomplete location information. Therefore, we would allow a small variation of reporting time and locations for data to suggest the same incidents. In Fig. 7, we plot the percentage of RCRS + CFS incident matched by Twitter incidents within a specified temporal and spacial radius of each RCRS + CFS incident. The temporal radius is 10, 20, and 30 min whereas the spatial radius ranges from 0 to 1.4 miles. This figure illustrates that a greater temporal or spatial radius results in greater matched percentages. When allowing 30-min reporting time discrepancy and 1-mile distance for the data to report the same incident, tweets (excluding those PennDOT and Pittsburgh Police tweets) can report 71% of the entire set of RCRS + CFS incidents. This implies that the majority of the RCRS + CFS incidents are also reported in Twitter.

As summarized in Table 10, among the 698 incidents reported by Twitter excluding PennDOT and PBP IUs, 492 of them are actually reported by either RCRS or CFS, when allowing 30-min reporting time discrepancy and 1-mile distance. Due to repetitive reporting in Twitter, those 492 reported incidents represent, in fact, 304 unique incidents reported by RCRS + CFS.



(a) Pittsburgh (b) Philadelphia

Fig. 6. RCRS and Twitter incidents in Pittsburgh and Philadelphia.

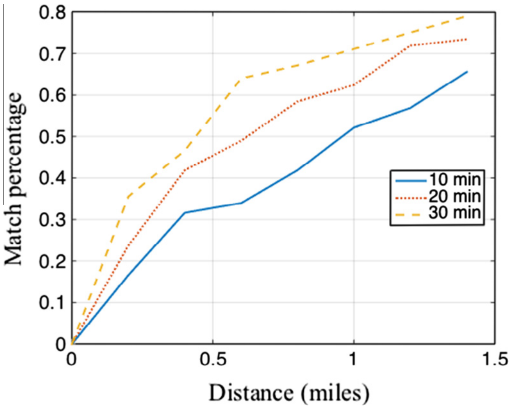
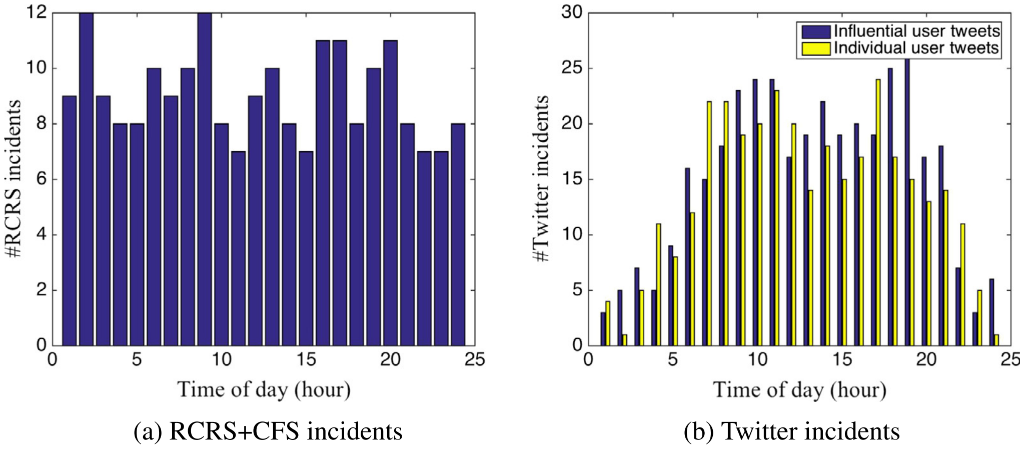


Fig. 7. The matching rate between RCRS + CFS incidents and Twitter incidents.

Table 10
The number of incidents reported by different data sources.

Data source	Number of incidents
Twitter	973
RCRS	322
CFS	106
RCRS + CFS	428
Twitter (excluding PennDOT and PBP IUs)	698
Twitter (excluding PennDOT and PBP IUs, matched by RCRS + CFS in 1 mile and 30 min)	492
RCRS + CFS (matched by Twitter in 1 mile and 30 min, excluding PennDOT and PBP IUs)	304
Twitter “additional” incidents	206



(a) RCRS+CFS incidents

(b) Twitter incidents

Fig. 8. Time-of-day distribution of incidents reported by RCRS and Twitter.

There are 206 “additional” incidents provided by Twitter that are reported by neither RCRS nor CFS. If we use the ratio 492/304 = 1.6 to denote on average the number of tweets reporting the same unique incident, Twitter can add 206/1.6 = 129 “additional” incidents on top of RCRS and CFS.

Now we examine the spatial and temporal distributions of incidents reported by Twitter and RCRS + CFS. Fig. 8 illustrates the number of incidents by RCRS + CFS and Twitter against time of day reported in both Pittsburgh and Philadelphia. The RCRS + CFS reported incidents are almost evenly distributed throughout the entire day with slight spikes in peak hours. However, for Twitter incidents (excluding PennDOT and PBP IUs), the time-of-day distributions of both IUs’ and individuals’ Twitter incidents are completely different: most of the incidents were reported during the day time, especially morning and afternoon peak hours. The advantage of Twitter-based incident detection is that it has a more extensive coverage during the day time, but it may not be as reliable as RCRS in the night from 8 pm to 5 am.

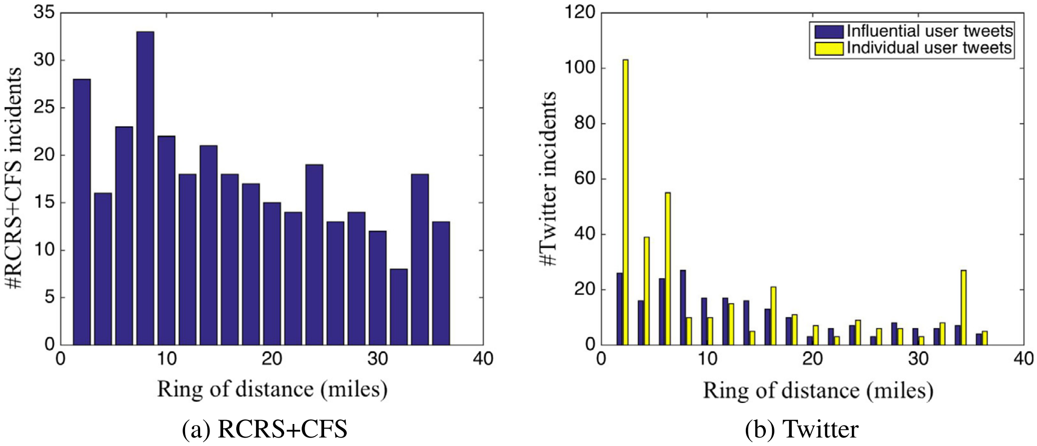
To compare the spatial distribution between Twitter incidents and RCRS + CFS incidents, we calculate the distance from the location of the incidents to the center of the city (with the location 40.440731, −79.995751) for Pittsburgh area. In Fig. 9, we plot the number of RCRS + CFS incidents and Twitter incidents against the radius around the city center. It can be seen that the number of incidents reported RCRS + CFS decay slightly with respect to the distance away from the city center in the Pittsburgh region. However, the dramatic drop in Twitter incident quantities over distance is the result of dense incident reports towards the city center, an important feature of the Twitter-based incident detector. The individuals' tweets are more likely to report incidents occurred near the city center, while IUs' incidents seem more evenly distributed along the spatial dimension, similar to the RCRS + CFS data. This is not surprising since there are more active Twitter users near the city center than outside. Those figures in Philadelphia seem very similar to Pittsburgh.

Furthermore, we apply the categorical classification on those geocodable TI tweets. It can be seen from Fig. 10 that in Pittsburgh, for both RCRS and Twitter, the two most frequent categories are accidents and road work. In addition, Twitter tends to report slightly more events, Hazards, weather conditions, and obstacle vehicles than RCRS and CFS.

5.5. Validation with HERE travel time data

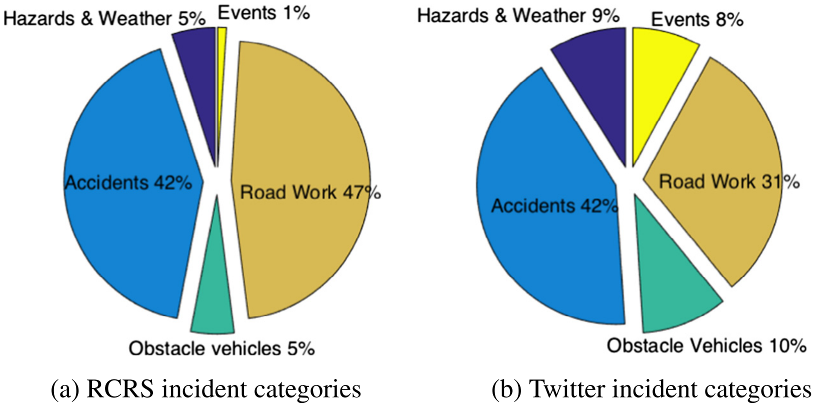
In addition to comparing the incidents reported by Twitter with existing data sources, we also validate the detection accuracy by examining the travel time measures near the incident location. The primary assumption of our analysis is that, if there is a traffic incident on the road, then the travel time will substantially vary from the typical travel time, and vice versa. By comparing the travel time near the location of the incident with the historical travel time at the same location and the same time-of-week, we are able to identify whether or not the travel time increase is statistically significant and thus infer whether there is an incident. In this section, we use statistical hypothesis tests on: (1) all Twitter-reported incidents combined and (2) each Twitter-reported incident.

The travel time data is obtained from HERE, as part of the National Performance Management Research Data Set. This data set contains time-varying travel time on major roads in both the Philadelphia and Pittsburgh region.



(a) RCRS+CFS (b) Twitter

Fig. 9. Spatial distribution of incidents reported by RCRS and Twitter.



(a) RCRS incident categories (b) Twitter incident categories

Fig. 10. Pittsburgh RCRS and Twitter incidents categories.

5.5.1. Hypothesis test on the entire set of Twitter incidents

Suppose there are in all N Twitter incidents. The measured travel time of the road segments that are in the vicinity of the occurrence time and location of the i-th Twitter incident is t_i, also known as "actual travel time". In particular, we define t_i as the average travel time from half an hour before the incident occurs to half an hour after. The average is taken over the 12 5-min intervals within the one-hour time period. Similarly, we define h_i the "typical travel time" at the same time and location as the i-th incident without the presence of any incident. We retrieve the travel times at the same location, the same time of day, and the same day of week over the previous eight weeks as H_i, a vector of eight measurements of h_i.

We standardize the travel time by:

$$t'_i = \frac{t_i - E(H_i)}{Std(H_i)} \tag{18}$$

and

$$h'_i = \frac{h_i - E(H_i)}{Std(H_i)}, \quad H'_i = \frac{H_i - E(H_i)}{Std(H_i)} \tag{19}$$

where $E()$ is the operator of mean and $Std()$ is the operator of standard deviation.

Since both the typical travel time and actual travel time are standardized, it makes sense to constitute a distribution for typical and actual travel time across all the time and locations, namely t' and h' respectively. The distributions t' and h' can be estimated using the sampled t'_i and H'_i respectively. By comparing h' and t', we are able to show how statistically different the typical travel time and actual travel time are for all time and locations.

The distributions of typical and actual travel time are shown in Fig. 11. It can be clearly seen that the distribution of the actual travel time at the time and location where Twitter reports an incident appears significantly different from the distribution of the typical travel time at the same time and location. To further quantify the difference, we performed a Kolmogorov–Smirnov (K–S) hypothesis test under the null assumption that "The typical travel time and actual travel time have the identical distribution".

The resulting P-value of this K–S test is 7.9965e−15, indicating the rejection of the null hypothesis. Therefore, we conclude that there is significant evidence when Twitter reports an incident the travel time of adjacent road segments is statistically different from the day-to-day typical travel time. This partly validates those incidents reported by Twitter are very likely to be true.

5.5.2. Hypothesis tests on individual Twitter incidents

For each Twitter-reported incident, we can perform a hypothesis test with the null hypothesis "The measured actual travel time follows the Gaussian distribution of the typical travel time at the same time and location where an incident is reported by Twitter". For the i-th incident, H'_i constitutes a sample that is used to estimate the Gaussian distribution of the typical travel time at a particular time and location. t'_i is a measured sample of the actual travel time. The test in general is a Z-test. The P-value of the i-th Twitter incident is given by,

$$P_i = Prob(h'_i \geqslant t'_i) \tag{20}$$

The final step of this hypothesis test is to define a significance level U for each incident i, where

$$\begin{cases} P_i \geqslant U & \text{Fail to reject the null hypothesis} \\ P_i < U & \text{Reject the null hypothesis} \end{cases}$$

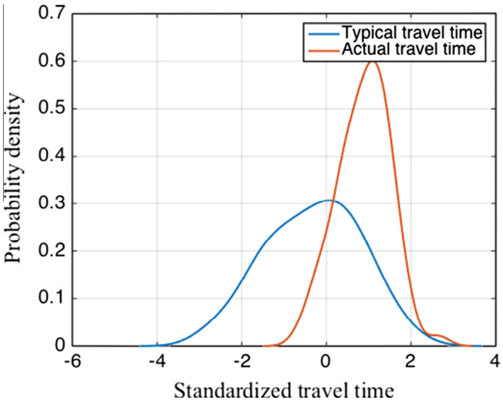


Fig. 11. Typical travel time and actual travel time.

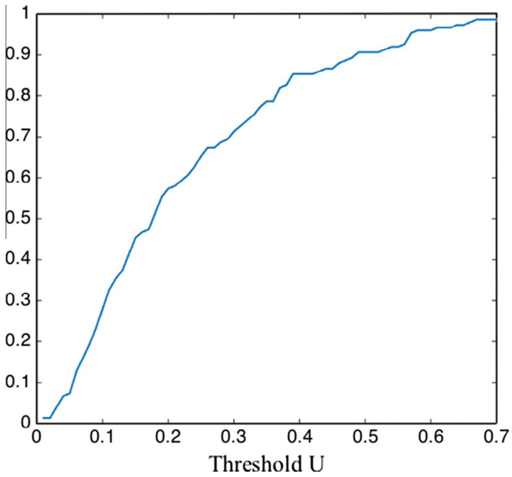


Fig. 12. The influence of U on the percentage of Twitter-reported incidents being statistically true.

Also notice that "rejecting the null hypothesis" implies that the *i*-th Twitter incident is likely to be a true traffic incident. Here instead of subjectively defining one single threshold for the significance level *U*, we explore how *U* can influence the percentage of Twitter-reported incidents being true. As we can see from Fig. 12 that when the threshold of *U* is set to be around 0.5, we will reject the null hypothesis for 87% of the Twitter-reported incidents, namely those incidents reported are likely to be true. When the threshold of *U* is 0.1, we are confident that at least 28% of Twitter-reported incidents are true.

The statistical tests on each incident should be used with cautions. Theoretically, the threshold of *U* guarantees that the Type I error of this statistical test rate is at most *U*, which is not the probability of the null hypothesis being true. "Failure to reject the null hypothesis" does not necessarily indicate that "the null hypothesis is more likely to be true". In addition, the underlying assumption that "if there is a traffic incident in a road, the travel time will substantially vary from the typical travel time, and vice versa" makes the hypothesis tests possible, but it may not be always true. For example, incidents on one lane of a multi-lane freeway segment in the light traffic do not leads to an increase of travel time. In this sense, those hypothesis tests are very conservative. After all, the statistical tests here provide some evidence that a significant portion of the incidents reported by Twitter is likely to influence the travel time, and thus to be true. However, the actual false positive and false negative would require intensive field test in the near future.

6. Conclusions and future work

This paper proposes to mine tweet texts to extract incident information on both highways and arterials as an efficient and cost-effective alternative to existing incident data sources. We present a methodology to crawl, process and filter tweets that are accessible by the public for free. Tweets are acquired from Twitter servers using the REST API. The data acquisition follows an iterative process. It starts with queries to APIs with a dictionary of "initial keywords" and iteratively expands the dictionary following a simple Natural language processing (NLP) approach until the acquired tweet data set converges. The process of adaptive data acquisition also selects the most important keywords and their combinations to form a feature space that is informative and non-redundant. Next, tweet texts are mapped into a high dimensional binary vector in this feature space defined by the dictionary. The vectorized tweets are then classified by a well-trained Semi-Naive-Bayes (SNB) classifier as either TI or NTI tweets. All the TI tweets are then pushed through a geo-parser and geocoder to determine their locations. Finally, the geocoded TI tweets are classified by a Supervised Latent Dirichlet Allocation classifier into one of the five incident categories.

We apply the methodology in two regions, the Pittsburgh and Philadelphia Metropolitan Areas in September 2014, to extensively examine both temporal and spatial coverage of detected incidents. The Twitter-based incidents are then validated using RCRS (Road Condition Report System) incident data, 911 Call For Service (CFS) incident data, and HERE travel time data. We demonstrate that mining social media data (using Twitter as an example in this paper) holds great potentials to complement incident reporting sources in a very efficient and cost-effective way.

In summary, we found that a small sample of tweets acquired from the Twitter API cover most of the incidents reported in the existing data set, and additional incidents can be identified through analyzing tweets texts on top of what PennDOT has known. A useful tweet, being traffic incident (TI) related and geocodable in the map, accounts for approximately 5% of all acquired tweets. Of those geocodable TI tweets, 60–70% are posted by influential users, namely public Twitter accounts mostly owned by public agencies and media, while the rest is contributed by individual users. There exists a clear weekly pattern on the daily number of TI tweets and geocodable TI tweets: more incident information is provided by Twitter on weekends than on weekdays. Within the same day, both individuals and IUs tend to report incidents more frequently during the day time than at night, especially during morning and afternoon peak hours. In terms of spatial distribution, Twitter-based incident detection has a lot more extensive coverage on arterials, which provides a cheap alternative to existing data sources. Individual tweets are more likely to report incidents near the center of a city, and the volume of information decays outwards from the city center. This is not surprising since there are more active Twitter users in the city than outside.

The methodology can be further enhanced and improved in several ways. First, we will acquired more tweets to examine the incident coverage. Currently, the data acquisition is limited by the maximum number of tweets randomly sampled from the entire tweet pool. Additional tweets provided by data vendors would allow us to evaluate the cost and benefits of higher tweet sampling rate. Second, more sophisticated NLP models can be applied to better classify TI tweets. The geocoder can be enhanced by incorporating additional names of roads and points of interests with the capacity to correct misspelled names. Last but not least, we will test the real-time Twitter-based incident detector in Pennsylvania TMCs to examine its efficiency and effectiveness in practice.

Acknowledgements

This research is initiated by the Pennsylvania Department of Transportation (PennDOT), and funded by PennDOT, Federal Highway Administration (FHWA) and Carnegie Mellon University's Technologies for Safe and Efficient Transportation (T-SET). T-SET is a National University Transportation Center for Safety sponsored by the US Department of Transportation. The contents of this research reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. The U.S. federal and state government assumes no liability for the contents or use thereof. We would like to thank Mark Kopko from PennDOT for his valuable comments and help in data collection.

References

Abdulhai, B., Ritchie, S.G., 1999. Enhancing the universality and transferability of freeway incident detection using a Bayesian-based neural network. Transport. Res. Part C: Emerg. Technol. 7 (5), 261–280.

Adler, J., Horner, J., Dyer, J., Toppen, A., Burgess, L., Hatcher, G., 2015. Using Crowdsourced Data from Social Media to Enhance TMC Operations, vol. FHWA-JPO-14-165. Federal Highway Administration.

Ahmed, F., Hawas, Y., 2015. An integrated real-time traffic signal system for transit signal priority, incident detection and congestion management. Transport. Res. Part C: Emerg. Technol. 60, 52–76.

Ahmed, M., Cook, A., 1977. Analysis of freeway traffic time-series data using Box–Jenkins techniques. Transport. Res. Rec. (722), 113–116

Ahmed, M., Cook, A., 1982. Application of time-series analysis techniques to freeway incident detection. Transport. Res. Rec. (841), 92-21

Ahmed, S.A., Cook, A.R., 1980. Time series models for freeway incident detection. Transport. Eng. J. Am. Soc. Civil Eng. 106 (6), 731–745.

Analytics, P., 2009. Twitter Study. Pear Analytics, San Antonio, TX <www.pearanalytics.com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009.pdf>.

Anbaroglu, B., Heydecker, B., Cheng, T., 2014. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. Transport. Res. Part C: Emerg. Technol. 48, 47–65.

Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent Dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022.

Bontcheva, K., Rout, D., 2012. Making sense of social media streams through semantics: a survey. Semantic Web 1, 1–31.

Collins, J., Martin, J., 1979. Automatic incident detection? TRRL algorithms HIOCC and PATREG. TRRL Supplementary Report (526).

Demissie, M.G., de Almeida Correia, G.H., Bento, C., 2013. Intelligent road traffic status detection system through cellular networks handover information: an exploratory study. Transport. Res. Part C: Emerg. Technol. 32, 76–88.

Dudek, C., Messer, C., Nuckles, N., 1974. Incident detection on urban freeway. Transport. Res. Rec. (495), 12–24

Fu, K., Nune, R., Tao, J.X., 2015. Social media data analysis for traffic incident detection and management. In: Transportation Research Board 94th Annual Meeting No. 15-4022.

Gao, H., Barbier, G., Goolsby, R., 2011. Harnessing the crowdsourcing power of social media for disaster relief. IEEE Intell. Syst. (3), 10–14

Gelernter, J., Balaji, S., 2013. An algorithm for local geoparsing of microtext. GeoInformatica 17 (4), 635–667.

Herrera, J.C., Work, D.B., Herring, R., Ban, X.J., Jacobson, Q., Bayen, A.M., 2010. Evaluation of traffic data obtained via GPS-enabled mobile phones: the Mobile Century field experiment. Transport. Res. Part C: Emerg. Technol. 18 (4), 568–583.

Hu, M., Liu, S., Wei, F., Wu, Y., Stasko, J., Ma, K.-L., 2012. Breaking news on Twitter. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, pp. 2751–2754.

Kamran, S., Haas, O., 2007. A multilevel traffic incidents detection approach: identifying traffic patterns and vehicle behaviours using real-time GPS data. In: Intelligent Vehicles Symposium. IEEE, pp. 912–917.

Khan, S.I., Ritchie, S.G., 1998. Statistical and neural classifiers to detect traffic operational problems on urban arterials. Transport. Res. Part C: Emerg. Technol. 6 (5), 291–314.

Krstajic, M., Rohrdantz, C., Hund, M., Weiler, A., 2012. Getting there first: real-time detection of real-world incidents on Twitter. In: 2nd Workshop on Interactive Visual Text Analytics: Task-Driven Analysis of Social Media Content with Visweek'12.

Lee, J.H., Gao, S., Janowicz, K., Goulias, K.G., 2015. Can Twitter data be used to validate travel demand models? In: The 14th International Conference on Travel Behaviour Research.

Mcauliffe, J.D., Blei, D.M., 2008. Supervised topic models. In: Advances in Neural Information Processing Systems, pp. 121–128.

Park, H., Haghani, A., 2015. Real-time prediction of secondary incident occurrences using vehicle probe data. Transport. Res. Part C: Emerg. Technol.

Payne, H., Tignor, S., 1978. Freeway incident-detection algorithms based on decision trees with states. Transport. Res. Rec. (682), 30–37

Pereira, F.C., Rodrigues, F., Ben-Akiva, M., 2013. Text analysis in incident duration prediction. Transport. Res. Part C: Emerg. Technol. 37 (December), 177–192.

Ritchie, S.G., Cheu, R.L., 1993. Simulation of freeway incident detection using artificial neural networks. Transport. Res. Part C: Emerg. Technol. 1 (3), 203–217.

Sakaki, T., Okazaki, M., Matsuo, Y., 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web. ACM, pp. 851–860.

Sasaki, K., Nagano, S., Ueno, K., Cho, K., 2012. Feasibility study on detection of transportation information exploiting Twitter as a sensor. In: Sixth International AAAI Conference on Weblogs and Social Media.

Schulz, A., Ristoski, P., Paulheim, H., 2013. I see a car crash: real-time detection of small scale incidents in microblogs. In: The Semantic Web: ESWC 2013 Satellite Events. Springer, pp. 22–33.

Sermons, M.W., Koppelman, F.S., 1996. Use of vehicle positioning data for arterial incident detection. Transport. Res. Part C: Emerg. Technol. 4 (2), 87–96.

Sethi, V., Bhandari, N., Koppelman, F.S., Schofer, J.L., 1995. Arterial incident detection using fixed detector and probe vehicle data. Transport. Res. Part C: Emerg. Technol. 3 (2), 99–112.

Stephanedes, Y.J., Chassiakos, A.P., 1993. Freeway incident detection through filtering. Transport. Res. Part C: Emerg. Technol. 1 (3), 219–233.

Systematics, C., 2005. Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation, vol. 6. Federal Highway Administration.

Teng, H., Qi, Y., 2003. Application of wavelet technique to freeway incident detection. Transport. Res. Part C: Emerg. Technol. 11 (3–4), 289–308.

Thancanamootoo, S., Bell, M., 1988. Automatic detection of traffic incidents on a signal-controlled road network. Res. Rep. (76)

Tigor, S., Payne, H., 1977. Improved freeway incident detection algorithms. Public Roads 41 (1), 32–40.

Tsai, J., Case, E., 1979. Development of freeway incident detection algorithms by using pattern-recognition techniques. Transport. Res. Rec. (722), 113–116

Wainwright, M.J., Jordan, M.I., 2008. Graphical models, exponential families, and variational inference. Found. Trends® Mach. Learn. 1 (1-2), 1–305.

White, J., Thompson, C., Turner, H., Dougherty, B., Schmidt, D.C., 2011. Wreckwatch: automatic traffic accident detection and notification with smartphones. Mob. Netw. Appl. 16 (3), 285–303.

Willsky, A.S., Chow, E., Gershwin, S., Greene, C., Houpt, P., Kurkjian, A., 1980. Dynamic model-based techniques for the detection of incidents on freeways. IEEE Trans. Autom. Control 25 (3), 347–360.

Xiang, Z., Gretzel, U., 2010. Role of social media in online travel information search. Tour. Manage. 31 (2), 179–188.

Yates, D., Paquette, S., 2011. Emergency knowledge management and social media technologies: a case study of the 2010 Haitian earthquake. Int. J. Inf. Manage. 31 (1), 6–13.

Yuan, F., Cheu, R.L., 2003. Incident detection using support vector machines. Transport. Res. Part C: Emerg. Technol. 11 (3), 309–328.

Zhang, K., Taylor, M.A., 2006. Effective arterial road incident detection: a bayesian network based algorithm. Transport. Res. Part C: Emerg. Technol. 14 (6), 403–417.