pennsylvania
DEPARTMENT OF TRANSPORTATION

Real-time Incident Detection Using Social Media Data

FINAL REPORT

May 9, 2016

By Zhen (Sean) Qian
Carnegie Mellon University

Carnegie Mellon

COMMONWEALTH OF PENNSYLVANIA
DEPARTMENT OF TRANSPORTATION

CONTRACT # CMUIGA2012
WORK ORDER # 03

DEPARTMENT OF TRANSPORTATION
UNITED STATES OF AMERICA

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-PA-2016-004-CMU WO 03		

4. Title and Subtitle	5. Report Date
Real-time Incident Detection Using Social Media Data	May 9, 2016
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Zhen (Sean) Qian (ORCID: 0000-0001-8716-8989)	WO-003

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Carnegie Mellon University 5000 Forbes Avenue Pittsburgh, PA1 15213	**11. Contract or Grant No.** CMUIGA2012 – CMU WO 03

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
The Pennsylvania Department of Transportation Bureau of Planning and Research Commonwealth Keystone Building 400 North Street, 6th Floor Harrisburg, PA 17120-0064	Final Report: May 11, 2015 to May 10, 2016
	14. Sponsoring Agency Code

15. Supplementary Notes

Mark Kopko, technical advisor

16. Abstract

The effectiveness of traditional incident detection is often limited by sparse sensor coverage, and reporting incidents to emergency response systems is labor-intensive. This research project mines tweet texts to extract incident information on both highways and arterials as an efficient and cost-effective alternative to existing data sources. This research report 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 10% of all the acquired tweets. Of those geocodable TI tweets, the majority are posted by influential users (IU), namely public Twitter accounts owned by public agencies and media, while a small number 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. We develop a prototype web application to allow users extract both real-time and historical incident information and visualize it on the map. The web application will be tested in PennDOT transportation management centers.

17. Key Words	18. Distribution Statement
Social media, incident detection, data mining, twitter, web application	No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	46	

Acknowledgement

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. We would like to thank Mark Kopko, the technical advisor from PennDOT for his valuable comments and help in facilitating this research. We would also like to thank the support of the Bureau of Planning and Research, Research Division at PennDOT.

Disclaimer

Executive Summary

The effectiveness of traditional incident detection is often limited by sparse sensor coverage, and reporting incidents to emergency response systems is labor-intensive. This research project mines tweet texts to extract incident information on both highways and arterials as an efficient and cost-effective alternative to existing data sources. This research report presents a methodology to crawl, process and filter tweets that are accessible by the public for free. Tweets are acquired from Twitter using the representational state transfer (REST) Application Program Interfaces (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 10% of all the acquired tweets. Of those geocodable TI tweets, the majority are posted by influential users (IU), namely public Twitter accounts owned by public agencies and media, while a small number 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.

We developed a prototype web application to allow users to extract both real-time and historical incident information and visualize it on the map. The web application will be tested in PennDOT transportation management centers.

Table of Contents

List of Figures

List of Tables

1 Introduction

For decades, research has been dedicated towards 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 detection automation is the assumption that significant change in flow characteristics immediately follows 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, street parking, etc. Given the current sensing coverage, it is notoriously difficult to accurately detect arterial incidents. Crowdsourcing seems to be a solution to this matter for its low cost, real-time capacity and reasonable accuracy.

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 Application Program Interfaces (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 tools (such as Waze).

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 are often considered as "pointless babble", making it difficult to separate useful information from plain noise (Analytics, 2009).

The aim of this research task is to bridge the gap between the massive potential information existing in the social media data (e.g. Twitter) and the need for accurate, inexpensive, and real-time traffic incident information. The reason of choosing Twitter as a representative of social media sites is as follows:

(1) we have access to a reasonably large Twitter database, CMU-Gardenhouse, available free of charge for research. A portion of Twitter data is accessible by developers in the real time through Twitter's APIs, which allows us to develop real-time incident detection tools based on Twitter;

(2) Comparing to Facebook, a portion of Twitter data is accessible. However, Facebook data is proprietary;

(3) Comparing to Google+, Twitter offers versatile APIs for crawling, searching, and mining the Twitter data. The APIs offered Google+ have very limited functionalities, and therefore Google+ may not be a good source for real-time incident detection.

We intend to answer the following questions:

(1) How frequently do Twitter users in selected region and corridor tweet about incidents?
(2) What types of incidents do they tweet about?
(3) Are there any locations about which people tweet more often?
(4) What is the ratio of overall Twitter data in selected regions and corridors to data that is relevant to incidents?
(5) Are there any particular times (of day/year) or conditions for which users tweet more about incidents?
(6) Can social network analysis techniques identify key influencers who tweet about incidents?
(7) How to identify in real time if a tweet is incident related?
(8) How to classify events in all incidents related tweets in the real time?
(9) How to infer the geo-location of the tweet and map it to the road network in the real time?
(10) In real time, what percentage of incidents can be detected and what percentage of those can be precisely geo-coded?
(11) How timely is the Twitter-based incident detector?
(12) How to establish a score system to indicate confidence levels of valid incident detection?

This report is organized as follows. We first discuss the basic assumptions of the domain model, the data structure of tweets, and the categorization of the incidents in ***Section 2,*** followed by descriptions of the data mining model used for incident detection in ***Section 3***. In ***Section 4***, we propose and test the offline version of the Twitter-based incident detector, and aim to answer question (1)-(6). In ***Section 5***, we extend the established offline version of Twitter-based incident detector to an online version, and aim to answer question (7)-(12). ***In Section 6***, we explore the possibility of using the audio data from public radio stations as an alternative of acquiring traffic incident data. Finally in ***Section 7***, we introduce a Web-based interface for users to access the Twitter-based traffic incidents in real-time.

2 Traffic Incident Domain Model

2.1 Incident categorization

PennDOT Road Condition Reporting System (RCRS) is a statewide tool used by all Pennsylvania Department of Transportation engineering districts to ensure consistency and accuracy when reporting road closure and condition information on state highways. RCRS data set has categorized all incidents as follows,

1 ACCIDENT
2 DEBRIS ON ROADWAY
3 WINTER WEATHER
4 SPECIAL EVENT
5 OTHER
6 ROADWORK
7 FLOODING
8 BRIDGE OUTAGE
9 DOWNED UTILITY
10 DOWNED TREE
11 BRIDGE PRECAUTION
12 ACCIDENT (MULTI-VEHICLE)
13 DISABLED VEHICLE
14 SLOW VEHICLE
15 VEHICLE FIRE
16 POLICE ACTIVITY

We categorize tweets-based incidents as follows,

1. Accidents (categories 1, 12, 15 in RCRS)
2. Roadwork: roadwork (categories 6, 8, 11 in RCRS)
3. Hazards & Weather: debris, downed trees, downed utility, flooding, winter weather, extreme weather (categories 2, 3, 5, 7, 9, 10 in RCRS)
4. Events: sports, ceremony, parade, etc. (category 4 in RCRS)
5. Obstacle vehicles: slow vehicles, disabled vehicles, stuck trucks, construction vehicles, fire fighters, ambulances, police vehicles and activities (categories 13, 14, 16 in RCRS)

To assign an incident category to a tweet, two data filtering processes will be performed: (1) determine whether or not the tweet is traffic-related; (2) if a tweet is traffic-related, then which category it belongs to, or in what probability the tweet belongs to each of the five categories.

To accomplish the first process, a dictionary (collection) of "keywords" to classify whether or not a tweet is traffic-related will be learned by checking the frequencies of each used word in all tweets and selecting the most relevant words. The selection process

is done using an Active Machine Learning method where the dictionary starts from a set of “seed keywords” and grows iteratively as we learn more tweets.

In all 264 “seed keywords” have been selected as a first step to filter out non-traffic-related tweets. Those “seed keywords” include, but are not limited to:

['CR-,entrance', 'PennDOT', 'DOT', 'I-', 'PA-', 'SH-', 'SR-', 'US-', 'accident', 'accidents', 'alert', 'alerts', 'ambulance', 'approach', 'approaching', 'ave', 'avenue', 'behavior', 'bicycle', 'bicycles', 'bike', 'block', 'blvd', 'boulevard', 'break', 'brg', 'bridge', 'bridges', 'broken', 'bus', 'bus w', 'car', 'caring', 'cars', 'caution', 'civilian', 'clear', 'cleared', 'close', 'closed', 'closing', 'closure', 'coach', 'collide', 'collision', 'commute', 'complaint', 'congest', 'congestion', 'connect', 'connects', 'construct', 'construction', 'crash', 'crash', 'crosswalk', 'crowded', 'curb', 'cycle', 'cycler', 'damage', 'deadly', 'debris', 'delay', 'delayed', 'delays', 'directing traffic', 'disabled vehicle', 'disruption', 'downhill', 'dr', 'drive', 'driver', 'drivers', 'e', 'e', 'east', 'eastbound', 'eb', 'emergency', 'enter', 'enters', 'exit', 'exits', 'fallen tree', 'fast', 'fee', 'feet', 'fine', 'flat', 'flooding', 'friction', 'ft', 'fuel', 'garage', 'hazard', 'headlight', 'highway', 'hill', 'hour', 'hr', 'hwy', 'in', 'in front of', 'inch', 'inches', 'intersection', 'intersection', 'jam', 'jam', 'jammed', 'junction', 'lake', 'lamp', 'lamps', 'lane', 'lanes', 'light', 'lights', 'limited', 'limited', 'lk', 'ln', 'maintenance', 'marker', 'meter', 'meters', 'metro', 'mi', 'mile', 'mile post', 'miles', 'motor', 'mountain', 'move', 'movement', 'mph', 'mt', 'multi vehicle', 'multiple', 'n', 'nb', 'never', 'never move', 'north', 'northbound', 'obstacle', 'on fire', 'outbound', 'overnight', 'park', 'parking lot', 'parking lots', 'passenger', 'passengers', 'pavement', 'peak', 'pedestrian', 'pedestrians', 'period', 'periods', 'pkwy', 'pl', 'place', 'places', 'police', 'ponding', 'puncture', 'rail', 'ramp', 'rate', 'rd', 'remains', 'remove', 'reopen', 'reopened', 'repair', 'report', 'restriction', 'ride', 'river', 'rivers', 'road', 'road work', 'roads', 'roadway', 'roundabout', 'route', 'routes', 'rte', 'rush', 'safe', 'safety', 'safety', 'sb', 'schedule', ‘Schuylkill’, 'seal', 'seat', 'seatbelt', 'sedan', 'sedans', 'segment', 'segments', 'severe', 'severely', 'shocking', 'shoulder', 'sidewalk', 'sign', 'signal', 'signals', 'signs', 'slip', 'slope', 'slow', 'south', 'southbound', 'speed', 'speeding', 'speeds', 'st', 'station', 'stopped', 'stops', 'street', 'streets', 'stuck', 'stuck', 'suv', 'suv', 'terrible', 'ticket', 'tire', 'toll', 'tow', 'towed away', 'tpk', 'traffic', 'train', 'transits', 'transportation', 'truck', 'trucks', 'tunnel', 'tunnels', 'turnpike', 'uphill', 'van', 'vans', 'vehicle', 'vehicles', 'victim', ‘victims’, 'way', 'wb', 'weather', 'weight', 'weights', 'west', 'westbound', 'working zone', 'wreckage', 'wt', 'zone', 'zones']

Notice that “seed keywords” is to be expanded to the final dictionary of “keywords” as more tweets get extracted and labeled.

For the second process, given a tweet is identified as traffic-related, we adopt a similar methodology as the first process to further assign its categories. For each category, the “seed unique keywords” (listed below) include, but are not limited to,

Incident Type	Seed Unique Keywords
Accidents	Crash Accident

	Collision Fatal Tow Break Damage Repair
Road closure	Road work Closure Zone Maintenance Schedule Seal
Hazards & weather	Rain Snow Slip Wind Flood Rainy Snowy Hazard Tree Block Wiper Inches Wet Cold Freeze Hot Visibility Fire Weather Animal Deer Dead Hail Melt Ice Slope Chilly Slick Tire Cover Friction Frozen

	Grip Cloudy Freeze Ponding
Events	Event Marathon Crime Riot Eagles NFL Football Flyers Hockey NHL Phillies Baseball 76ers Basketball Fans Soccer Race Ironman University Concert Demonstration Gang Holiday Exhibition Conference
Obstacle Vehicles	Debris Obstacle Disabled Overweight Tall Height Heavy Stuck

2.2 Case study location

PennDOT and CMU have agreed to use Philadelphia County as the location for the case study. The boundary of the Philadelphia County is shown as **Figure 1**.

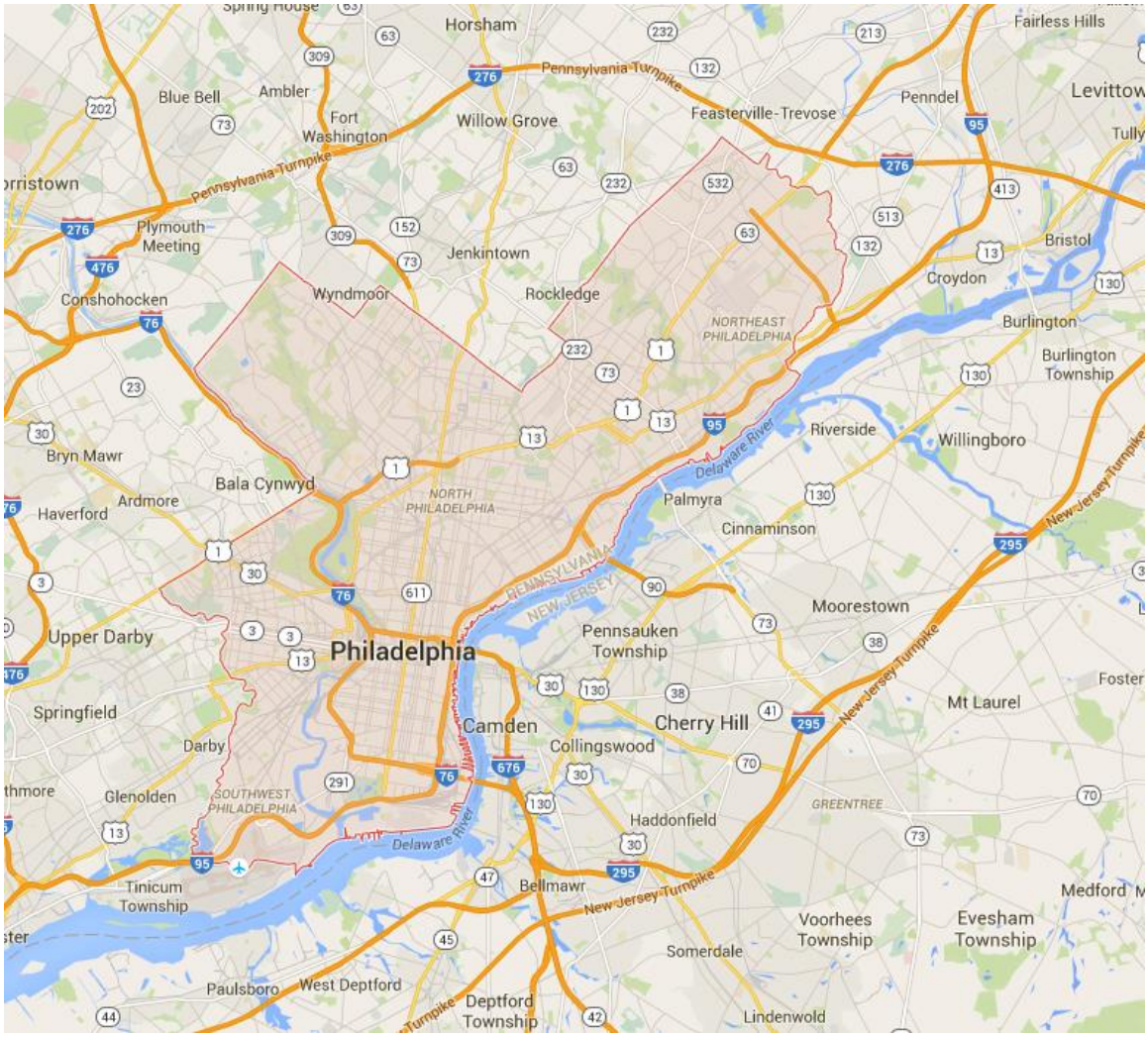


Figure 1. Case study: Philadelphia County (the screenshot was taken from Google Map)

2.3 Twitter data structure

The domain model includes the database design, flow chart for the data analytics, a brief description of each module and software engineering design of the deliverable tool.

2.3.1 Database design

We use the CMU Gardenhouse Twitter data as the historical data set to establish models. The CMU Gardenhouse Twitter data is in raw JSON (JavaScript Object Notation) file. To utilize the highly structured raw JSON file in a more efficient way, the first step is to build a database that could perform efficient query. In this project, a new database called MongoDB is used instead of traditional Relational Databases (RD). The advantages of using MongoDB over RD for this application are as follows:

(1) The schemaless nature of MongoDB allows quick and easy development and any modification could be directly made on the structure of entries;

(2) Naturally support dictionary-like data structures like JSON, which is exactly the data format in the CMU Gardenhouse data;
(3) Rich queries: query on text is not as easy as query on traditional databases, and MongoDB enables a document-based query language that could enforce a rich query procedure;
(4) Supports Big Data: MongoDB is designed to scale for large database applications and natively support Hadoop Map-Reduce, a common big-data machine learning routine.

The object-oriented concepts of the tweets and twitter users in this project is shown in **Figure 2**.

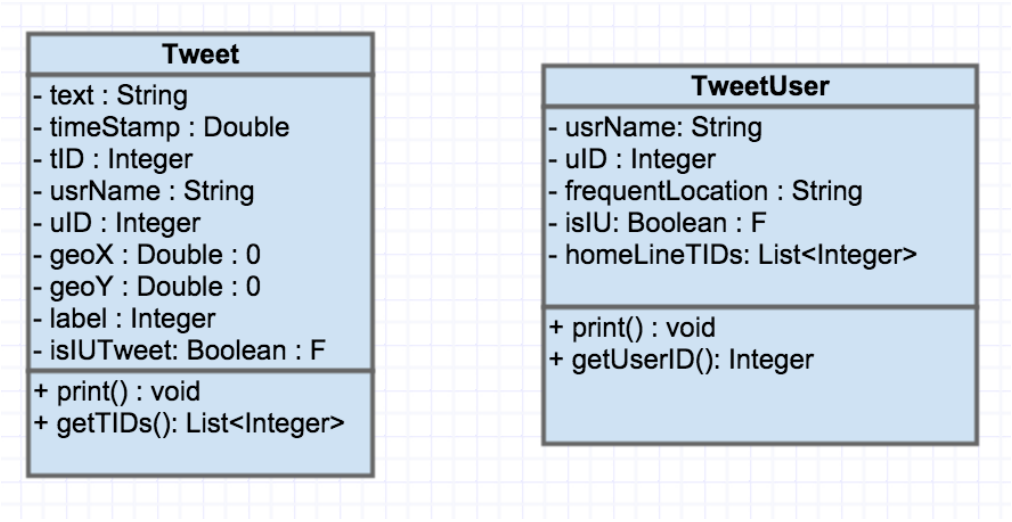


Figure 2. Tweets and Twitter user data structure

The expected output of this project is specified in **Figure 3**.

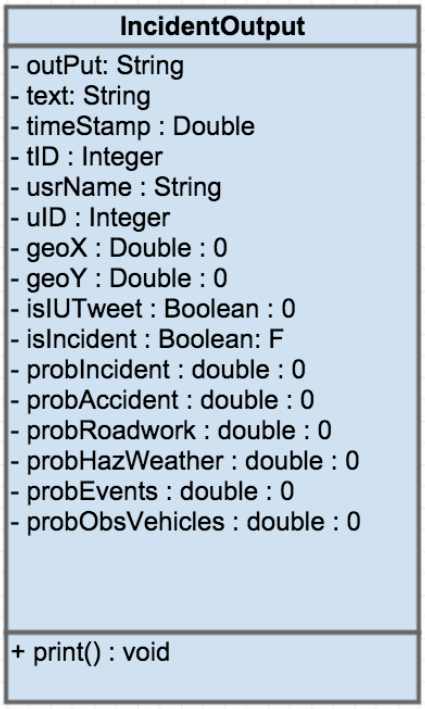


Figure 3. Back-end Incident output – database

Each variable is defined as follows,

- 'timestamp', 'geoX', and 'geoY' – the time and location of the incident;
- 'isIncident' – whether or not the tweet is a traffic-related tweet;
- 'probIncident' – the probability of the tweet being a traffic-related tweet (namely a confidence score);
- 'probAccident' – the probability of the tweet being under the category of an accident;
- 'probRoadwork' – the probability of the tweet being under the category of a roadwork;
- 'probHazWeather' – the probability of the tweet being under the category of hazard & weather;
- 'probEvent' – the probability of the tweet being under the category of an event;
- 'probObsVehicles' – the probability score of the tweet being under the category of obstacle vehicle

2.3.2 Software engineering design

The final deliverable of this project is a prototype map-based website and incident database that acquire tweets and display incidents in the real time. The software design of this prototype system, all the objects of the software with their relations, is described in **11. Appendix: the diagram of software engineering design**.

Under the control of SuperController, the whole Twitter data processing module has three parts: (1) AdaptiveController; (2) Classifier; (3) GeocodingController.

The entire process of adaptive data acquisition is under AdaptiveControler, which firstly crawls an initial batch of tweets using initial queries. The acquired tweets are then processed by the module called LabelAndMapReduce, which assists users to manually label the tweets by whether or not it is traffic-related, and utilizes a map-reduce procedure to generate "important words" which are highly related with traffic. These "important words" are then added into a "traffic incident dictionary" and new queries is assembled from that dictionary to crawl another batch of tweets. This iterative process continues until a convergence criterion is met.

After acquiring all the tweets with the finalized "traffic incident dictionary", a classifier is development and used to determine: (1) whether or not the tweet is traffic-related; (2) what category the tweets belongs to and with what probability. Two possible methods of the classification are Support Vector Machine (SVM) and Naïve Bayes.

To further extract locational information from tweets, we will develop GeocodingController that consists of two parts: Geoparser and Geocoder. Geoparser is to transform the locational information in non-homogeneous tweets into a standard form and the Geocoder is to use the standard form of locational information to perform the geo-coding process in order to obtain the precise longitude and latitude of an incident.

3 The Model for Incident Detection Based on Tweets

To train a reliable model to identify incidents, we adopt CMU Gardenhouse database as our data pool. CMU Gardenhouse database was built and has gathered Twitter data from September 2009 to August 2014 at the Machine Learning Department CMU. Since September 2010, the database contains about 10% of the entire Twitter public messages. The data is stored in internal CMU servers and is accessible via secured HTTP requests. For each day, there are roughly 1 million messages, and around 80% of them are actually tweets (others are non-message events like deletion events). The full database contains about 2170 million tweets and the sizes of the unzipped tweets exceeds 1 terabyte (TB). Thanks to a special agreement shared by Twitter and CMU, the amount of this data (10% of Twitter messages) is massive comparing to the fact that the best of portion of Twitter data could be acquired via public Application Program Interfaces (APIs) is Twitter STREAM API, which contains at most 1% of all Twitter public messages.

Following the domain model designed in **Section 2**, a MongoDB document-oriented database is built to store the 1TB CMU-Gardenhouse historical Twitter data. The main reason of using MongoDB is that it is able to handle the parallel computing of massive amount of data. To better describe the methodologies, we define terms as follows:

- TI tweets: tweets that are related to a traffic incident

- NTI tweets: tweets that are not related to a traffic incident
- Geo tweets: tweets that contain geo-location information (either geo-tagged or have texts indicating geo-locations) and can be geocoded

As shown in **Figure 4**, the model training is in the following steps:

(1) **Initial crawling**: crawl the initial batch of tweets from CMU Gargenhouse historical Twitter database;
(2) **Adaptive data acquisition**: this procedure employs the principle of Active Learning, to recursively crawl tweets using the newly assembled queries by investigating the previously crawled tweets. The goal of the adaptive data acquisition module is to maximize the amount of crawled traffic-related tweets for further classification. This step is further described in the following subsection.
(3) **Feature extraction**: the goal of this module is to map the crawled traffic-related tweets into a proper space where further classification could be performed. A preliminary feature space is the space of all the words in the traffic-related dictionary identified in step 2;
(4) **TI/NTI tweets classification**: given an acquired tweet, the goal of the TI/NTI classification module is to determine whether or not the tweet is related to traffic incidents;
(5) **Categorical classification**: given a TI tweet, the goal of categorical classification is to determine what category (accidents, road work, hazards & weather, events, obstacle vehicles) the tweets belong to;
(6) **Geo-coding**: extract the geo-locational information of a tweet.

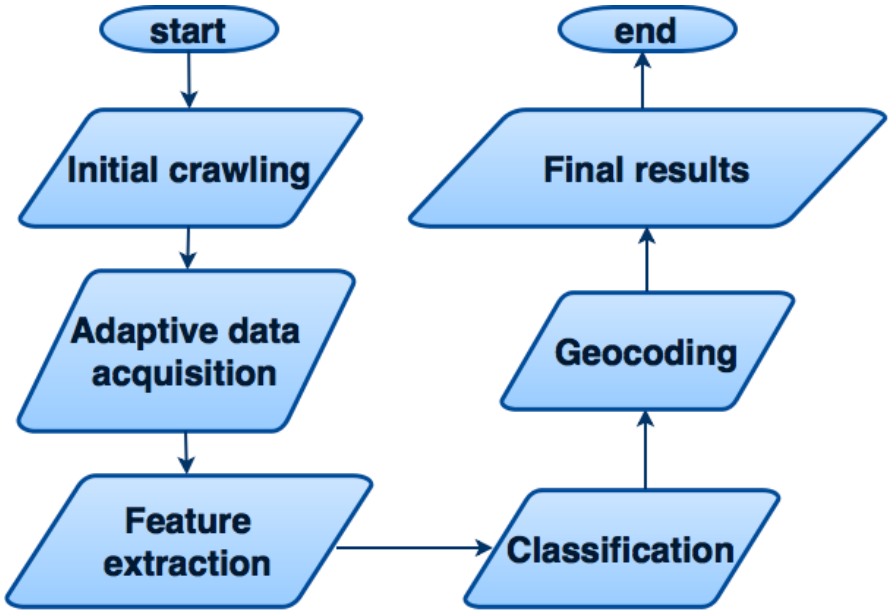


Figure 4. Twitter textual mining flowchart

3.1 Adaptive data acquisition

The objective of adaptive data filtering is to acquire as many TI tweets as possible from the CMU-Gardenhouse database with a reasonable computational efficiency. For each iteration, there are two parts of filtering: (1) Keywords matching for Twitter texts; (2) Keywords matching for user descriptions.

The entire process of iterations is shown in **Figure 5**. First, we assume the Twitter Text Dictionary has the words {A,B,C}, and User Description Dictionary has words {D,E,F}, then the query, in pseudo code, is:

```
For tweet in the data pool:
        If (tweet.text HAS (A OR B OR C)) OR (tweet.usrName.description HAS (D OR E OR  F)):
                Include tweet in the query results
```

The morphological affixes of words are removed from all texts in the query process using Natural Language Tool Kit (NLTK), and also, only English tweets are processed.

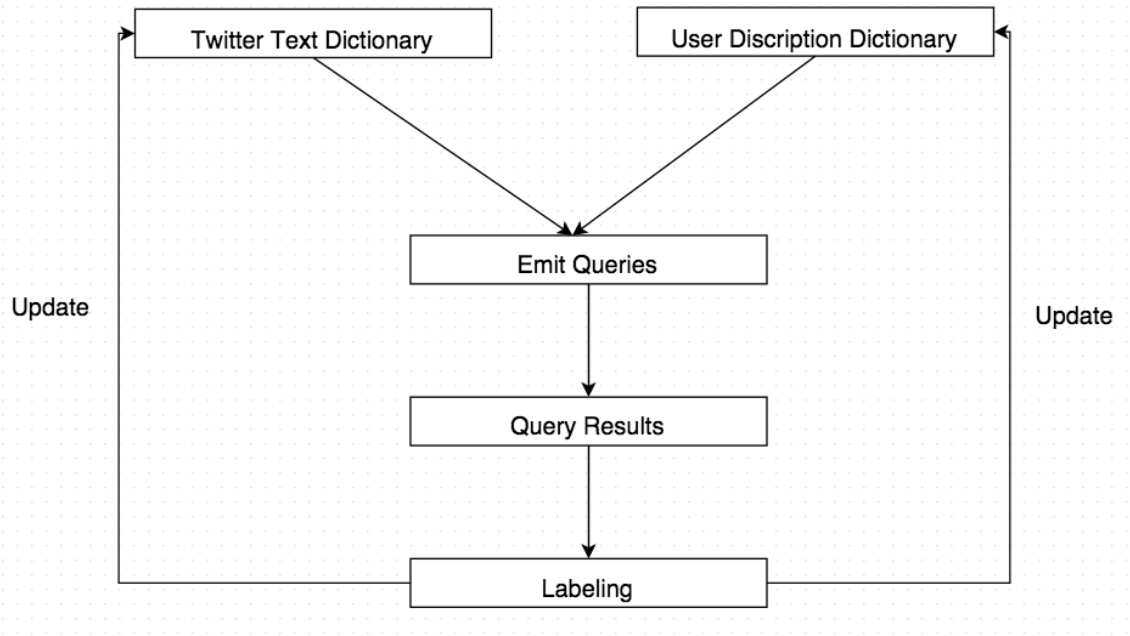


Figure 5. Adaptive data filtering flow chart

Another noteworthy point is that, instead of querying on Influential Users (IUs) directly, we query on users' descriptions. The benefit of this method is to discover new IUs all over the Twitter population without manually browsing for them. Additionally, the results of discovered IUs can be used for queries.

It typically takes several days of processing time to query data from the entire database. We implemented 4 batches of this data acquisition, with 1,093,631 tweets acquired and 19,411 tweets labeled. This results in the expansion of original Twitter Text Dictionary into 203 words.

3.2 Feature extraction and TI/NTI classification

In this project, the commonly used "bag of words" model is applied, meaning only the count of the occurrence of words in Twitter Text Dictionary is used as features, regardless of the order. For example, tweet:

Multi vehicle accident on I-95 southbound at Exit 30 - Cottman Ave/Rhawn St. There is a lane restriction.

The corresponding features are:

vehicle	1
accident	1
#road-name#	3
southbound	1
exit	1
lane	1

restriction	1

Also notice that specific road names like “I-95”, “Cottman Ave”, and “Rhawn St.” are generalized as “#road-name#” because we assume the occurrence of any road name contributes equally to the probability of an TI tweets. This also largely reduces the dimensions of Twitter Text Dictionary.

The model we used for TI/NTI classification is “Semi-Naïve-Bayes”. 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,

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

where Y=1 indicates a tweet is related to traffic incidents and Y=0 otherwise. All features X 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,

$$P(X_i, ...X_j|Y) = \frac{\#\{X_i, ...X_j, Y\}}{\#\{Y\}}$$

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,

$$Y^* = \arg\max_{Y} P(Y|X)$$

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

$$P(Y=1|X) \propto \frac{\#\{Y=1\}}{\#\{\text{"allsample"}\}} \frac{\#\{\text{"work + zone"}=1, Y=1\}}{\#\{Y=1\}} \times \frac{\#\{\text{"pkwy"}=1, Y=1\}}{\#\{Y=1\}} \times \frac{\#\{\text{"delay"}=1, Y=1\}}{\#\{Y=1\}} \times \frac{\#\{\text{"crash"}=0, Y=1\}}{\#\{Y=1\}}$$

$$P(Y=0|X) \propto \frac{\#\{Y=0\}}{\#\{\text{"allsample"}\}} \frac{\#\{\text{"work + zone"}=1, Y=0\}}{\#\{Y=0\}} \times \frac{\#\{\text{"pkwy"}=1, Y=0\}}{\#\{Y=0\}} \times \frac{\#\{\text{"delay"}=1, Y=0\}}{\#\{Y=0\}} \times \frac{\#\{\text{"crash"}=0, Y=0\}}{\#\{Y=0\}}$$

3.3 Categorical classification

The objective of this section is to develop a categorical classifier that can estimate the probability of a tweet belonging to each category given it is related to traffic incidents.

First, as noted in **Section 2**, we define five categories of incidents:

0 => NTI

1 => Accidents

2 => Road work

3 => Hazards & Weather

4 => Events

5 => Obstacle Vehicles

In this research, we use the state-of-art Supervised Latent Dirichlet allocation (sLDA), a generative probabilistic model for collections of discrete data such as text corpora, as our classifier. The reasons of using sLDA are as follows: (1) sLDA is a classical topic model, which is widely used over the years in the domain of Natural Language Processing; (2) it has relatively fast training and prediction runtime; (3) it is able to tell not only the categorical label but also the probability of that label; (4) it is a three-level hierarchical Bayesian model, making the model easy to interpret (comparing to Neural Network or Support Vector Machine). Details of the sLDA model are described in the **Appendix**.

Also notice that the output of the sLDA is a vector of probability of the tweet belonging to each category. The category with highest probability is chosen. Some examples of tweets are:

Kwinana Fwy southbound at Paganoni Rd, Karnup - RIGHT LANE BLOCKED – crash Labeled: {1} Classified: {1=>84% Others=>16%}
CLEARED: Vehicle fire on US 422 eastbound at PA 23.

Labeled: {5} Classified: {4=>89% Others=>11%}
New Fire reported on 91 at Raymond in Anaheim. Both sides of 91 being affected. Take Orangethorpe East\/West as ATL. @KNX1070 traffic guy. Labeled: {4} Classified: {4=>78% Others=>22%}

3.4 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; (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/longitude 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/longitude coordinates. The process of geocoding tweet texts is conceptually depicted in **Figure 6**.

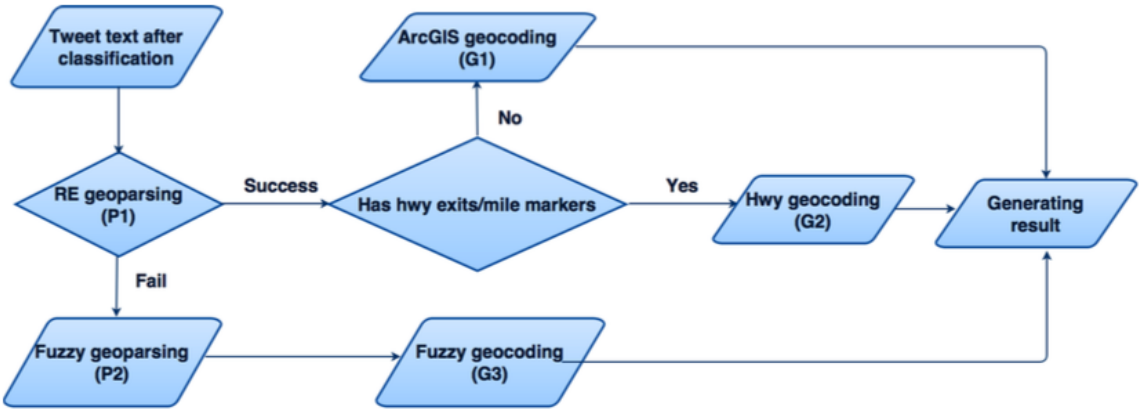


Figure 6. Flow chart of tweets geocoding

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 **Figure 6**, we use two geo-parsers. The first one (named P1 in **Figure 6**) 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 (named P2 in **Figure 6**) 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. 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 1** where 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. 375 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 2**.

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 1. The data structure of RE-based geo-parser

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...

Table 2. An example of geo-parsing result

The output of geo-parsers is then translated to latitude/longitude by geocoders. If a tweet is processed by the fuzzy geo-parser (namely P2), the output words are fed into a Gazetteer (named G3 in **Figure 6)** to identify the location. For the tweets processed by the RE geo-parser (namely 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 (named G1 in **Figure 6**) 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 (named G2 in **Figure 6**) 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.

4 Macro Analysis of Historical Social Media Dataset

4.1 Case I: Sep. 2014, Pittsburgh and Philadelphia

In the month of September, 2014, there are in all 3776 TI tweets acquired for Pittsburgh area and 4571 TI tweets acquired for Philadelphia area using the Twitter REST API and adaptive data acquisition described above. Notice that the dictionary and the IUs are trained by data acquired from REST API, instead of the full CMU-Gardenhouse database. Case I tests how effective our algorithms are when using REST API data to train our model.

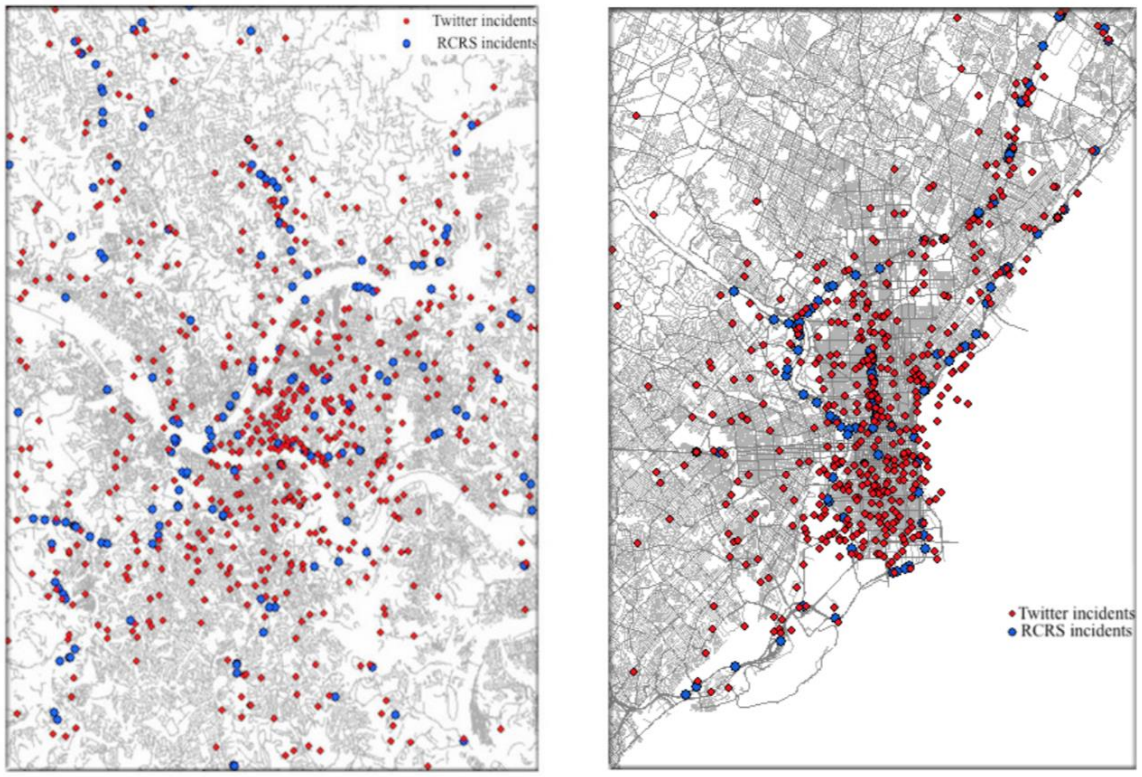
A summary of the data acquisition and geocoding results is shown in **Table 3**. For Pittsburgh, among 10542 tweets, with 3776 TI tweets acquired, only 554 tweets can imply meaningful traffic incidents with accurate time and location. This shows the power of massive data mining on big data. Another noteworthy point is that, although IUs' tweets consists only 5.9% of all acquired TI tweets in Pittsburgh, they contribute to 69.8% of final Geo-TI tweets. The reason of this phenomenon is that IUs' tweets tend to report traffic incidents with detailed location and clear linguistic structure, which makes the accurate geocoding possible. For example, a typical IUs' tweet is "Turnpike Roadwork on Pennsylvania Turnpike I-476 northbound between Exit 31 - PA 63 and Exit 44 - PA 663 420 affecting the right lane", where there is an explicit locational information contains in the tweet. For comparison, a typical individual user's tweet is ""@TMZ: Dog The Bounty Hunter – Daughter & Grandkids In Serious Car Crash http://dlvr.it/6tpycK " @DogBountyHunter @MrsdogC all our prayers", with a picture of the accident. From the vague description in the tweet, it is hard to extract locational information of this traffic accident.

	Pittsburgh	Philadelphia
All tweets acquired	10542	11658
TI tweets	3776	4571
IUs' tweets	621	554
IUs' TI tweets	595	518
Geo-TI-tweets	**554**	**419**
IUs' Geo-TI tweets	381	244
IUs' portion in TI tweets	15.8%	11.3%
IUs' portion in Geo-TI tweets	69.8%	58.2%
RCRS incidents	**217**	**105**

Table 3. Data summary: Case I

After adaptive data acquisition and geocoding, there are 554 Geo-TI tweets left for Pittsburgh and 419 for Philadelphia in September. Each of these Geo-TI tweets can be assumed to describe a traffic incident reported by Twitter. On the other hand, the RCRS reported 217 incidents in Pittsburgh and 105 in Philadelphia. The scatter plots of monthly Twitter and RCRS incidents are shown in **Figure 7a** and **Figure 7b**, where the red dots are incidents reported by Twitter and the blue dots are those reported by RCRS. It can be

seen that there are more red dots (Twitter incidents) than blue dots (RCRS incidents), especially on local arterials. A closer look at those locations implies that many of those incidents reported by Twitter occurred on local roads that are owned by local jurisdiction. The additional incident information can complement the RCRS incidents on state-owned roads.



(a) Pittsburgh RCRS and Twitter incidents

(b) Philadelphia RCRS and Twitter incidents

Figure 7. RCRS and Twitter incidents in Pittsburgh and Philadelphia, Sep 2014

Figure 8 illustrates the difference in terms of temporal distribution between RCRS and Twitter Incidents in the city of Pittsburgh. As can be seen in **Figure 8a**, the RCRS incidents from the first to the last hour of a day is almost evenly distributed. When comparing **Figure 8a** with **Figure 8b**, it can be observed that the temporal distribution of influential users' tweets are extremely similar with the distribution of RCRS incidents though out a day. An explanation is that, the "sources" of RCRS and influential user tweets are quite similar, which are mostly governmental agencies. However, the distribution of individual user tweets is totally different: the individual user tweets are concentrated during the day time and morning/afternoon peak hours. It can be concluded Twitter contains more authority-reported incidents than RCRS, and it have better coverage during morning/afternoon peak hours, but not off-peak. For Philadelphia, the pattern is very similar.

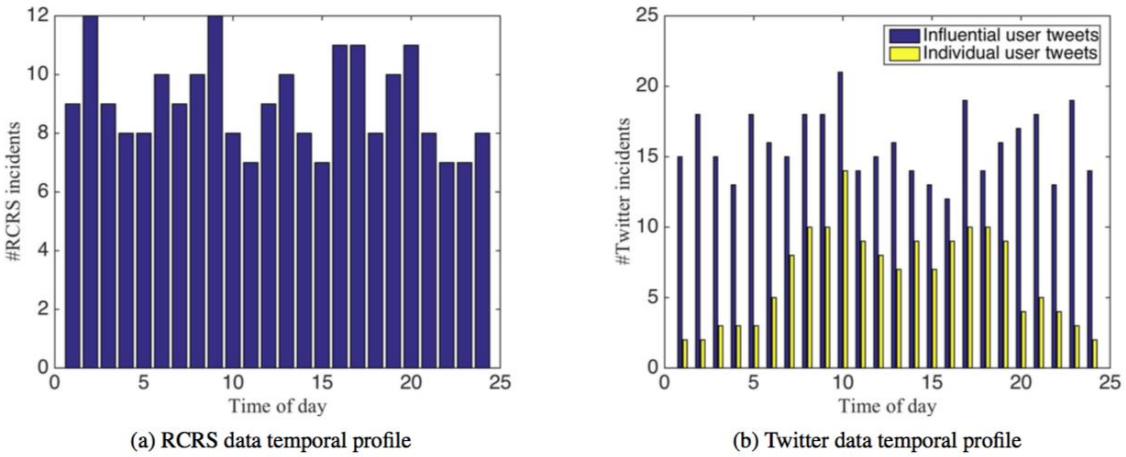
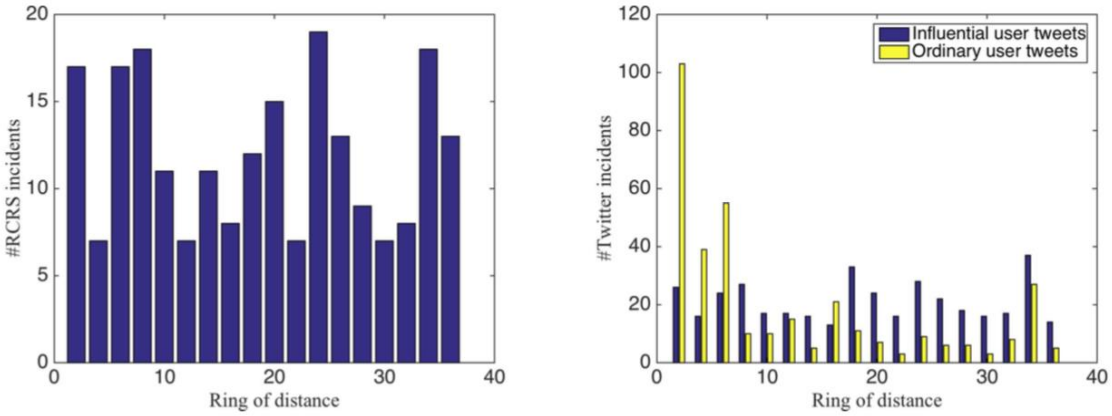


Figure 8. Temporal profiles of incidents reported by RCRS and Twitter in Pittsburgh

Figure 9 shows the spatial distribution of incidents reported by RCRSand Twitter for the greater Pittsburgh area. Each bar in the histogram in **Figure 9** shows the number of incidents falling in the rings of distance to the center of the city, which is chosen according to Google Maps: (40.440731, -79.995751). It can be seen from **Figure 9** that the spatial distribution of RCRS incidents is quite even in the metropolitan area, whereas the Twitter incidents lies more in the center of the city than the suburb area of the city. Another observation is that, in **Figure 9**, the individual user tweets tend to concentrate in the center of the city while influential user tweets tend to be evenly distributed, just as the RCRS data. Therefore, it can be concluded that comparing to RCRS, Twitter incidents tend to have a better coverage on incidents near the center of the city. For the case of Philadelphia, the conclusion remains exactly the same.



(a) RCRS data spacial profile (b) Twitter data spacial profile

Figure 9. Spatial distribution of incidents reported by RCRS and Twitter in Pittsburgh

4.2 Case II: Aug. 2015, Philadelphia

In this case, we conduct the data acquisition using the Twitter REST API and STREAM API, using the trained Twitter Text Dictionary (203 words) and Twitter IUs from CMU Gardenhouse database. In particular, the Classifiers are also trained from the full database of CMU Gardenhouse and the refined Geo-coder are trained from the Open Street Map Points of Interest (POI), Roads, and Areas shapefiles, as well as the Tiger shape files. For those Geo-TI tweets, we also labeled the categories. A summary of the data acquired is shown in **Table 4**.

All tweets acquired	28115
TI tweets	3072
IUs' tweets	2634
IUs' TI tweets	1752
Geo-TI-tweets	**1390**
IUs' Geo-TI tweets	1229
IUs' portion in TI tweets	57.0%
IUs' portion in Geo-TI tweets	88.4%
RCRS incidents	**135**

Table 4. Data summary: Case II

The first noteworthy point is that, there is indeed a large amount of traffic incident information in Twitter, where on average, every 1.5 minutes there is a tweet that could potentially contain incident related information in the Philadelphia area. Additionally, it can be seen that IUs play a very crucial rule in the Geo-TI tweets (88.4%) acquired. Comparing to Case I, training models based on the CMU Gardenhouse database allow us established more IUs that report useful incident information. As of the performance of TI/NTI classifier, the success rate is 92.6%. For the overall performance of TI/NTI

classifier and the sLDA categorical classifier combined, the success rate is 69.2%, meaning the percentage of the actual TI tweets labeled as the correct category is 69.2%.

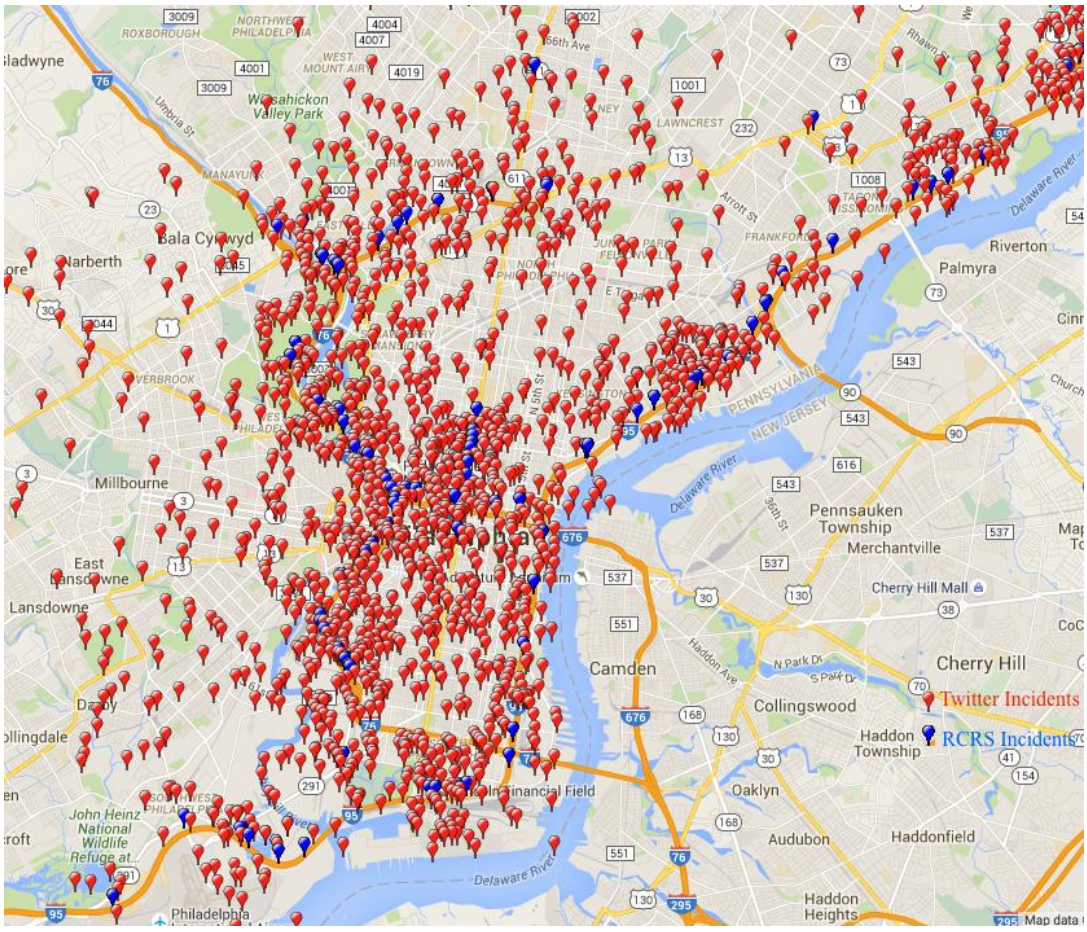
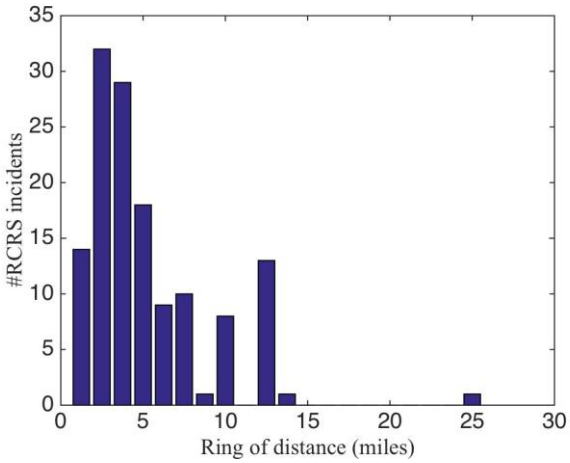


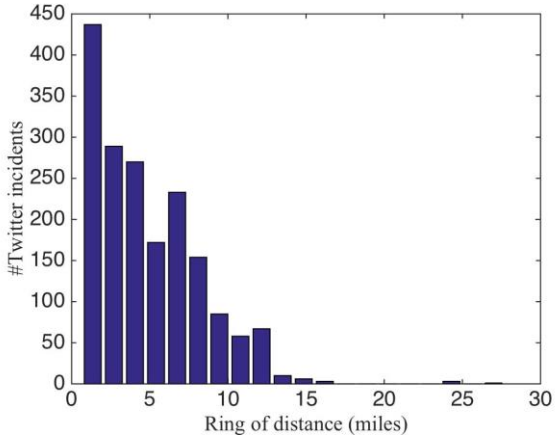
Figure 10. Incidents reported by RCRS and Twitter in Philadelphia, Aug 2015

Figure 10 shows the spatial distribution of incidents reported by Twitter and RCRS: the red dots are Twitter incidents and blue dots are RCRS incident. Clearly, there are much more traffic incidents acquired by Twitter than reported by RCRS. A quantitative analysis is also shown in **Figure 11**. It can be seen that for the city of Philadelphia, RCRS incident tend to concentrate within the 10-mile radius from the city center. However, the number of incidents reported by Twitter is much greater than RCRS, and linearly decreases with respect to the distance from the city center.

Similar to Case I, it is found that many of incidents reported by Twitter occurred on local roads that are owned by local jurisdiction. The additional incident information can complement the RCRS incidents that occurred on state-owned roads. With more incident alerts on local roads, it may help Transportation Management Centers to coordinate between the City and PennDOT for efficient traffic management.



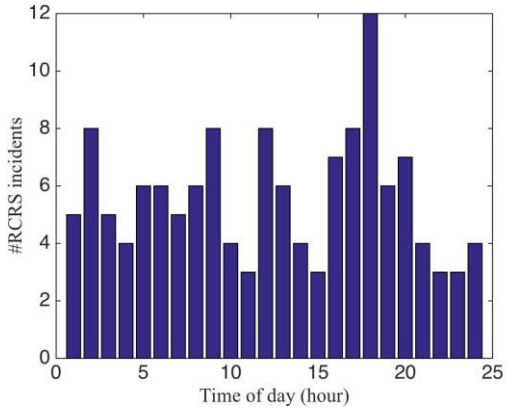
(a) Spatial distribution of incidents reported by RCRS



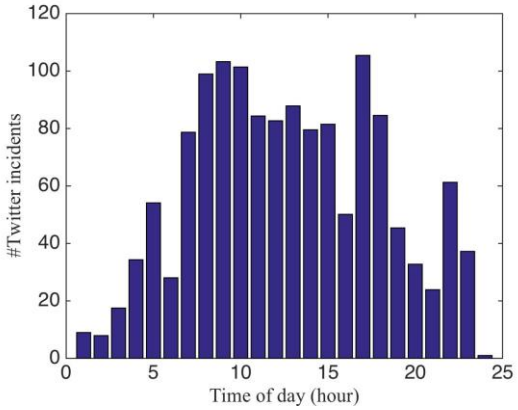
(b) Spatial distribution of incidents reported by Twitter

Figure 11. Spatial distribution of traffic incidents

The temporal distribution of RCRS and Twitter reported incidents is shown in **Figure 12.** Similar to the conclusion drawn from Case I, RCRS incidents tend to distribute uniformly over the time of day, whereas Twitter has a better coverage during the daytime, especially the two peak periods.

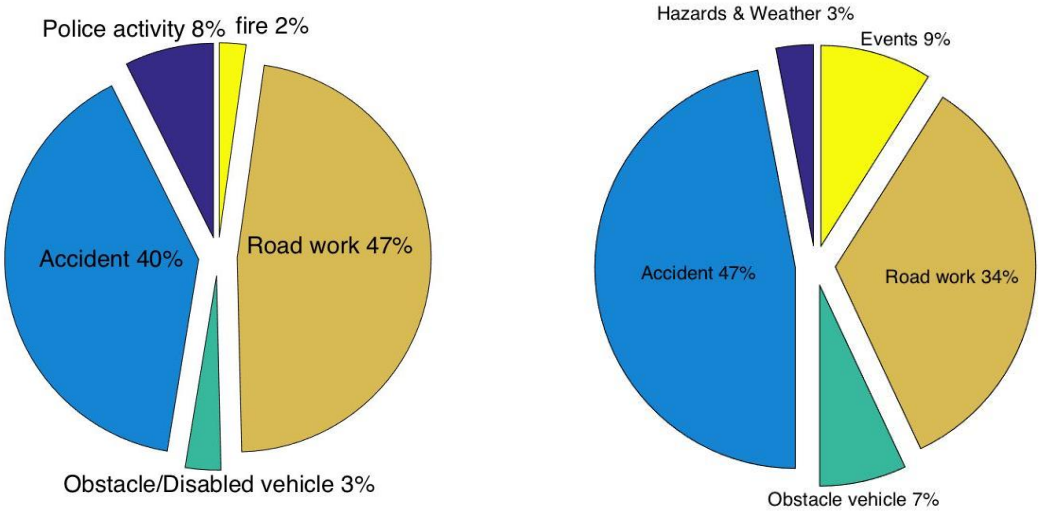


(a) Temporal distribution of incidents reported by RCRS



(b) Temporal distribution of incidents reported by Twitter

Figure 12. Temporal distribution of traffic incidents



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

Figure 13. Incident categories of Aug. 2015 traffic incidents

Figure 13 shows the composition of incident categories between RCRS and Twitter incidents. Twitter tends to report slightly more accidents and road work than RCRS. Moreover, the least portions of incidents in RCRS are obstacle/disabled vehicle, police activity, and fire in RCRS (in all 10%), which correspond to the category of events and obstacle vehicle in Twitter (in all 16%).

5 Real-time Micro Analysis of Social Media Dataset

5.1 The pipeline framework

In Section 3, we have already trained a series of models, namely a Semi-Naïve-Bayes (SNB) Classifier to determine whether or not a tweet is related to a traffic incident (TI tweet), a Supervised Latent Dirichlet Allocation Model to determine which category a traffic incident belongs to, and a geocoder which extracts the locational information indicated by a tweet. To extend the models trained using historical data in Section 4 to a real-time model, we applied a series of real-time algorithms and connect them using the Unix Pipeline technique.

Unix Pipeline is a series of processes chained by standard streams. Due to the need of the real-time data processing, the streaming type of data structure is suitable to generate final results one by one in stream similar to a production line. The code block looks like the below.

```
$ python crawling.py runRT | featureExtract.py feature.p | \
> snbClassfication.py snbmodel.p | geocoding.py google | slda.py sldamodel.p
```

The crawling.py module initiates streams and queries on keywords and Influential Users (IUs) every a few minutes and returns the streaming output to the Real-time Feature Extractor module. Similarly, the output stream of the feature extractor is the input stream

of the Semi-Naïve-Bayes classifier. The major difference between stream-type of process and traditional batch-type of process is that the downstream module does not need to wait for the upstream module to finish all the jobs before it can start working. The streaming technique enables modules to run almost simultaneously on different CPU cores, which could improve the timeliness of the Twitter-based incident detector.

Based on the pipeline framework, the entire real-time models can be shown in **Figure 14**.

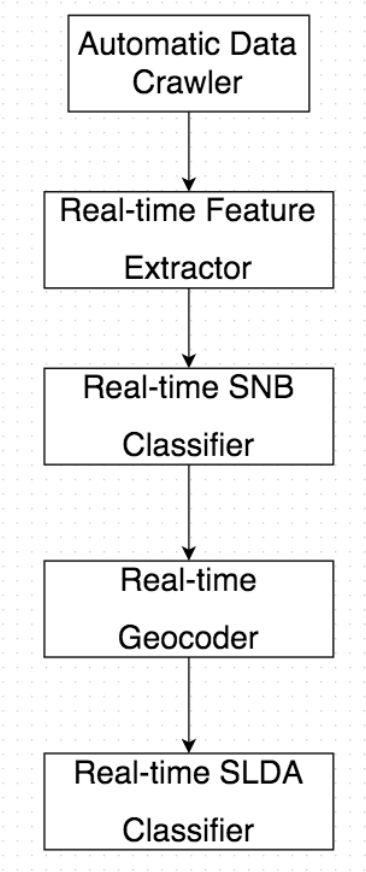


Figure 14. Structure of the real-time Twitter-based incident detector

It can be seen that the real-time version of the Twitter-based incident detector is very similar to the offline version introduced in Section 3, except the fact that the models trained by historical data are implemented and updated in real-time.

5.2 From historical data to a real-time classifier

We follow the framework shown in **Figure 15** to learn the information contained in historical data and apply the algorithms in real-time. Specifically, we used the well-trained and tested model parameter in the stage of offline training and apply these parameters directly into online training. Therefore, since there is no training stage in the online training, the processing speed in the online classification and geocoding will be extremely fast.

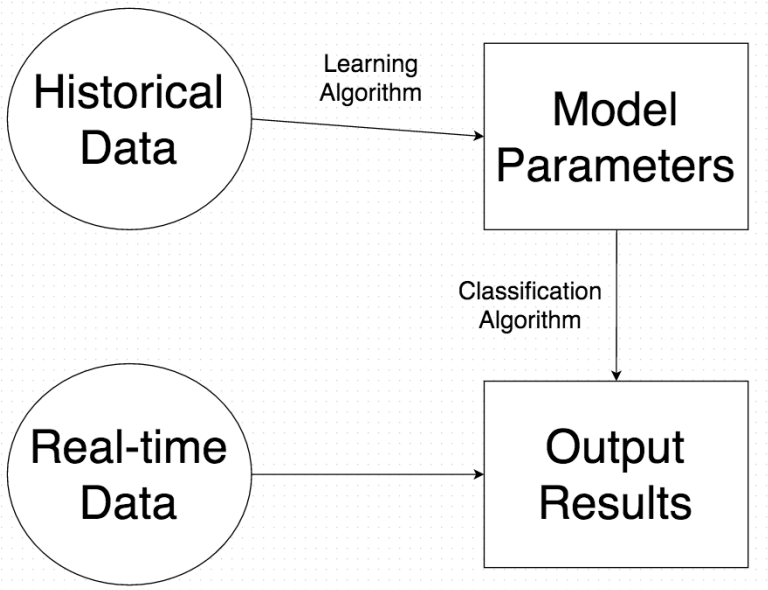


Figure 15. Apply models trained by historical data in the real time

5.3 Test on a week-long data in Philadelphia

The objective of this section is to: (1) test the performance of our real-time Twitter-based incident detector in terms of accuracy and timeliness; (2) further validate whether or not the tweets actually reports traffic incidents using INRIX travel time data and statistical hypothesis test.

We continuously ran the real-time pipeline framework described above for one week (Dec 1 22:00pm – Dec 7 22:00pm). In **Section 4**, we use the Semi-Naïve Bayes classifier, the Supervised Latent Dirichlet Allocation classifier, and the geocoders trained by historical data. The Twitter crawling techniques including all IUs and keywords are the same as what was used in **Section 4**. However, in the real time processing, the tweets are classified and geocoded as soon as they are crawled (acquired) using the Unix pipeline technique, one distinction from historical data processing in **Section 4**. In addition, the final result of **Section 4** was compared to historical RCRS data, whereas the final result of this section is now compared to the real-time INRIX travel time data.

Instead of using RCRS incident data as the reference, we use INRIX travel time data to validate the results of Twitter-based incident detector. The reason is that RCRS does not necessarily cover all incidents on the state-owned roads. For those incidents that are not reported by RCRS, we do not have the ground truth. While INRIX travel time data measure the traffic speed on the 5 min basis for years along, it may be used to infer the occurrence of an incident in some cases. In addition, when processing those acquired tweets, we intentionally excluded the tweets acquired from those official PennDOT accounts (namely, @511PAPhilly, @PennDOTNews). The reason is that we would like to explore the "extra" information that can be provided to PennDOT apart from what PennDOT has already known.

We obtained 190 incidents reported by Twitter from December 1, 2015 to December 7, 2015 that are additional to what PennDOT had known. The details about the data acquired are shown in **Table 5**.

	Statistics
Tweets acquired (excluding PennDOT tweets)	2709
TI tweets	1178
IU's TI tweets	132
Geo-TI-tweets	190
IU's Geo-TI-tweets	124
IU's portion of Geo-TI tweets	65.3%

Table 5. Results for the week-long real-time experiment

According to **Table 5**, about 43% of our acquired tweets are relevant to incidents. Note that this does not mean 43% of all the tweets contain incident information, because we have used the specially trained keywords and influential users to crawl data. In the real world, the percentage of tweets that report incidents is very small. In addition, 16% of the TI tweets can be geocoded.

To establish a filter system for the incident detector, we use the probabilities as scores. For the TI/NTI classification, the score is the probability P(Y|X) generated by the Semi-Naïve-Bayes classifier. Y is the indicator of the whether or not a tweet is a TI tweet, and X is the tweet text. Therefore, P(Y=1|X) means the probability of the given tweet X being a TI tweet, and P(Y=0|X) is the probability of the given tweet X being an NTI tweet. Notice that P(Y=1|X)+ P(Y=0|X)=1. We define:

$$\begin{cases} \mathrm{P}(\mathrm{Y}=1|\mathrm{X}) > \mathrm{P}(\mathrm{Y}=0|\mathrm{X}) & X\ is\ TI\ tweet \\ \mathrm{P}(\mathrm{Y}=0|\mathrm{X}) > \mathrm{P}(\mathrm{Y}=1|\mathrm{X}) & X\ is\ not\ TI\ tweet \end{cases}$$

The formula above means that if the probability of a tweet being a TI tweet is greater than its probability of being a NTI tweet, then we say the tweet is classified as a TI tweet. P(Y=1|X) is precisely the confidence level of an incident detection given a tweet X. A histogram of P(Y=1|X) of the 190 Geo-TI tweet detected is shown in **Figure 16**. It can be seen that the majority of the probability scores lie between 0.7 and 0.9.

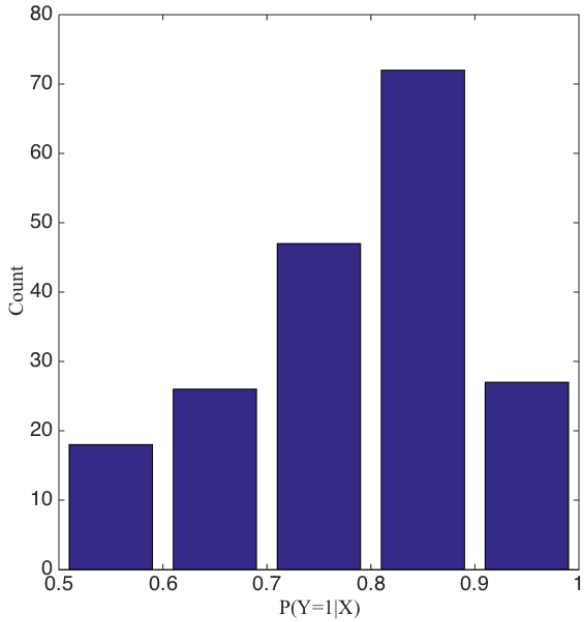


Figure 16. Histogram of P(Y=1|X)

Additionally, we analyzed the timeliness of the entire real-time algorithms. As shown in **Figure 17**, the total computation time (T) needed from the actual occurrence of an incident to the incident displayed on the web is composed of the following seven times indicated in **Figure 17**.

$$T=T1+T2+T3+T4+T5+T6+T7$$

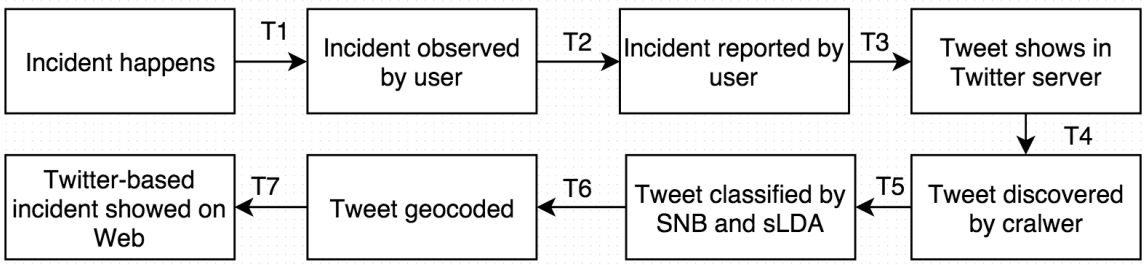


Figure 17. The timeline of an incident being reported by Twitter

Among those time intervals, T1, T2 and T3 are irrelevant to our system. We did some experiments for T5 through T7. T5+T6+T7 is usually less than 5 seconds. The breakdown for T4, T5, T6, and T7 is recorded by our program precisely (**Table 6**).

Time period	Average time for one tweet
T4	It is bounded by crawling frequency (300 seconds on current setting, or user defined crawling frequency)
T5	3.1ms

T6	87ms
T7	0.9ms

Table 6. Breakdown of computation time

It can be concluded that T5, T6, and T7 are almost negligible comparing to T4. The crawling frequency, set to 300 seconds by default, dominates the time from a TI tweet is posted to the corresponding incident is shown in the web application.

5.3.1 Validation using INRIX travel time data

INRIX travel time data is provided by RITIS and could be downloaded by registered users in the Vehicle Probe Project Suite. It is a data set containing the travel speed, travel time data on major roads in the City of Philadelphia with certain confidence levels. The primary assumption of our analysis is that, "if there is a traffic incident on a road, the travel time will substantially vary from the typical mean 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 same time-of-day, 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 test on: (1) each incident that are reported by Twitter; and (2) all incidents reported by Twitter together.

5.3.1.1 Hypothesis test on the entire set of incidents

Suppose the measured travel time of the road segments that are close to the occurrence time and location of the i-th traffic incident is T_i. In particular, we define T_i as the average travel time from half an hour before the incident occurrence and half an hour after. Also, we retrieve all the travel times at the same location, the same time of day, and the same day of week over the previous eight weeks as $\boldsymbol{H_i}$, which is a vector of eight elements (namely real numbers). Notice that $\boldsymbol{H_i}$ is also one-hour average travel time. We standardize the travel time by:

$$T'_i = \frac{T_i - E(\boldsymbol{H_i})}{Std(\boldsymbol{H_i})}$$

and

$$\boldsymbol{H'_i} = \frac{H_i - E(\boldsymbol{H_i})}{Std(\boldsymbol{H_i})}$$

Notice that $\boldsymbol{H'_i}$ is also a vector consisting historical travel times in the previous eight weeks, and $E()$ is the operator of expectation and $Std()$ is the operator of standard deviation. Theoretically, the distribution of $\boldsymbol{H'_i}$ for all Twitter incidents shows the "typical travel time". Comparatively, the distribution of T_i implies the "actual travel time" at the time and location that the i-th incident occurs. By comparing the distribution of T'_i and $\boldsymbol{H'_i}$, we are able to show how statistically different between the typical travel time and actual travel time. The result is shown in **Figure 18**.

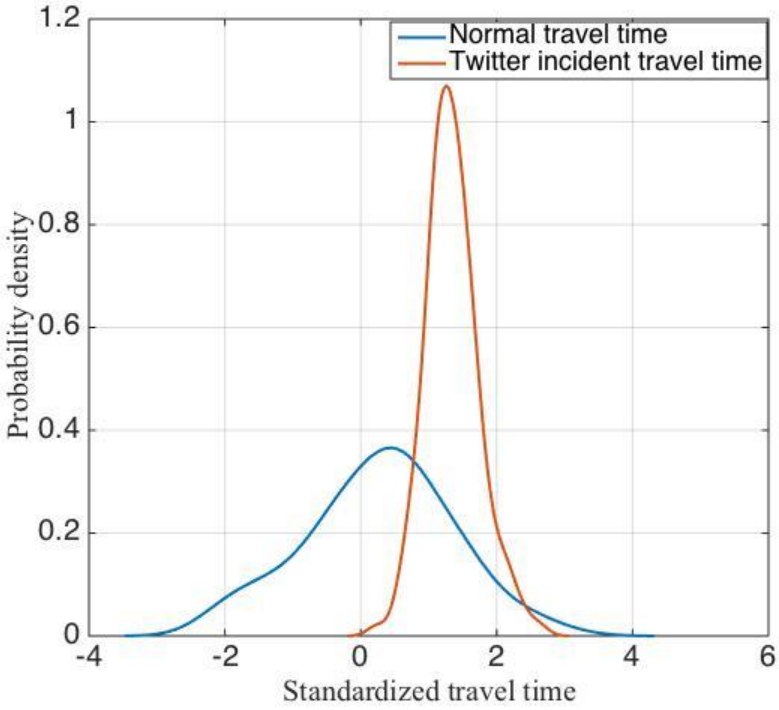


Figure 18. **Comparison on typical travel time and actual travel time**

In **Figure 18**, it can be clearly seen that the distribution of the actual travel time at the time and location where Twitter reports an incident is significantly different from the distribution of the typical travel time at the same time and location. Moreover, we performed a Kolmogorov–Smirnov (K-S) hypothesis test under the null assumption that "Typical travel time and actual travel time have the same distribution".

The P-value of this K-S test is 1.3277e-26, which is significantly smaller than normal significance level 0.01, meaning we should reject the null hypothesis. Therefore, we can conclude that "there is significant evidence for two random variables: typical travel time, and actual travel time when Twitter indicates an incident, follow two different distributions". The actual travel time is significantly higher than the typical travel time, which validates that those Twitter-reported incidents are likely to be true.

5.3.1.2 Hypothesis test on individual incidents

The K-S hypothesis test above gives the overall validation that the generally Twitter-reported incidents are truly traffic incidents. For each individual Twitter-reported incident, we perform a specific hypothesis test with the null hypothesis "Typical travel time and actual travel time where Twitter reports an incident follow the same **Gaussian** distribution". The test in general is a Z-test. If the test statistic shows that we reject the null hypothesis, we can conclude that "there is significant evidence to show that when Twitter reports an incident, the traffic is likely to be abnormal due to the incident. To

perform this Z-test, we prepare the data in the following way. First, similar to Section 3.2.1, we compute

$$T'_i = \frac{T_i - E(\boldsymbol{H_i})}{Std(\boldsymbol{H_i})}$$

and

$$\boldsymbol{H'_i} = \frac{H_i - E(\boldsymbol{H_i})}{Std(\boldsymbol{H_i})}$$

Notice that in this test, $\boldsymbol{H_i}$ is all the travel times at the same location, same time of day, and same day of week as T_i in the previous eight weeks. As assumed in the null hypothesis, $\boldsymbol{H_i}$ follows a Gaussian distribution, and therefore, $\boldsymbol{H'_i}$ conform a standard Gaussian distribution with mean 0 and variance 1. The P-value of our statistical test is

$$PValue = P(Z > T'_i)$$

where Z is the standard Gaussian random variable. The Z-test is shown in **Figure 19**, where the area to the right of the black line T'_i and under the blue curve is the P-value of interest for the i-th incident detected by Twitter.

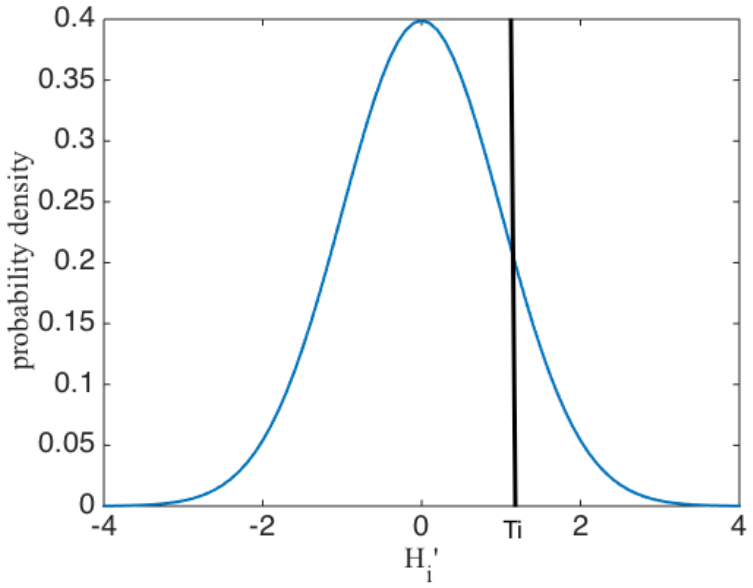


Figure 19. Standard Z test for individual sample

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

$$\begin{cases} \text{PValue} > \text{U} & \textit{Fail to reject the null hypothesis} \\ \text{PValue} < \text{U} & \textit{Reject the null hypothesis} \end{cases}$$

Also notice that "rejecting the null hypothesis" means that the Twitter-based incident corresponding to T'_i is truly a traffic incident. Here instead of defining one single

threshold U, we explore how U can influence the percentage of Twitter-reported incidents that reject the null hypothesis (namely being truly an incident). As we can see from **Figure 20** that when the threshold of U is set to be around 0.25, almost all the sample travel times T'_i will reject the null hypothesis, in other words, their corresponding Twitter-reported incidents are actual.

Overall, by comparing the travel time at the same time and location when an incident reported by Twitter occurs to that of previous eight weeks, we conclude that statistically those Twitter-based incidents are likely to be true.

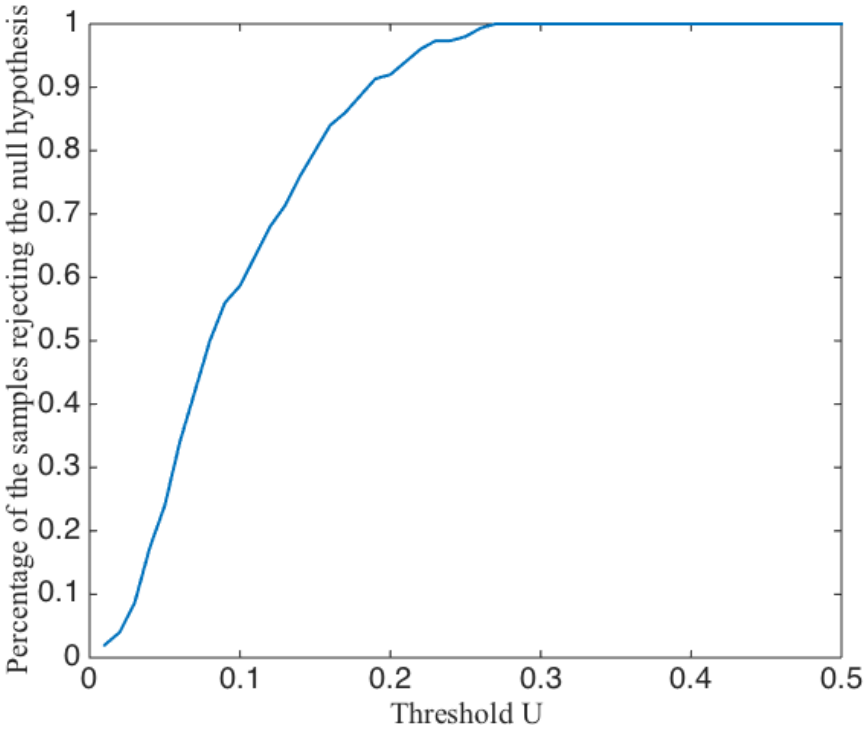


Figure 20. **The influence of U on the rate of the samples rejecting the null hypothesis**

6 Audio-Based Incident Detection

6.1 Methodology

6.2 Data Description

Due to the data availability, in this section, we only consider commercial stations, instead of some highway advisory radio (HAR)-specific stations (e.g., HAR376). We considered commercial audio stations from both Pittsburgh and Philadelphia. For Pittsburgh, we collected stream data from three audio stations: KDKA 1020, KQV 1410 and WXDY 107.9. For Philadelphia, we collected stream data from two audio stations: KYW 1060

and WXPN 88.5. All the five audio stations report traffic news regularly. Specifically, KDKA reports traffic-related news every 10 minutes during 5-9am and 3-7pm, seven days a week; KQV reports news every 10 minutes during 6-9am, seven days a week; WXPY reports the same news as KDKA does. KYW reports news every 10 minutes during 3-6pm, seven days a week; WXPN reports news every hour during 5-10am, on weekdays only.

One challenge of crawling and analyzing those station stream data is that traffic news only take up a very small fraction (time) of the entire data. We identify the time of day periods for each audio station when traffic news will be reported. In this way, audio data in those periods only will be extracted automatically using a program. Analyzing this subset of data is much more computationally efficient. The time of day periods when traffic news is reported is referred to as "traffic news pattern" in the remainder of this section.

All the raw stream data were collected from one online source: http://tunein.com/. For Pittsburgh, we crawled data from August 25, 2015 to October 20, 2015. And for Philadelphia, we crawled data from September 2, 2015 to October 20, 2015.

Then we follow the information given by Pittsburgh Highways (http://pittsburgh.pahighways.com/trafficinfo.html) and extract the traffic news pattern for each stations. For Philadelphia, we learned the pattern by manually listening to the audio streams. Based on the patterns information, we divided the stream data into two parts, with or without traffic news. Notice that the patterns we learned may vary by minutes from day to day, we define each sub-stream data with traffic news as a two-minute stream around the possible reporting time points. For example, given that KDKA reports news at :02, :12, :22, :32, :42, :52, we define sub-stream data with traffic news as the stream :01-:03, :11-:13, :21-:23, :31-:33, :41-:43, :51-:53. Detailed patterns are shown in **Table 7**.

Station	City	News time in each hour	Hours per day	Days per week
KDKA 1020	Pittsburgh	:00; :10; :20; :30; :40; :50	5-9am, 3-7pm	7 days
KQV 1410	Pittsburgh	:08, :18, :28, :38, :48, :58	All day	7 days
KQYW 1060	Philadelphia	:02; :12; :22; :32; :42; :52	All day	7 days

Table 7. Traffic News Patterns in Each Audio Station

To record the audio streams from online sources, we used a python package, *timeshift*. To extract the traffic-related streams, we used another python package, *pymp3cut*, to slice the MP3 recordings. The lengths of audio data with traffic news for this study are as follows: 6,486 minutes for KDKA, 6,902 minutes for KQV, and 9,686 minutes for KYW.

6.2.1 Audio to text

The reason we divided stream data as described above is that we need to parse audio stream data to transcripts with relatively high accuracy. However, most of existing tools

of speech recognition are not satisfactory because our stream data may be noisy and irregular.

In order to convert the audio streams to texts, we tried several tools but few of them performed satisfying results. For example, we've tried CMU *Sphinx*, which is an open source speech recognition toolkit. This tool requires the adaptive acoustic models to obtain a relatively high translation accuracy. However, the existing acoustic models are hard to deal with the high noise in the traffic-news-related streams. In addition to CMU Sphinx, we also test several python-based packages (e.g., Speech Recognition). These packages cooperate free-version speech recognition tools, such as Google Speech, IBM Watson Speech to Text, etc. However, the free-version tools cannot identify the traffic-news speeches, which have features such as fast speed, noisy background, and multiple-people interaction.

After several trials, we decided to apply a commercial tool: VoiceBase. For each account, the first 50 hours' data for transcription are free in VoiceBase. After that, it charges $0.01 per minute. We chose the Machine Transcription option. This option is cheaper than manual transcription, with reasonable accuracy sufficient for our study.

6.2.2 Feature extraction, classification, and geocoding on the audio scripts

Following the similar methodology of processing twitter data, we developed the algorithms of processing audio script as shown in **Figure 21**. It can be seen that after transforming the audio data into audio scripts, we treat these sentences analogous to tweets and use similar models trained by Twitter data to process classification and geocoding.

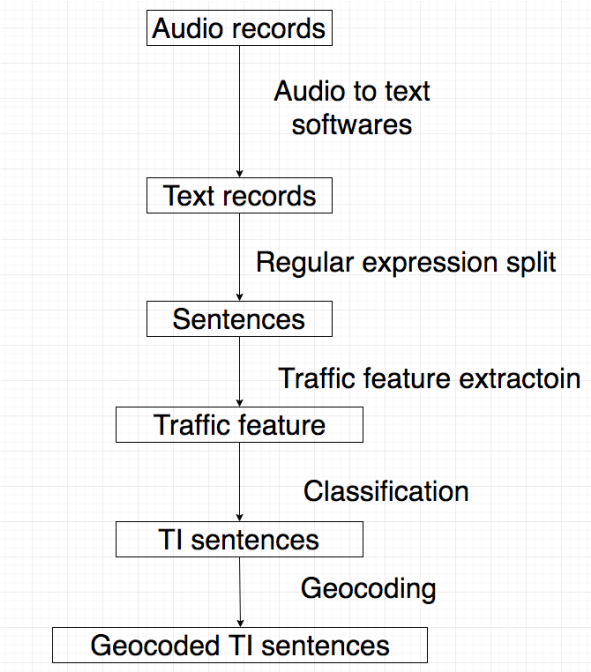


Figure 21. Audio flowchart (TI: traffic incident)

6.3 Results

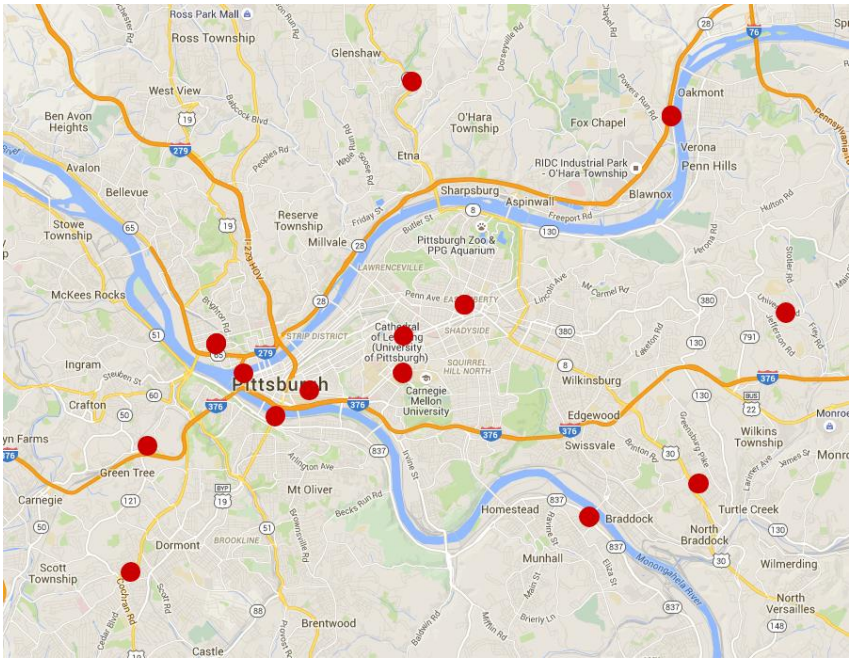
The data processing results are shown in **Table 8**. It can be seen from **Table 8** that although there are a very large amount of sentences from the audio records that could potentially lead to information about traffic incidents, the actual number of sentences that can be classified and geocoded is very few. First, there are a large amount of errors when translating audio data into texts. Oftentimes the translated text is hard to understand, even for humans. Road names in the scripts can have errors, or are sometimes ambiguous. It is hard to geocode using those road names. Second, the classifier developed specifically for Twitter data is not particularly suitable for audio transcripts because the "traffic dictionary" used in the Twitter model and the semi-Naive-Bayes classifier trained by the Twitter data may be biased.

Data	Description
Audio records	Two months of audio
Text records	18.8Mb of text
Sentences	113,802
TI sentences	1,223 (1% of all sentences)
Geocoded TI sentences	40 (3% of all TI sentences)

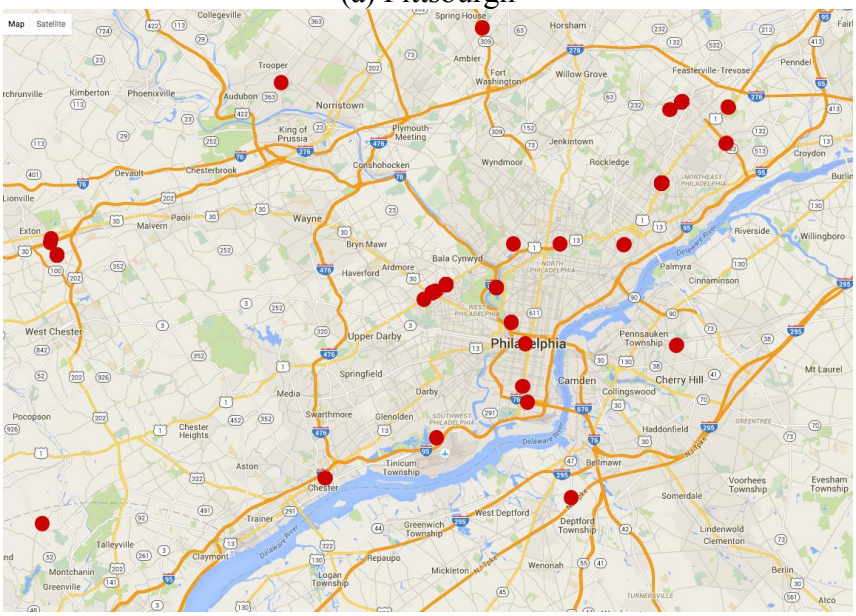
Table 8. Data processing results

The scatter plots of geocoded TI (traffic-incident-related) sentences are shown in **Figure 22**. It can be seen that the audio-reported traffic incidents are mainly distributed on highway roads. Regardless of the small quantity of the geocoded TI sentences, the quality of these audio-based traffic incidents is surprisingly good. The manual inspection shows

that only one of these 40 audio-based traffic incidents is false, an extremely high true positive rate.



(a) Pittsburgh



(b) Philadelphia

Figure 22. Audio-reported traffic incidents in the City of Pittsburgh and Philadelphia

To further validate these audio-based traffic incidents, we compare the audio-based incidents with RCRS data. Here we examine both audio-based incidents and RCRS traffic incidents from September 1, 2015 to October 20, 2015 for both cities: Pittsburgh and Philadelphia. We define the audio-data "coverage rate" as R_a:

$$R_a = \frac{A(r_t, r_s)}{total\ number\ of\ audio\_based\ incidents}$$

where $A(r_t, r_s)$ is the number of RCRS traffic incidents that lie within a temporal radius r_t, and a spatial radius r_s, of all audio-based incidents. The value of R_a reflects the portion of audio-based incidents that is already covered by RCRS. The variation of R_a with respect to r_t *and* r_s is shown in **Figure 23**. It can be seen from **Figure 23** that if the incidents taking place within 60 minutes temporal radius and 1 mile spatial radius can be assumed to be the same incident as in RCRS, the R_a is 100%, meaning all of the audio-based incidents is already covered by RCRS.

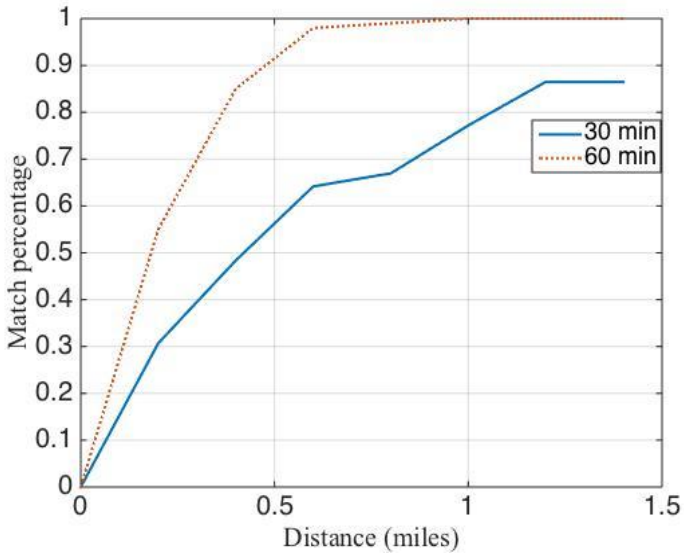


Figure 23. The coverage rate of audio-based incidents

6.4 Discussions

In this section, audio data are processed to extract traffic-related information. The audio data is first transformed into text transcripts using audio-to-text software. Then the sentences of the transcripts are further classified into being a traffic incident related or not. Those traffic incident related sentences are geocoded. Finally, some traffic incidents, mostly on highways, can be identified. Comparing the audio-based incidents with RCRS data shows that though the result of identifying audio-reported incidents is accurate, the information extracted from the audio data can be covered mostly by RCRS.

This work has a few limitations:

(1) The audio stations we are concerned about here are the major public stations in each city. However, some other stations, such as the highway advisory radio streams (e.g., HAR376), are not included in this project. The reason is that we could only crawl stream data from online publicly-available sources, whereas the

HAR376 is not available online. Future studies may consider such additional streams to extract additional useful information.

(2) In order to convert the audio stream to text, we utilized the non-free online source (i.e., VoiceBase). Although it is the best we could find so far, it still contains certain noisy (or even wrong) transcriptions (especially in terms of the street names). To further validate the data, we may need to manually check the data or use some supervised learning techniques to train a better language model.

(3) The classifier trained by Twitter data may not be particularly suitable for processing audio transcripts. This can be improved by intensively training the model with years of audio data and ground truth.

(4) Due to the above limitations that calls for additional efforts for processing noisy audio data and the fact that most audio data reproduces RCRS, we therefore do not suggest further explore the potential of audio data in real-time incident detection.

7 A Prototype Web Application for Twitter-based incident detection

We developed a prototype web application to allow user to extract incident information and visualize it on the map. The web application consists of three components: a control panel on the left, an interactive map on the middle-right, and an interactive table on the bottom. The layout of the control panel is shown in **Figure 24**. Through the control panel, the users can

(1) query historical Twitter incidents (if there are any) via the ”QUERY” button along with users’ choice of time periods;

(2) visualize real-time Twitter incident via the ”UPDATE REAL-TIME” button;

(3) reset the frequency of the database crawling from Twitter APIs;

(4) input and remove the keyword of the underlying ”traffic dictionary” and/or any Influential Users included in the real-time data crawling.

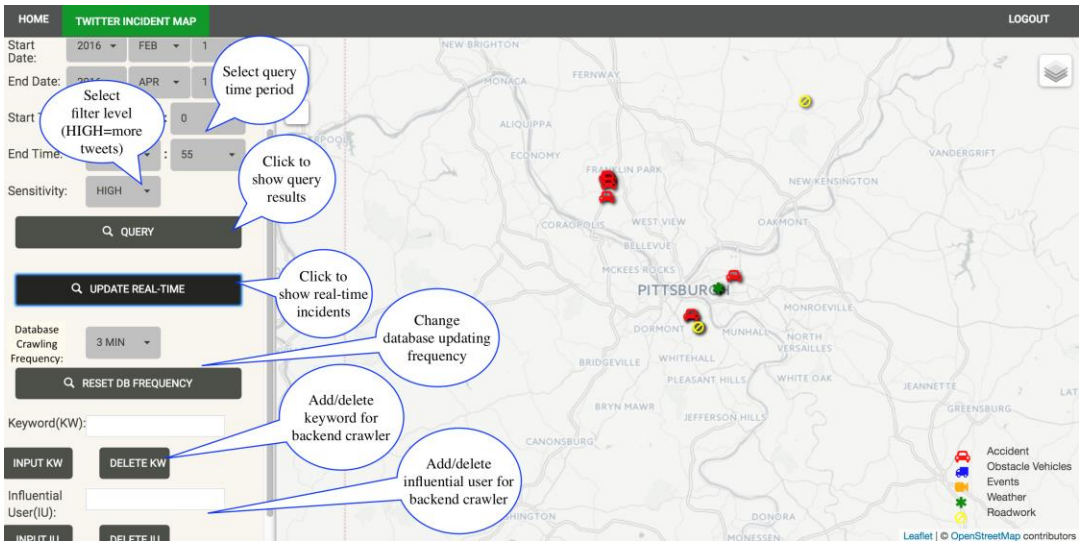


Figure 24. Control panel of the web application

The "Sensitivity" dropdown box has three options, (high, medium and low), indicating the confidence level of detected incidents. Option "high" extracts and visualizes only those incidents that are highly likely to be true reported by tweets, which can sometime omit those tweets that report a true incident with, however, a low confidence level due to our data mining model. On the other hand, option "low" extract and visualizes all incidents that are potentially reported by tweets, even though many may not contain incident information and are noises. Option "medium" provides a compromise between accuracy and coverage, and is the default value of the web application.

The layout of the interactive data table is shown **Figure 25**. The contents of the table are descending by time with the most recent tweets listed on top, up to the past 12 hours. Every entry in the data table has a corresponding icon visualized on the map. To zoom to the icon/location on the map, the user can click on the button 'Show on map'.

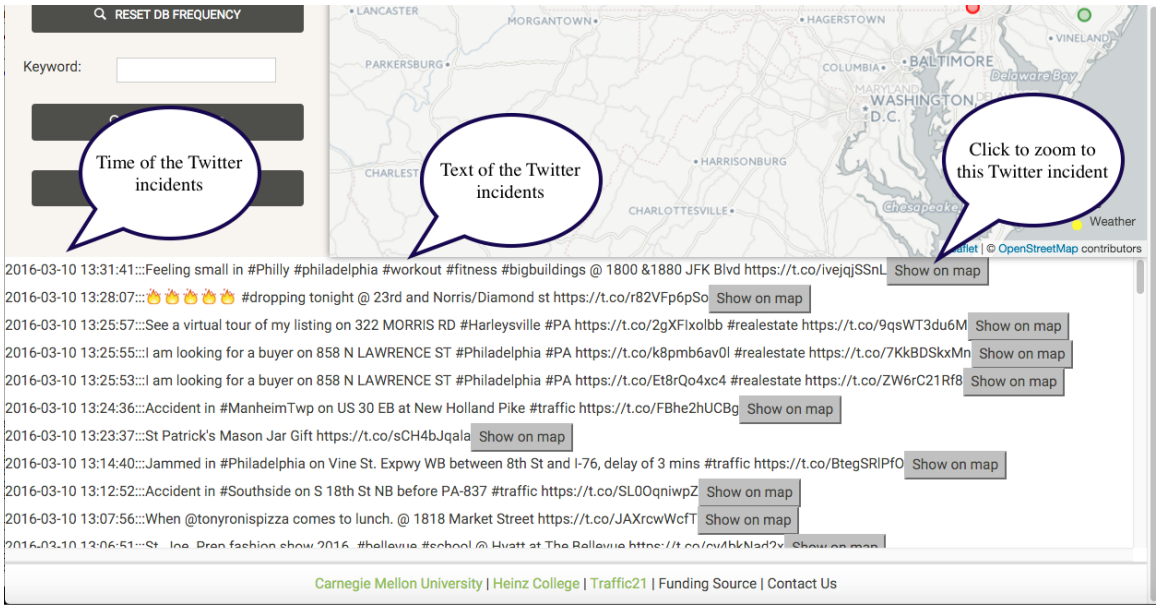


Figure 25. Data table layout

The layout of the pop-up is shown in **Figure 26**. The popup is activated when the user left click on the icon on the map. In this pop-up, the details about the Twitter incident, Twitter user name/ID, texts, time, and classified category is shown. If the users find the incident category is incorrect, they can manually submit a new label to the web application. The web server will update the labels and correct the data mining model periodically.

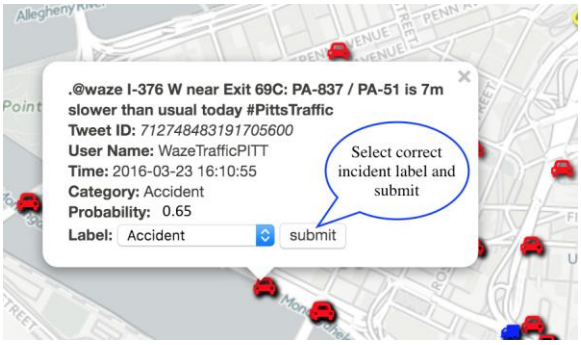


Figure 26. Pop-up layout

8 Conclusions

We apply Natural Language Processing algorithms on the massive database of CMU-Gardenhouse historical Twitter database to train a data mining model that performs accurate classifications on tweets for incident detection. To test the performance of these algorithms, tweets have been crawled from Twitter-REST-API and Twitter STREAM API. By extensively mining the massive data of CMU Gardenhouse Twitter database, we applied Naïve Bayes and Supervised Latent Dirichlet Allocation on the task to classify whether or not a tweet is incident related, and to which incident categories it belongs. Additionally, we developed a series of geo-parser and geocoder to extract locational information inside of Twitter texts. To further test the model, we applied the model to the real-world data in Sep. 2014 for both Philadelphia and Pittsburgh areas, and in Aug. 2015 for Philadelphia area.

Answers to questions proposed in Section 1 are summarized as follows:

(1) How frequently do Twitter users in selected region and corridor tweet about incidents?

A: In the Philadelphia area, Aug 2015, there are 1390 tweets that contain information about the time, category, and location of traffic incidents. Note that these tweets are extracted from a small portion of total tweets that are public accessible from REST API.

(2) What types of incidents do they tweet about?

A: They tweet mostly about accidents and road work, and sometimes special events.

(3) Are there any locations about which people tweet more often?

A: People are more likely to tweet about incidents near the center of the city. The number of incidents reported by Twitter is much greater than RCRS, and linearly decreases outwards with respect to the distance from the city center.

(4) What is the ratio of overall Twitter data in selected region and corridor to data that is relevant to incidents?

A: About 10% of our acquired tweets in the historical dataset were relevant to incidents. Note that this does not mean 10% of the tweets contain incident information, because we have used adaptive data acquisition to crawl as many TI tweets as possible. In the real world, the percentage of tweets that report incident is very small.

(5) Are there any particular times (of day/year) or conditions for which users tweet more about incidents?

A: People tend to tweet more often on weekends and during the daytime instead of night time, especially during the rush hours.

(6) Can social network analysis techniques identify key influencers who tweet about incidents?

A: Yes, by mining the full CMU-Gardenhouse database, we have identified more than 40 influential users who routinely report incidents via tweets which accounts for 88% of all incidents reported by tweets. These users are used in the real-time data acquisition process.

To extend offline version of the Twitter-based incident detector into an online version, we applied a series of improvements and modifications on the offline Twitter-based incident detector to make it more efficient under a standard Unix Pipeline framework. Specifically, we successfully answered the following questions,

(7) How to identify in real time if a tweet is incident related?

A: We used the Semi-Naïve-Bayes classifier trained in **Section 4**, and found it can achieve similar performance in the real time.

(8) How to classify events in all incidents related tweets in the real time?

A: We used the Supervised Latent Dirichlet Allocation (sLDA) classifier trained in **Section 4**, and found it can achieve similar performance in the real time.

(9) How to infer the geo-location of the tweet and map it to the road network?

A: We used the geo-parser and geo-coder trained in **Section 4**, and found it can achieve similar performance in the real time.

(10) What percentage of incidents can be detected and what percentage of those can be precisely geo-coded?

A: In the real time, about 43% of our acquired tweets were relevant to incidents. Note that this does not mean 43% of all the tweets contain incident information, because we have used the specially trained keywords and influential users to crawl real-time tweets. In the real world, the percentage of tweets that report incidents is very small. In addition, 16% of the TI tweets could be geocoded.

(11) How timely is the Twitter-based incident detector?

A: There are several unknown components in the Twitter-based incident detector, such as the time from a tweet is posted to the time it can be actually crawled from the APIs.

Given a tweet can be acquired through the APIs, the processing time for classification and geocoding is minimal and negligible.

(12) How to establish a score system to indicate confidence levels of valid incident detection?

A: The score is the probability P(Y|X) generated by the Semi-Naïve-Bayes classifier, discussed in **Section 3**. Y is the indicator of the whether or not a tweet is a TI tweet, and X is the tweet text.

9 References

Analytics, P. (2009), 'Twitter study–august 2009', San Antonio, TX: Pear Analytics. Available at: www.pearana-lytics.com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009.pdf.

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

10 Appendix: Supervised Latent Dirichlet Allocation (sLDA)

To formalize the notation, we define here:

- Word: an item from Traffic Dictionary indexed by {1,...,V}. We represent words using unit-basis vectors that have a single component equal to one and all other components equal to zero.
- Tweet: a sequence of N words.
- Corpus: a collection of M tweets.

sLDA assumes the following generative process for each tweet w in a corpus D:

1. Choose N~Poission(ζ)
2. Choose θ ~Dir(α)
3. Choose φ ~Dir(β)
4. For each of the N words w_n:
   (a) Choose a topic z_n~Multi(θ)
   (b) Choose a word w_n from p(w_n |z_n, φ), a multinomial probability conditioned on the topic z_n and prior φ.

where Poission(ζ) is a Poisson distribution with parameter ζ, Dir(α) is a Dirichlet distribution with parameter α, Multi(θ) is a Multinomial distribution with parameter θ, and p(w_n |z_n, φ), is a conditional distribution of w_n.
The generative process described above is shown in **Figure 27**.

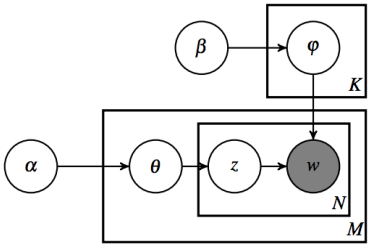


Figure 27. sLDA plate model

11 Appendix: the diagram of software engineering design