
# Transportation Sentiment Analysis for Safety Enhancement

FINAL PROJECT REPORT

Dec 19, 2013

By Feng Chen, Ramayya Krishnan

Technologies for Safe and Efficient Transportation University

Transportation Center (T-SET)

Carnegie Mellon University

CONTRACT No. DTRT12GUTG11

# Table of Contents

CHAPTER 1: Automatic Twitter Data Crawling and Content Filtering ..... 7
1.1 Automatic Twitter Data Crawling ..... 7
1.2 Automatic Content Filtering ..... 9
CHAPTER 2: Geocoding of Transportation and Safety Related Tweets ..... 16
2.1. Fuzzy Location Extraction from User Profile ..... 17
2.2. User Location Extraction Using Social Ties ..... 17
2.3. Location Extraction from Tweet Text ..... 19
2.4. Mile Marker Extraction from Tweet Text ..... 20
2.5. Ensemble Locations Information Using Probabilistic Soft Logic ..... 21
CHAPTER 3: Topic Modeling and Sentiment Analysis ..... 23
3.2. Tweets Categorization ..... 23
3.1. Topic Modeling ..... 23
3.2. Topic Chaining ..... 24
3.3. Sentiment Analysis ..... 25
CHAPTER 4: Implementation of an Interactive Web Interface ..... 27
4.1. Flow of Data Processes ..... 27
4.2. Design of Main Modules ..... 27
4.3. Optimizations on the Server and Client Sides ..... 29
4.3. Design of Web Pages on the Client Slides ..... 30
CHAPTER 5: Conclusion and Future Work ..... 39
Acknowledgments ..... 40
References ..... 40

## List of Figures

Figure 1 Examples of Positive and Negative Tweets ..... 9
Figure 2: Example of Tweet Features ..... 10
Figure 3: Feature Mapping for the Linear SVM Classifier ..... 11
Figure 4: Training of a Linear SVM Classifier ..... 12
Figure 5: Examples of Positive and Negative Labeling by Linear SVM Classifier ..... 12
Figure 6: Interface of Web Interface for Active Tweets Labeling ..... 14
Figure 7: Adjustment of the Linear SVM Classifier Using Additional News Labels ..... 16
Figure 8: Example of Location Mentions in Tweets ..... 19
Figure 9: Word Cloud Visualization of Automatically Discovered Topics ..... 24
Figure 10: Example of Sentiment Analysis ..... 26
Figure 11: Flow of Data Processes ..... 28
Figure 12: Main Modules of the Web Based Prototype System ..... 29
Figure 13: ER Relationships between Web Pages ..... 30
Figure 14: The Dashboard Page ..... 31
Figure 15: The City and State Analytics Page - 1 ..... 32
Figure 16: The City and State Analytics Page - 2 ..... 33
Figure 17: The Topic Analytics Page ..... 34
Figure 18: The Twitter User Analytics Page ..... 35
Figure 19: The Traffic Map Page - 1 ..... 36
Figure 20: The Traffic Map Page - 2 ..... 37
Figure 21: The Emergency Page ..... 38

## Executive Summary

Traffic injuries and fatalities are an enormous public health problem. To reduce transportation related injuries and fatalities, it would be helpful to monitor traffic in real time in order to quickly identify any regions and activities that have the potential to become a risk to public safety. It is clearly impractical to deploy and maintain a large sensor network capable of monitoring every corner of the transportation network, but thanks to the explosion of social media in all forms, including blogs, online forums, Facebook, and Twitter, it should be possible to treat social media as a human sensor network. This would enable us to collect timely and comprehensive information about the current status of the transportation network and traffic flow to support advanced safety enhancement.

The main objective of this project was to develop a real-time Twitter monitoring system to automatically retrieve tweets related to transportation safety, extract the potential safety topics (e.g., traffic accidents, road flooding), calculate public sentiments, and finally visualize the topics and sentiments using word clouds, Open StreetMap, and other graphic tools. The potential users include transportation engineers (e.g., early identification of safety bottlenecks), transportation planners (e.g., adjustment of transportation policies in response to public sentiments and opinions), and public users (e.g., improved routing to avoid potentially unsafe regions). This objective was accomplished through four tasks, where are described in detail in this report. Task 1 (chapter 1) introduces related work for automatic Twitter data crawling and domain-specific content filtering, and presents improved and customized algorithms for transportation safety analysis. Task 2 (chapter 2) provides a literature review of related geocoding techniques for Twitter data and presents a hybrid approach that integrates location information from user profile, user social relationships, and location mentions in tweet texts. The geocoding was implemented at multi-resolutions, including state, city, and street levels, and estimated the quality score for each resolution. Task 3 (chapter 3) presents the procedures that were implemented to automatically discover safety related topics

from the collected Twitter data, study the temporal evolution of these topics, and calculate their sentiment scores. The safety related topics were discovered at multiple geographic (e.g., city, state) and temporal (e.g., hourly, daily) resolutions. Task 4 (chapter 4) presents the implementation of a web-based interactive prototype system that allows users to query and visualize safety information and patterns that were discovered by previous tasks. Particularly, this chapter first presents the design of an optimized file management system that pre-computes and optimizes the storage of intermediate results for support of real-time performance at the client-side interface. It then presents technical details for the implementation of advanced visualization functions, such as word clouds, Open StreetMap, and dynamic query-based charting functions. Finally, it presents the graphic design of major web interfaces.

The overall results show that there are a significant number of tweets discussing or reporting information related to transportation safety. The prototype system was able to retrieve high quality tweets in real time, and to geocode them to streets or even latitude/longitude locations. The web-based interactive interface allows users to quickly view the summary statistics of raw tweets, and to identify potential safety bottlenecks using the advanced topic discovery and sentiment analysis functions. Based on this prototype system, the second phase of the project for the next year period will focus on the development of move advanced machine learning functions for safety enhancement, including traffic accident detection, traffic congestion detection, and safety constrained routing for bikers and pedestrians.

# CHAPTER 1: Automatic Twitter Data Crawling and Content Filtering

## 1.1 Automatic Twitter Data Crawling

In Twitter, there are over 400 million tweets posted every day, but there are no more than 0.1 percent tweets related to transportation. Twitter provides two categories of search APIs that allow users to search and download tweets based on search keywords and geographic constraints [8].

The first category, named as Twitter REST APIs, allows users to submit queries against the indices of recent or popular tweets. The REST query format includes a centroid (latitude, longitude), the radius (e.g., 0.5 mile), and a set of keywords with support of operators, including AND, OR, and EXCLUDE (e.g., "traffic AND (accident OR collision)"). The limits include 1) 3500 total tweets per REST query and 2) 350 queries per 15 minute for one user account.

The second category, named as Twitter streaming APIs, allows users to keep a persistent HPPT connection and crawl up to 1 percent pubic tweets of most recent. The streaming query format includes the combination of a centroid (latitude, longitude) and the radius (e.g., 0.5 mile) or a set of keywords with the operators similar to the above. However, this APIs does not support the joint of location and keywords constraints.

Given the above Twitter search APIs, it is challenging to design appropriate queries in order to retrieve transportation related tweets with a high recall. Here, recall is defined as the rate of transportation tweets that are retrieved over the total amount of transportation tweets in the whole Twitter space. First, there are limits of rates for both categories of APIs, and it is impractical to retrieve 100% of public tweets. Second, the total amount of transportation tweets is unknown, and hence the true recall cannot be calculated.

The parameters that need to be optimized include 1) the number of user accounts that need be registered; 2) a pool of specific queries; and 3) the number of requests for each query in a 15 minute time window. Before we present out strategy to optimize the preceding parameters, we first introduce the sampling strategy designed to measure the approximate recall of a specific combination of parameters:

- Randomly select a set of testing users; crawl tweets from their Twitter home pages; and manually label transportation and safety related tweets. Denote the resulting set of transportation tweets as $Q$.
- Intersect testing users' tweets with tweets crawled based on the specific set of parameters. Denote the result set of tweets as $E$.
- The ratio $|E|/|Q|$ is considered as the approximate estimator of recall. For example, suppose there number of testing users' transportation tweets equal to 1000, and the number of intersected tweets equal to 900. Then the approximate estimator of recall equals to 90 percent.

Three rules are defined for designing good quality search queries:

- Capable to retrieve highly related tweets;
- Capable to retrieve the maximum amount (e.g., 3500 for REST APIs) of tweets;
- Capable to reduce overlaps between different queries.

We first collected more than 1000 words that are related to transportation and safety, such as "accident", "traffic", "vehcile", "bus", "street", "crash", "killed", and "pedestrian". Then we followed the above three rules and designed a large number of queries that could achieve approximately more than 80% recall. Some typical ones are illustrated as follows:

- Q1: "(Traffic OR Car OR Vehicle OR Bus) AND (Accident OR Collision OR Crash)"
- Q2: "(Road OR Highway Or Street) AND (Accident OR Collision OR Crash) AND –Traffic AND –Car AND –Vehicle"
- Q3: "(Slow Traffic) OR (Road Closed) OR (Traffic AND (congestion OR jam OR congested)) AND –Accident"
- Q4: "(Traffic light) OR (Traffic AND light AND (Street OR Highway Or Road Or Freeway)) AND –Accident"
- Q5: "(Cycle OR Cycling OR Cycler OR Biker OR Bike OR Biking) AND (Street OR Highway Or Road Or Freeway OR Lane) AND –Accident"
- Q6: "Pedestrian"

- Q7: “I-99 OR I-97 OR I-66 OR …”

Note that, in addition to the above data crawling from Twitter search APIs, we also collected approximately 10 percent of public tweets from the Gardenhose/Decahose stream.

## 1.2 Automatic Content Filtering

After we have retrieved a collection of tweets that have a high recall with respect to transportation tweets, we observed that the precision is still very lower, less than 10 percent. It is therefore necessary to design an advanced classifier to identify transportation tweets, and filter out unrelated tweets (noises). Denote transportation tweets as positive tweets and the other tweets as negative tweets. Figure 1 illustrates some positive and negative tweets. The positive tweets are marked using red-colored rectangles.

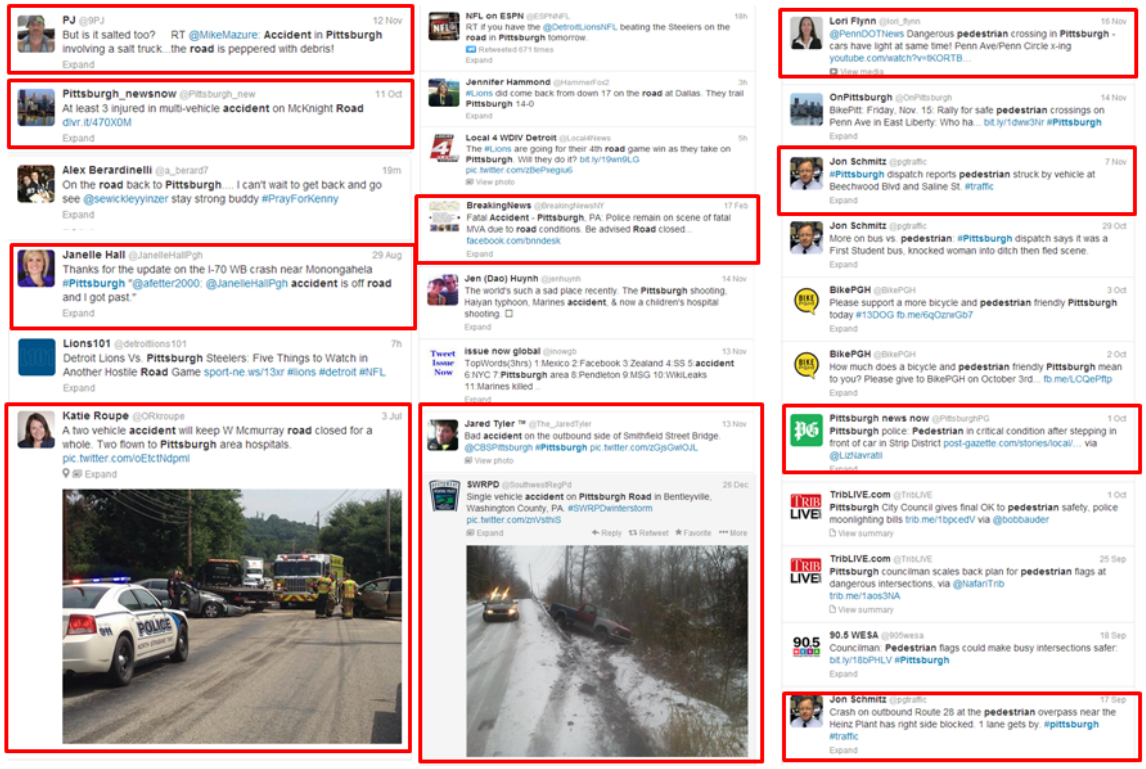


**Figure 1 Examples of Positive and Negative Tweets**

The design and implementation of the classifier model is described as the following steps:

**Step 1: Feature Extraction**

Given a specific tweet, we filtered out the stop words and extract the following features, including 1) unigram; 2) bigram; 3) trigram; 4) total number of words; 5) existence of foreign countries (YES or NO); 6) maximum length of words; 7) sentiment score; 8) emoticons; 9) existence of road names (YES or NO); and 10) existence of organizations (YES or NO). A bigram refers to every sequence of two adjacent words, and a trigram refers to that of three adjacent words [9]. For the above features, we further filter out those with frequencies smaller than 15 that is a popularly used threshold in Twitter data analysis.


**Figure 2: Example of Tweet Features**

**Step 2: Training of a Linear Support Vector Machine (SVM) Classifier**

After feature extraction, each tweet is now represented as a high dimensional feature vector. The tweets are all mapped to the feature vector space as shown in Figure 3. It has been well studied that linear discriminative classifiers, such as logistic regression and linear support vector machine (SVM), are the best-performed classifier models for texts. In this project, we consider the linear SVM classifier for the task of content filtering. The training of the linear SVM classifier is basically to search for a linear separating hyper plane that separates the objects of different classes, such that the margin (shaded area in Figure 4) can be maximized with the consideration of penalties due to misclassified objects. A simplified example of the classifier can be represented using the linear

function “f(tweet) = 0.6 * Accident + 0.1 * Slow + 0.3 * Traffic – 0.7 * Soccer – 0.9 * Grandfather – 0.6 * Pittsburgh”. Given a specific tweet, if f(tweet) > 0, then it indicates that the tweet is positive (transportation related); otherwise, the tweet is negative. Figure 5 shows one positive example tweet and one negative example tweet.

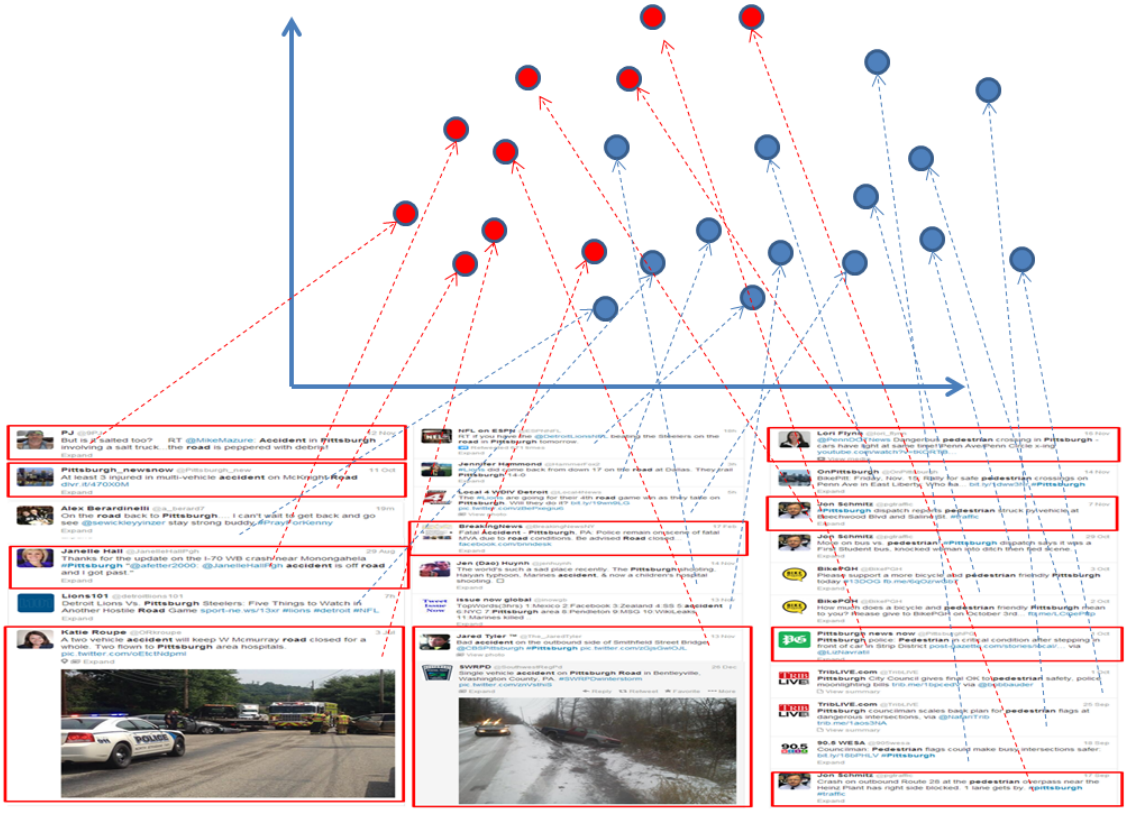


**Figure 3: Feature Mapping for the Linear SVM Classifier**

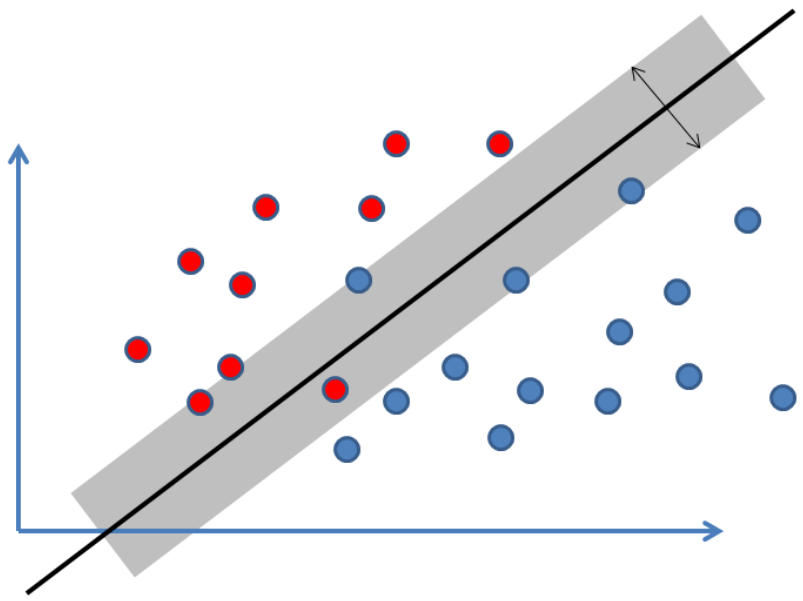
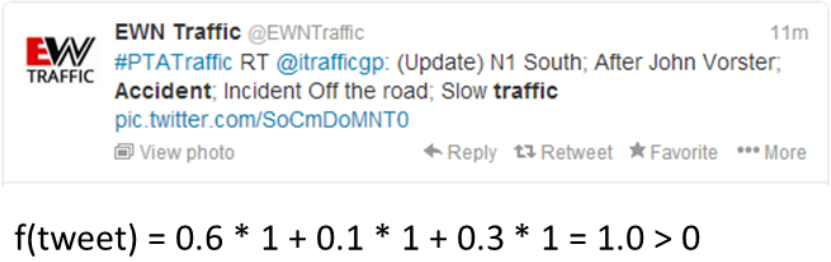


**Figure 4: Training of a Linear SVM Classifier**




$$f(tweet) = 0.6 * 1 + 0.1 * 1 + 0.3 * 1 = 1.0 > 0$$

a) Example of positive tweet


$$f(tweet) = 0.6 * 1 - 0.9 * 3 = -0.1 < 0$$

b) Example of negative tweet

**Figure 5: Examples of Positive and Negative Labeling by Linear SVM Classifier**

**Step 3: Design of Interactive User Interface for Tweets Labeling and Training**

In order to train a well-performed linear SVM classifier, it is required to label sufficient positive and negative tweets. We observed that tweets are short texts, which are different

from traditional documents such as news reports. The feature of shortness leads to the serious problem of data sparsity. The number of non-zero features is far much smaller than the number of non-zero features in a traditional document. For this reason, the required size of sufficient training data is much larger than that for traditional documents. In order to reduce the labeling cost and speed up the labeling process, we designed a web based interactive interface as shown in Figure 6 that supposes active learning. The web interface displays each tweet as a row, with predicted class label shown on the second column (PRED) using the currently trained linear SVM classifier, and the labeling status (positive label, negative label, or not labeled) on the third column. The meaning of PRED color is defined in Figure 6. If the current linear SVM classifier predicts this tweet as positive (True) and the tweet is labeled as positive as well, then PRED will display black color. If the current linear SVM classifier predicts this tweet as positive (True) and the tweet is labeled as negative (False) as well, then PRED will display black red. If the current linear SVM classifier predicts this tweet as positive (True) and the tweet is not manually labeled, then PRED will display green color. Other combinations are defined similarly. The web interface also provides the button to retrain a linear SVM classifier based on the current updates of labels. Therefore, this web interface allows users to iterative label new tweets and to check the performance of the linear SVM classifier.

Another important feature of the web interface is that it allows users to filter tweets based on keywords. This feature enables users to actively label tweets based on the current quality of the classifier trained. For example, if we observe that the classifier performed poorly on tweets related to traffic lights, then we can filter out tweets based on keywords related traffic lights, and specifically label more tweets related this this topic. By using this active labeling interface, we significantly reduced the labeling cost, and were able to train a high-quality linear SVM classifier with a set of training labels that is much smaller than the size of required training data without using active labeling.

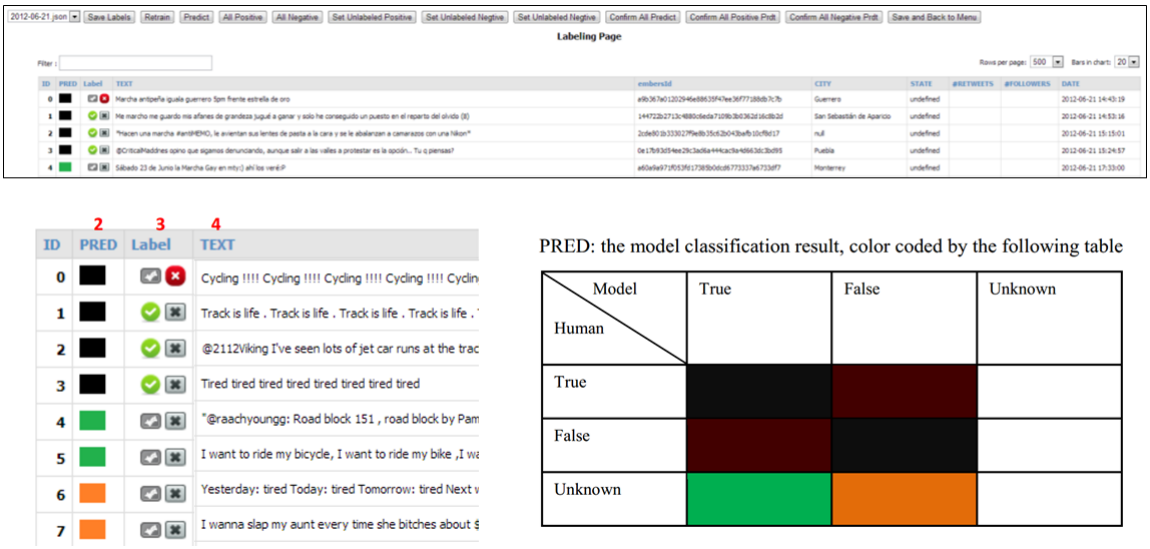



**Figure 6: Interface of Web Interface for Active Tweets Labeling**

**Step 4: Transfer Labels from News Reports to Tweets**

In order to collect more labels, we also explored the usage of news reports and web articles. There are a number of websites that have specific subsections focusing on transportation safety and related, including National Transportation Safety Board (NTSB), New York Times, Washington Post, Bloomberg, USA.gov, and more. These sources provide high-quality positive documents that can be potentially used as positive labels for training the linear SVM classifier. The process of the labels transferring from the news report space to the Twitter space is described as following steps:

**Step 1: News Reports Crawling**: We implemented a web crawler that is able to automatically download transportation related news reports from the above websites that have specific sections or boards related to transportation safety. The crawled web documents are mostly HTML format. We implemented a content extraction component that is able to extract the title and first paragraph for each web document.

**Step 2: Keywords Mapping**: The dictionary of keywords used in news reports is different from the dictionary of keywords used in Twitter. It is necessary to find the mappings between keywords in these two dictionaries. We implemented two strategies. The first strategy first builds a co-occurrence graph of keywords by considering both tweets and news reports as one integrated corpus. Two keywords are connected via an edge in the co-occurrence graph if these two keywords co-occur in a tweet or in a news

report. The weight of the edge is calculated based on the frequency of co-occurrence of these two keywords. After the co-occurrence graph is constructed, it then partitions the graph into disjoint sub-graphs using the maximum cut metric. The keywords in each sub-graph will be treated as "synonyms". The second strategy is based on topic modeling and transfer learning. Similar to the first strategy, the co-occurrence graph is constructed and represented as a similarity matrix. It then jointly learns topic models for both tweets and news reports by constraining that both topic models share "keywords" in a latent space shared by both tweets and news reports, which can be discovered by decomposing the similarity matrix.

**Step 3: Linear SVM Classifier Re-Training**: After the news reports related to transportation are collected and the keywords are mapped to keywords used in Twitter, we can then transform these news reports to positive feature vectors in the feature vector space for training the linear SVM classifier. As indicated in Figure 7, the linear separating hyper plane will be adjusted based on the addition of labels from news reports.

**Figure 7: Adjustment of the Linear SVM Classifier Using Additional News Labels**

## CHAPTER 2: Geocoding of Transportation and Safety Related Tweets

The geographic location information in Twitter is rich but very noisy. First, no more than 3 percent tweets directly provide latitude/longitude coordinates from geo-tagging enabled intelligent phones. These coordinates relate to the locations where users posted the tweets, but are not necessarily related to the tweet content. Second, there is a JSON "Place" object from tweets delivered by the Twitter APIs, which encodes a location associated with the tweet. This object may provide the fields such as city and country, and even finer-grained information such as business names and street addresses. Third, user profile provides location information that was inputted by users. Fourth, there are mentions of locations in tweet texts. Finally, the words in tweet texts that are not explicitly about location names could still provide implicit information that has positive correlations to the related locations. For example, a tweet that mentions "rocket" may

relate to the city of Huston with a high probability [1], which is the headquarter city of NASA and also the home city of the NBA basketball team Rockets. This section discusses five major components that were implemented for the geocoding of transportation and safety related tweets.

### 2.1. Fuzzy Location Extraction from User Profile

Many users publicly provide location information in their profiles. The location information was inputted in free-form form, such as "Pittsburgh, PA", "Pitts, PA", and "Pittsburgh Pennsylvania". Twitter did not do validation, and hence the information may have typos, such as "Pittsburgh, Penmsylvania", and even the location information is not useful at all, such as "I am from outside of the earth". Following the techniques proposed by M. Dredze et al. [2], we first normalize the location string and extract location names using the following steps:

- Removal of stop words, extra spaces, and unrelated punctuation, such as ": ! # ; ? () . - /".
- Extraction of location names using regular expression: ".+,\\s*(\\w+)" and match with U.S. states, cities, and abbreviations for each from a predefined dictionary.
- Edit distance [3] calculation to identify location names that have typos. For example, the edit distance between the strings "Penmsylvania" and "Pennsylvania" is 1. If the edit distance between a word and a location name in the predefined dictionary is smaller than a threshold such as 3, then the word will be mapped to the location name.

Note that there are more complicated patterns that can be used to extract more location information. For example, the location string from one user's profile is "CALi b0Y $TuCC iN V3Ga$". This name relates to a male from California "stuck" in Las Vegas. We did not do the deep text analysis to identify these complicated patterns and will leave this task for future work.

### 2.2. User Location Extraction Using Social Ties

Rich studies have been shown that social networks are strongly correlated to the spatial proximities of users [4]. Friends tend to live to nearby, and people are more likely to be friends if they are living close. Recently work demonstrates that the location of one user

can be estimated based on the geometric median of his/her friends [4]. Following this strategy we implemented a label propagation algorithm to estimate the geo-locations of users based on their social ties:

- **Construction of social network graph**: We extracted mentions of users in tweet texts over 12 months and generated users mention graph as the approximate social network graph. Note that, follower-followee information is not directly available from the crawled raw Twitter data, but studies have been shown that users mention graph is a good approximation to follower-followee network [4].
- **Extraction of seeding users**: In order to predict the locations of tweet users, we collected a subset of seeding users who have high quality locations that were extracted from their geo-tags. If a user has more than 15 geo-tagged tweets and the geo-locations are with a radius of 5 miles, then the user will be considered as a seed user.
- **Geolocation Label propagation:** Given the seed users, we first checked the other user (denoted as set S) whose neighbors have the largest number of seed users. Then for each user in S, we estimated his geo location based on the geometric median of his neighbor users who are also seed users. When all users in S were processed, we iterated the preceding process until convergence. Note that, there are also other estimators instead of geometric media for location prediction, such as Oja's simplex median, triangle heuristic, nearest neighbor, random neighor, and most frequent neighbor. The work [4] empirically demonstrates that geometric median performed the best.
- **Geocoding of latitude/longitude coordinates to streets, cities, and states:** The above label propagation procedure is able to predict an approximate latitude and longitude for each user. This step applies Open StreetMap APIs [5] to do reverse geocoding and estimate the nearest street, city, and state for each user.

We observed that the above geocoding algorithm based on label propagation and social ties still failed to generate high quality predications for many users. However, this algorithm is the only available algorithm that is capable to predict a latitude/longitude coordinate for each user.

## 2.3. Location Extraction from Tweet Text

In addition to the location information from user profiles and geo-tags, there are rich location mentions in tweet texts as well. Some examples are shown in Figure 8. There two technical challenge. First, we need to generate a large dictionary of road names. Second, the locations are expressed in free form and there exist a lot of ambiguity issues. For the first technical challenge, we implemented a complicated crewel that was able to crawl all the major freeways and all the exist names from internet for the three states, including Pennsylvania, Virginia, and New York. For the second technical challenge, we implemented regular expressions to match road names and exits. The location names that were extracted are illustrated in Figure 8. The database stores all the crawled road names and exit numbers.

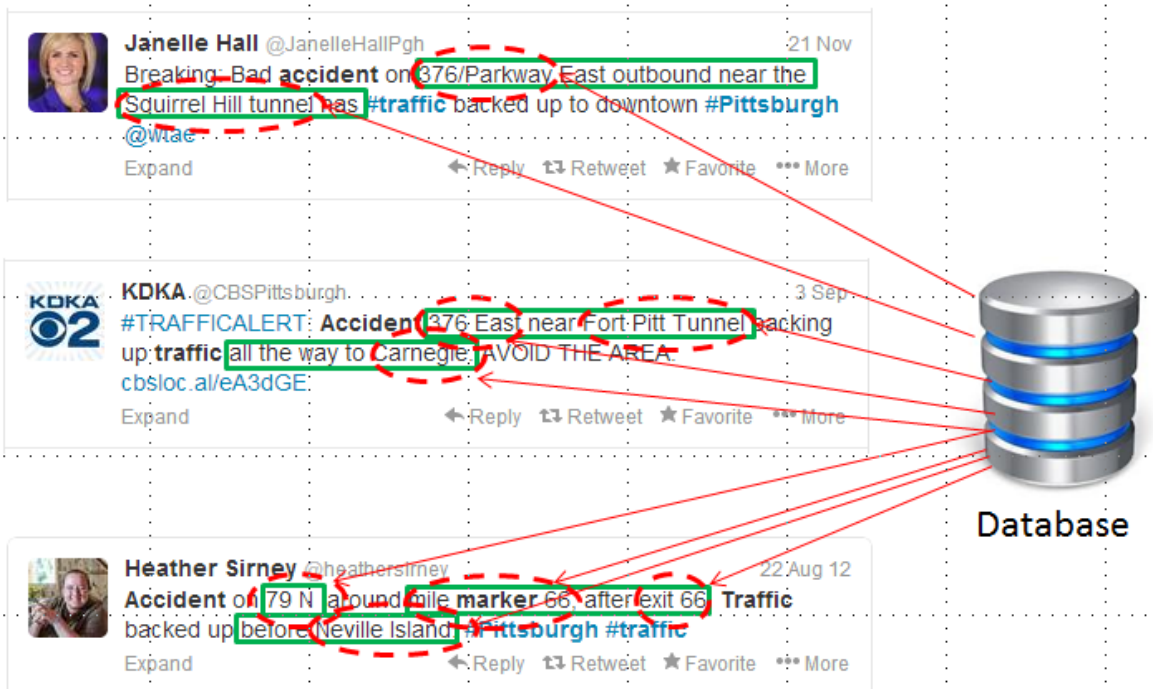



**Figure 8: Example of Location Mentions in Tweets**

Note that there are still a lot of location mentions that could not be extracted using our current algorithm. As shown in Figure 8, some examples include "East outbound near the squirrel hill tunnel", "all the way to Carnegie", and "backed up before Neville Island". These free form patterns are complicated and need to be modeled using deep text analysis. This task is left to our future work and we are planning to apply probabilistic

soft logic as introduced below, which is one of the state of the art entity recognition frameworks for texts.

### 2.4. Mile Marker Extraction from Tweet Text

Among the users who posted tweets related to transportation and safety, we observed that some of the users are influential users from organizations related to transportation, such as the user "511PA Pittsburgh", who provides travel information, traffic alerts and news for the Pittsburgh/Southwest region from 511PA and PennDOT's offices; the user "UEA Incident Alerts" who is a 100% Volunteer Group of Dispatchers from all over the Country who bring users Police/Fire/Rescue/Weather/Breaking News Live as it's Happening; and the user "WPXI Traffic" who provide traffic updates for morning commute from Trisha Pittman.

The transportation tweets posted by influential users are usually better structured then those posted by public users, some example tweets are illustrated as follows:

- "Bad wreck along turnpike at mile marker 61 between Monroeville and Irwin. Avoid turnpike if possible in that area #pittsburgh #traffic"
- "Accident along the PA Turnpike Westbound between Irwin and Pittsburgh at mm 63.7. Car off to the side of the road with injuries reported."
- "Accident on PA Turnpike, both directions mm 56.4. Drivers exiting at Pittsburgh Int will encounter Ax on ramp just prior to toll booth."
- "Accident reported on the PA Turnpike Westbound between Irwin and Pittsburgh at mm 62. http://www.wpxi.com/s/traffic"

We observed that influential users usually provide mile marks when report traffic incidents and congestions. The mile markers provide very valuable information that can be used accurately geocode the tweets to latitude and longitude coordinates. Our algorithm to extract and geocode mile markers is described as following steps:

- Extract mentions of mile markers using regular expression to recognize the forms "Mile Marker", "MM", "M Marker", and "Mile M".
- Extract the road names and directions using regular expression to recognize the forms such as "I-79 North", "I-79 N", "Interstate 79", "Interstate 79 SB", "EB I-279".

- Search the database and find the nearest freeway exits for both directions based on the distance on mile markers.
- Estimate the latitude/longitude of the mile marker based on the linear interpolation of the two nearest exits.

Note that our database only stores the latitude/longitude information for all the exits. In order estimate the latitude/longitude for any given mile marker, we conducted linear interpolation based on the preceding steps.

## 2.5. Ensemble Locations Information Using Probabilistic Soft Logic

The preceding four geocoding components are able to predict locations at either user level or tweet level, and at different resolutions, such as latitude/longitude, street, city, state, and country levels. Some predicted locations may relate to the same user or tweet. The predicted locations may conflict with each other. For example, the predicted location of a user based his/her profile is "Pittsburgh, PA", but the predicted location using social ties is "Falls Church, VA". Some predicted locations may be complimentary to each other. For example, the predicted location of a user based on his/her profile is "Pittsburgh", and the predicted location based on the user's tweets is somewhere near I-279. Then, we have a higher confidence that the user locates currently near I-279 exit 70 B, because only this exit number of I-279 is neighboring to the city of Pittsburgh.

We applied probabilistic soft logic (PSL) [6] to fuse all the location information in to a unique prediction for transportation and safety related tweets. A PSL program is defined as a set of first order logic rules that have conjunctive bodies and single literal heads. Only non-negative weights are allowed. Some example PLS rules defined for this task are shown as follows:

- 0.8: Profile (User A, City C) AND Post(User A, Tweet T) AND Location(Tweet T, City C) → votesFor(User A, City C)
- 0.4: Profile (User A, City C) AND Post(User A, Tweet T) AND Location(Tweet T, City D) → votesFor(User A, City C)
- 0.2: Profile (User A, City C) AND Post(User A, Tweet T) AND Location(Tweet T, City D) → votesFor(User A, City D)

- 0.8: Profile(User A, City C) AND Post(User A, Tweet T) AND Location(Tweet T, City C) AND Neighbor(User A, User B) → votesFor(User B, City C)
- 0.7: Profile (User A, Abbreviation B) AND FullName(Abbreviation B, City C) AND Post(User A, Tweet T) AND Location(Tweet T, City C) AND Neighbor(User A, User Z) à votesFor(User Z, City C)
- 0.6: Profile (User A, City C) AND Post(User A, Tweet T) AND Nearby(Tweet T, City C) → votesFor(User A, City C)

The functions used above are described as follows: 1) Profile(User A, City C) means the predicted location based on the profile of user A is city C; 2) FullName(Abbreviation B, City C) means city C is the full name of the abbreviation B; 3) Post(User A, Tweet T) means tweet T is posted by user A; 4) Location(Tweet T, City C) means the predicted location based on the text of tweet T is city C; and 5) votesFor(User A, City C) means the rule will vote the location of user A for city C. The real number between 0 and 1 ahead of each rule refers to the confidence of the rule.

Given the defined PSL rules, the predicted locations of users and tweets, and raw tweets, the PSL tool will search for the best prediction of locations that can the aggregated confidence by integrating all the rules.

The confidences will be difficult to be decided. The PSL tool provides the function that can automatically adjust the confidences, but only local optimum of the confidence values will be returned, and it is required to provide labeled training data. The quality of the local optimum depends on the training data size. In our current implementation, we manually defined and fixed the confidences, and the results look reasonable. The further optimization of the confidence values will be conducted in the second phase of this project.

# CHAPTER 3: Topic Modeling and Sentiment Analysis

After the transportation and safety related tweets are retrieved and their geo-locations are accurately estimated, the next task is to extract useful patterns from these geocoded transportation tweets. This section presents four major sub-tasks for pattern discovery, including tweets categorization, topic modeling, topic chaining, and sentiment analysis.

## 3.2. Tweets Categorization

The geocoding tweets were classified into different categories, including biking, accident, congestion, road damage, pedestrian, traffic lights, and others. For the current implementation, we did not train a classifier for this task, due to the time constraints. Instead, we predefined a set of keywords for each category, and rules based on keyword matches to do the categorization. For example, for the category “biking”, the predefined keywords include “cycle”, “bike”, “bicycle”, “cyclist”, and “bicyclist”. The keywords are matched after the stemming preprocess. Taking the word “bike” as an example, other forms such as “biking” and “biker” will be matched as well. For the category “accident”, the predefined keywords include “collision”, “kill”, “crash”, “accident”, “fire”, “disabled”, and “blocking”.

## 3.1. Topic Modeling

For each category of tweets, this sub-task is to automatically identify hot topics from the tweets. A topic is defined as a multinomial distribution of words from a predefined dictionary. For example, if a topic is talking about biking safety in the Forbes Ave road, the words such as “biking”, “forbes”, and “ave” will have high probabilities and the unrelated words will have close to zero probabilities.

We applied Latent Dirichlet Allocation, the popular topic modeling approach, to automatically generate topics. As the preprocessing step, we filtered out stop words, and other words that have frequencies lower than the threshold 15. The default number of topics is set to 10. Note that, it is difficult to estimate the optimal number of topics. In the current literature, there are two popular methods to do the estimation. The first method is so-called nonparametric Bayesian modeling based on Chinese restaurant process. The second method is to set a fixed but reasonable large number and later filter out the learned topics that have priors smaller than a threshold, such as 0.01. In our current

implementation, we considered the second method. Figure 9 is an example of word cloud visualization of topics discovered. The tweets that are marked with read rectangles were formed one topic, and the tweets that are not marked were formed to another topic. The left topic indicates that people were complaining traffic congestions. The second topic indicates that people were complaining traffic accidents and safety issues.


**Figure 9: Word Cloud Visualization of Automatically Discovered Topics**

### 3.2. Topic Chaining

After the topics are discovered for each spatial region (e.g., road, city, state) and time interval (e.g., hour, day), this sub-task is to identify connections between the discovered topics across the time, which is called topic chaining. The connections will be useful to do casualty analysis and to study how event evolves. The chaining algorithm is described as following steps. For a given topic (T),

- Consider the initial set of connected topics $S = \{T\}$.
- Topic similarity: The similarity between two topics is defined as the Jaccard index [7] of the related sets of top 100 keywords.
- We first identify the topics that were discovered one time step ahead. The similarities between the topics in $S$ and the identified topics (named step-1

forward topics) and the step-1 forward topics whose similarities are greater than the threshold 0.5 will be reserved. Denote the set of topics as $S^{-1}$.

- For the given topic, we identify the topics that were discovered one time step afterward. The similarities were calculated similarly as above and the step-1 backward topics whose similarities are greater than the threshold 0.5 will be reserved. Denote the set of topics as $S^{+1}$.
- The reserved topics in previous two steps are combined with the original given topic, and are formed as the updated candidate set of topics: $S = S \cup S^{+1} \cup^{-1}$.
- Repeat the previous steps, until the entire forward and backward topics are processed.

Note that the threshold 0.5 is empirically tuned, and only the topics that are adjacent in time will be calculated, which ensures that the topics that are not adjacent in time will be not connected.

### 3.3. Sentiment Analysis

This sub-task aims to estimate the sentiment for a set of tweets that may relate to a specific topic, a spatial region (e.g., street, city, state), or a time stamp (e.g., hour, day). We applied a sentiment analysis package to calculate the sentiment score for a given tweet. The sentiment components of a twee text include two types: facts and opinions. Opinions refer to people's sentiments and feelings toward the related event. Two corresponding scores were calculated, including polarity and subjectivity. The polarity score is valued between -1.0 and 1.0, and the subjectivity score is valued between 0.0 and 1.0. If the polarity is below 0.0, then it indicates that the tweet is talking about some sad topic; otherwise, it is talking about happy topic. If the subjectivity score equals 0.0, that means the words in the text are objective; otherwise, the words are subjective, and the value between 0 and 1 is the balance between objectivity and subjectivity.

Figure 10 illustrates the sentiment analysis function in combination with word cloud visualizations. The time series chart refers to the day by day sentiment polarity scores of a specific region (e.g., Pittsburgh). There is a bottom point during the date 2013-02-02 and a peak point during the date 2013-02-15. The bottom point refers to some negative

sentiment topic. From the related word cloud, we observe that people were complaining the current traffic congestion. The peak point refers to very positive sentiment topic. From the related word cloud, we observe that people are talking about the smooth traffic during morning and evening hours.

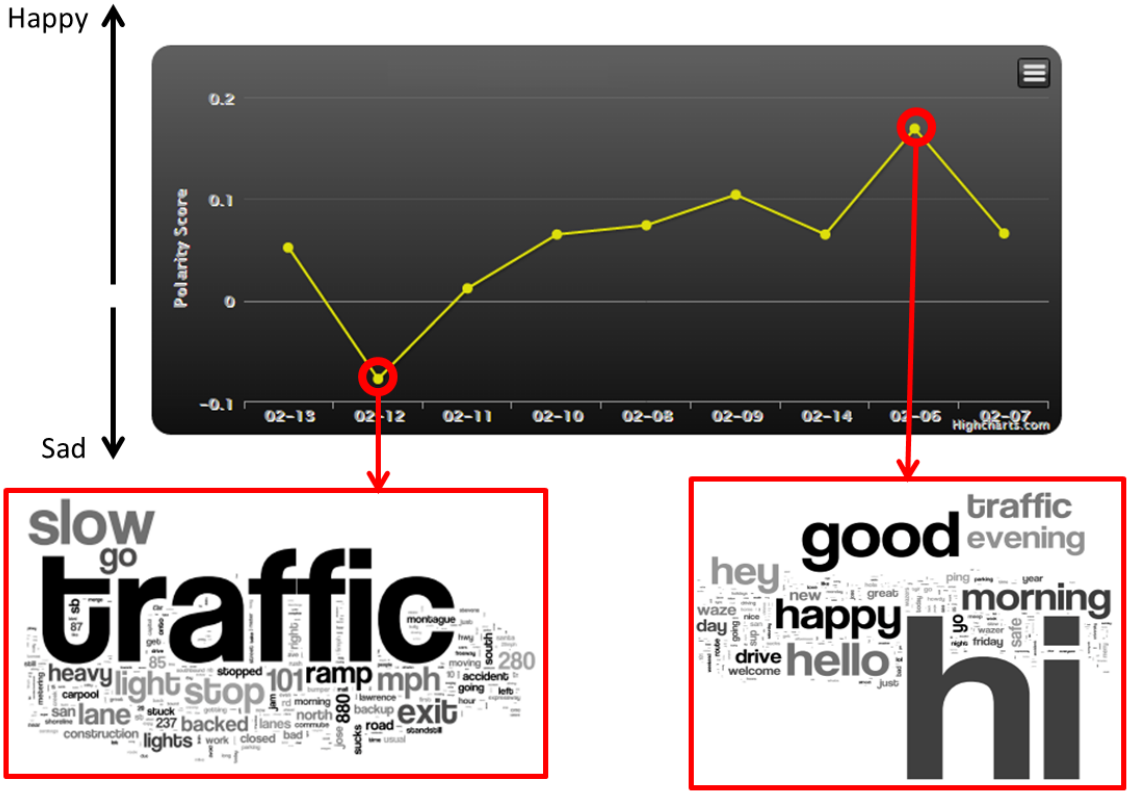



**Figure 10: Example of Sentiment Analysis**

# CHAPTER 4: Implementation of an Interactive Web Interface

This task presents the implementation of a web based interactive web interface that allows users to study different summary statistics of transportation and safety related tweets, and to identify potential safety bottlenecks by using the advanced pattern learning functions, such as topic discovery, topic chaining, and sentiment analysis.

## 4.1. Flow of Data Processes

The flow of data processes is show in Figure 11. First, the raw tweets were retrieved by using the Twitter Search API, Twitter Streaming API, and Gardenhose API, and then merged to one data set. Second, the automatic content filter (the linear SVM classifier) was called to filter out all noise tweets, and then separated to daily and hourly data files.

The geocoding module implemented in Task 2 was called to estimate the most possible geolocation (at latitude/longitude, street, city, and state levels) for each tweet in the data files. Given the estimated geolocation information, the data files were future separated into the regions of cities and states. After that, for each combination of geographic region (city, state) and temporal window (hour and hour), the topic modeling and sentiment analysis module was called to the corresponding data file to generate 10 default topics and calculate the overall sentiment score for the data file and the sentiment score for each topic. All the results were pre-computed and maintained by a file management system.

## 4.2. Design of Main Modules

As shown in Figure 12, the web based prototype system was implemented based on the functions implemented by the preceding three tasks. The execution sequence of the modules is: 1) data crawling; 2) automatic content filtering; 3) topic modeling; 4) topic chaining; 5) sentiment analysis; and finally web interface visualization. The organization of the web pages is shown in Figure 13.

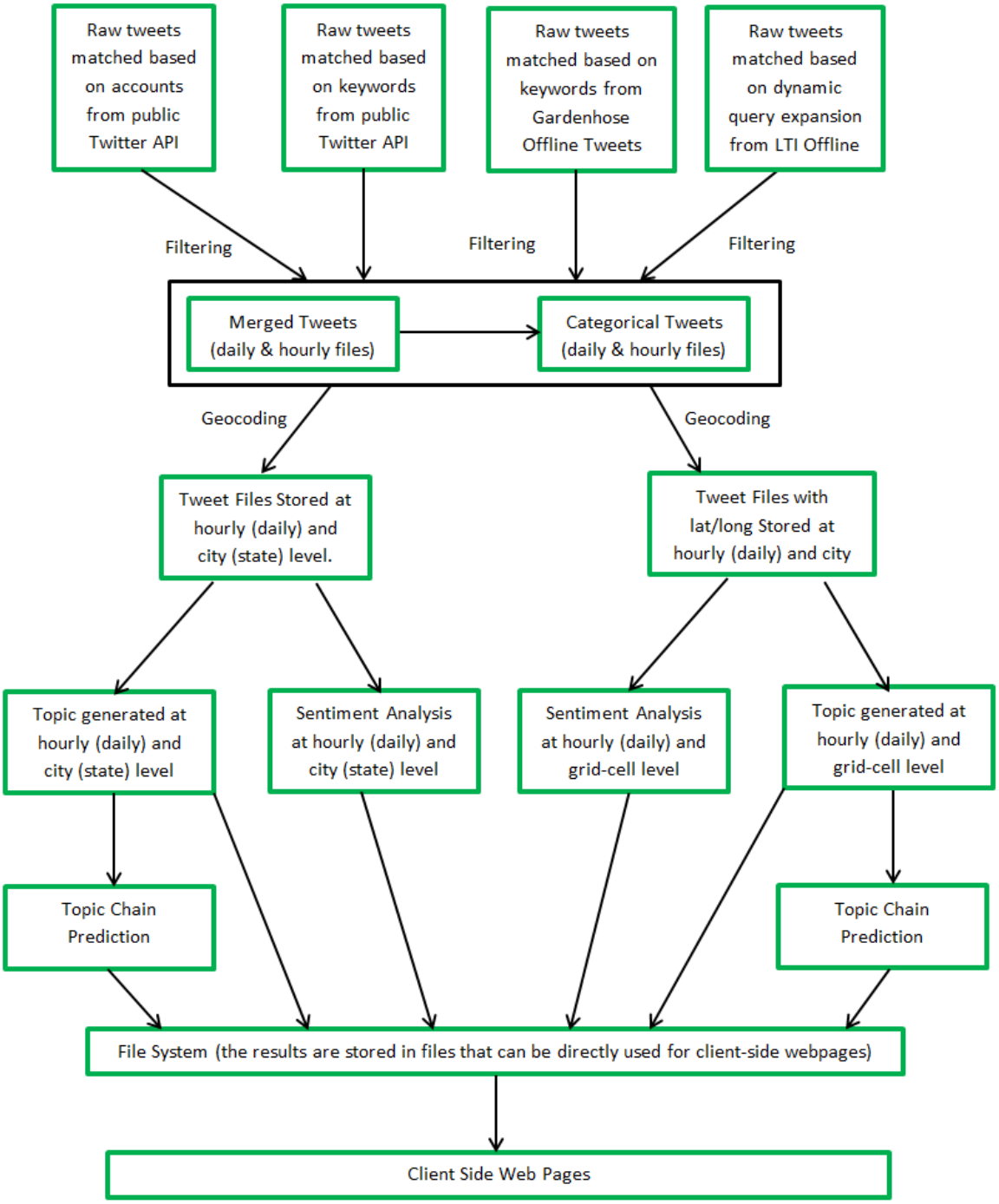



**Figure 11: Flow of Data Processes**

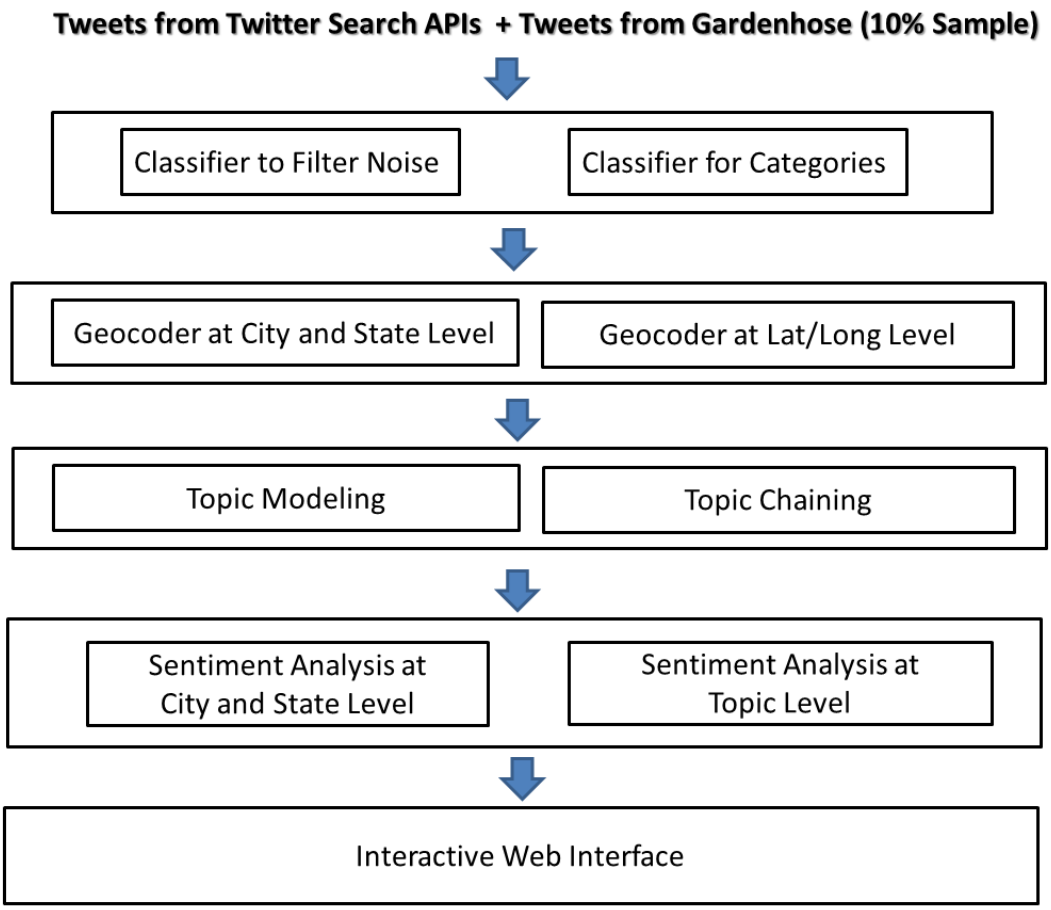



**Figure 12: Main Modules of the Web Based Prototype System**

### 4.3. Optimizations on the Server and Client Sides

Due to the large volume of Twitter data to be processed, it is technically challenging to process and analyze the inherent patterns. Several optimizations were conducted in order to ensure real-time performance. First, we applied a linear support vector machine (SVM) classifier, instead of nonlinear models. The linear classifier guarantees linear time cost to filter out the large amount of noise tweets. We observed that the number of transportation and safety related tweets is less than 0.1 percent. Second, all the geocoding, categorization, topic modeling, and sentiment analysis components were preprocessed for each combination of geographic location and time window. Therefore, the web interface is able to load the results directly using the popular Ajax techniques. Third, the generation of word cloud was implemented using the strategy of lazy update. We did not pre-compute all the potential word clouds, because the number of word clouds to be generated is more than ten thousands, and the required storage space is huge. In order to

save the storage space, the lazy update strategy will first check if the required word cloud files have been generated. If yes, then the web interface will directly load the word cloud files; otherwise, the word cloud files will be generated and achieved in the server. Using this strategy, although the initial users will observe some downgraded system performance, but after an enough number of users have used, all the popularly requested word clouds have already been generated and archived, and the system will perform very efficiently. Finally, the keyword based search function in the web interface uses advanced caching strategy. All the related raw tweets were uploaded to the web client and stored in memory. Each time a user submits a new query, the query process will be conducted directly in the memory of the client computer, instead of going through the network and sever.

### 4.3. Design of Web Pages on the Client Slides

The web interface is composed of seven main pages, including the dashboard page, the emergence page, the top-topic analytic page, the top-user analytic page, the top-city analytic page, and the traffic map page. The organization of the main pages is shown in Figure 13.

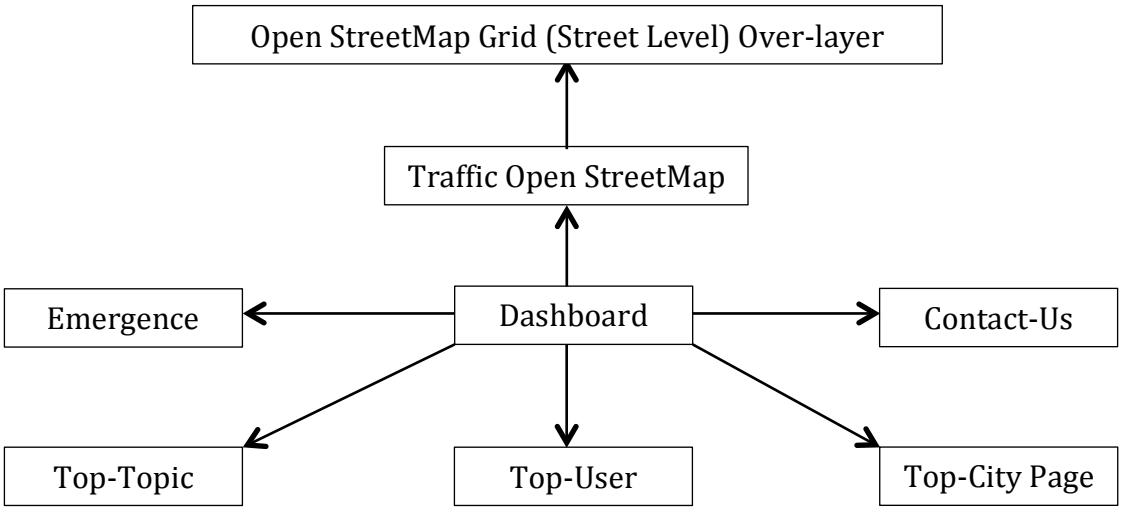



**Figure 13: ER Relationships between Web Pages**

The interfaces of main webpages are presented as follows:

## 1. The Dashboard Page

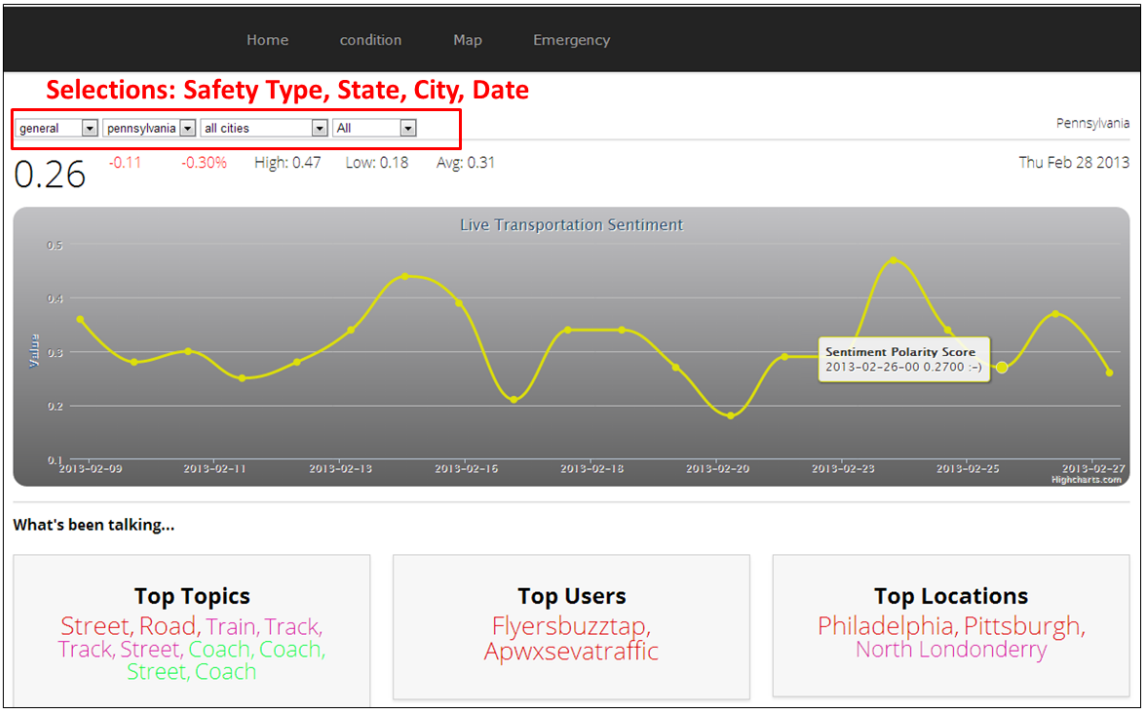



**Figure 14: The Dashboard Page**

The dashboard page shows the hour-by-hour and day-by-day sentiment curves for the selection of a specific city (or state) and a specific category (e.g., biking, accident, congestion, road damage, pedestrian). The left top panel shows summary statistics of the sentiment scores (negative score means feeling bad, and positive score means feeling good). As shown in this example, the average sentiment score is 0.31, with the highest value 0.47 and the lowest value 0.18. The sentiment score of the current day is 0.26, which decreased 0.30 percent from the score of the previous day. The numbers indicate that during the time period from 2013-02-09 to 2013-02-27, the public was happy in overall about the transportation of the state Pennsylvania. The middle-layer time series curve shows the day by day happiness degrees of the public, which provides very useful information for transportation planners and managers. The bottom panel shows the top ranked topics, users, and geographical locations. The ranking of topics is decided based on the volume of related tweets for each topic from large to small. The rankings of users and locations are decided similarly. The label for each topic was decided by the word that

has the highest probability over all the words. The list boxes on the top allow users to select a specific combination of city (or state), date (or hour), and category.

## 2. City and State Analytics Page

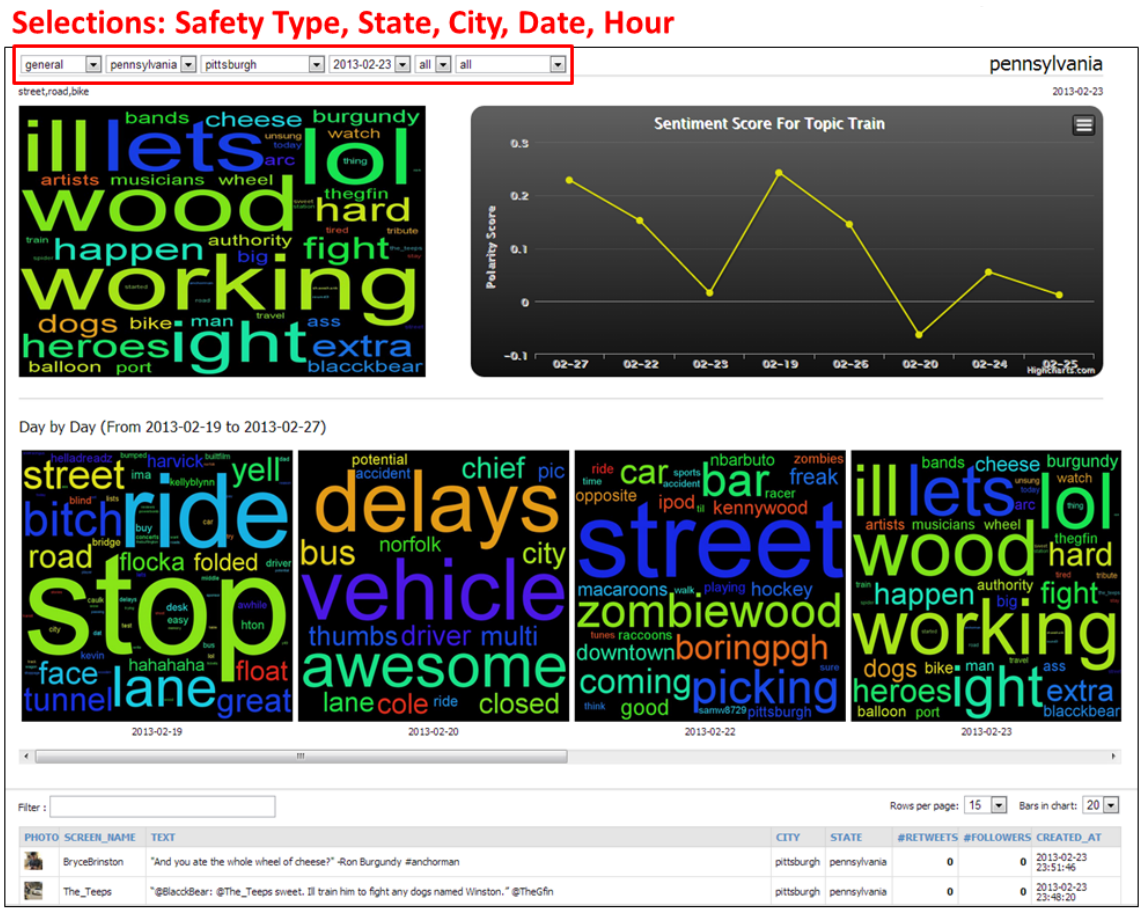
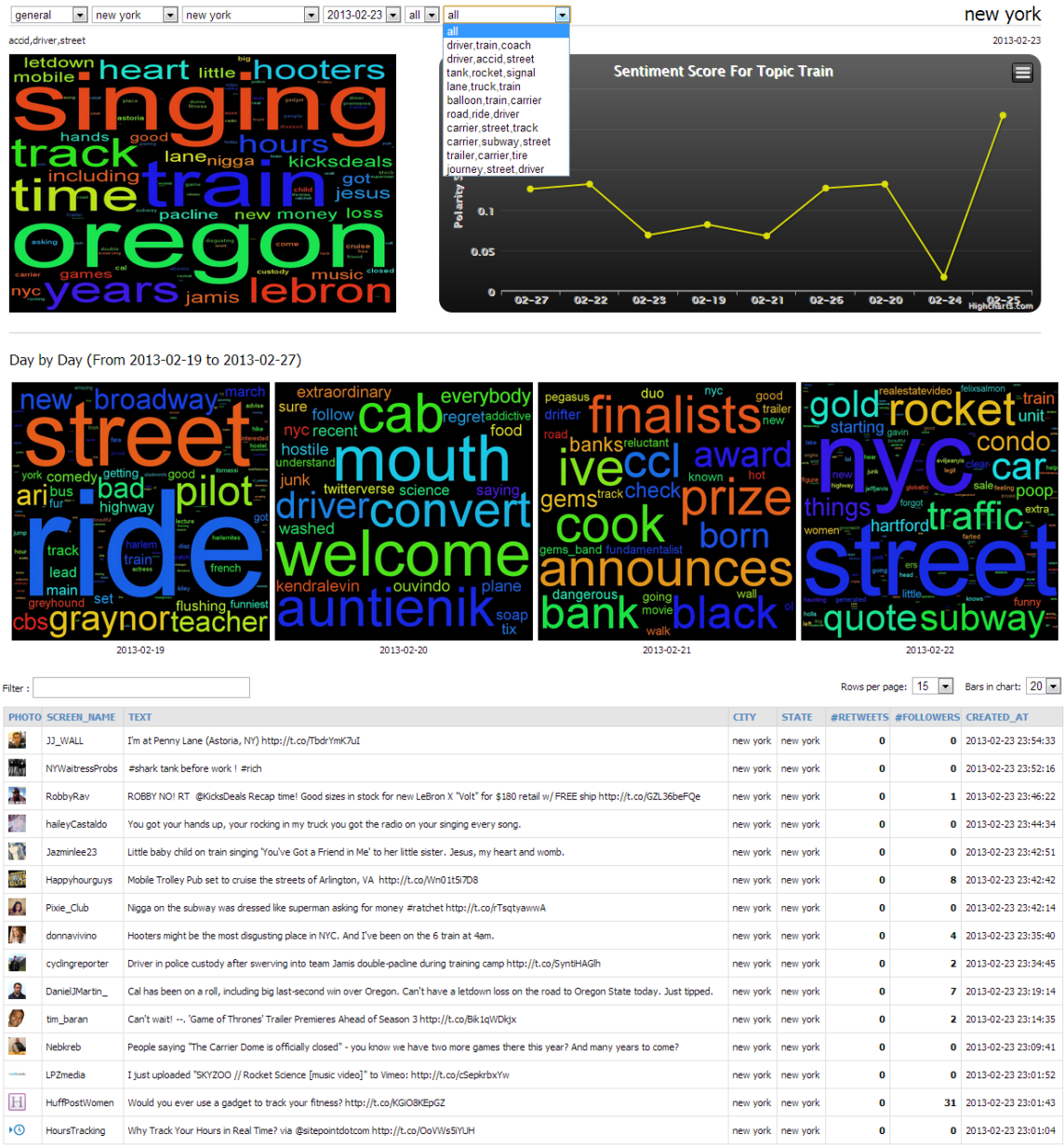



**Figure 15: The City and State Analytics Page - 1**

The city and state analytics page allows users to evaluate the traffic patterns at city and state levels. The list boxes on the top allow users to specify a specific category, state, city, date, hour, and topic. As shown in Figure 15, ten topics are shown in the list box. Each topic is represented using the three top ranked terms in the set of related tweets. The top left word cloud shows the distribution of words for the current configuration of parameters. If the user does not select a specific topic, then the word cloud will be generated based on the raw tweets related to the selection of category, location, and time. The middle layer shows day by day word clouds that help users visualize and evaluate the evolution of tweet content. As shown in this example, in the date 2013-02-19, people

talked about issues related to biking; and in the date 2013-02-20, people talked about lane closes, congestions, and delays. The bottom layer shows the list of related raw tweets to help users to do evaluation based on the direct understanding of raw tweet texts. Note that, there is also a text box labeled as "Filter" that allows users to filter tweets based on keywords. For example, if the user inputs "traffic", then only tweets containing this specific keyword will be shown.




| PHOTO | SCREEN_NAME | TEXT | CITY | STATE | #RETWEETS | #FOLLOWERS | CREATED_AT |
|---|---|---|---|---|---|---|---|
| | JJ_WALL | I'm at Penny Lane (Astoria, NY) http://t.co/TbdrYmK7uI | new york | new york | 0 | 0 | 2013-02-23 23:54:33 |
| | NYWaitressProbs | #shark tank before work ! #rich | new york | new york | 0 | 0 | 2013-02-23 23:52:16 |
| | RobbyRav | ROBBY NO! RT @KicksDeals Recap time! Good sizes in stock for new LeBron X "Volt" for $180 retail w/ FREE ship http://t.co/GZL36beFQe | new york | new york | 0 | 1 | 2013-02-23 23:46:22 |
| | haileyCastaldo | You got your hands up, your rocking in my truck you got the radio on your singing every song. | new york | new york | 0 | 0 | 2013-02-23 23:44:34 |
| | Jazminlee23 | Little baby child on train singing 'You've Got a Friend in Me' to her little sister. Jesus, my heart and womb. | new york | new york | 0 | 0 | 2013-02-23 23:42:51 |
| | Happyhourguys | Mobile Trolley Pub set to cruise the streets of Arlington, VA http://t.co/Wn01t5i7D8 | new york | new york | 0 | 8 | 2013-02-23 23:42:42 |
| | Pixie_Club | Nigga on the subway was dressed like superman asking for money #ratchet http://t.co/rTsqtyawwA | new york | new york | 0 | 0 | 2013-02-23 23:42:14 |
| | donnavivino | Hooters might be the most disgusting place in NYC. And I've been on the 6 train at 4am. | new york | new york | 0 | 4 | 2013-02-23 23:35:40 |
| | cyclingreporter | Driver in police custody after swerving into team Jamis double-pacline during training camp http://t.co/SyntiHAGlh | new york | new york | 0 | 2 | 2013-02-23 23:34:45 |
| | DanielJMartin_ | Cal has been on a roll, including big last-second win over Oregon. Can't have a letdown loss on the road to Oregon State today. Just tipped. | new york | new york | 0 | 7 | 2013-02-23 23:19:14 |
| | tim_baran | Can't wait! --. 'Game of Thrones' Trailer Premieres Ahead of Season 3 http://t.co/Bik1qWDkjx | new york | new york | 0 | 2 | 2013-02-23 23:14:35 |
| | Nebkreb | People saying "The Carrier Dome is officially closed" - you know we have two more games there this year? And many years to come? | new york | new york | 0 | 0 | 2013-02-23 23:09:41 |
| | LPZmedia | I just uploaded "SKYZOO // Rocket Science [music video]" to Vimeo: http://t.co/cSepkrbxYw | new york | new york | 0 | 0 | 2013-02-23 23:01:52 |
| | HuffPostWomen | Would you ever use a gadget to track your fitness? http://t.co/KGiO8KEpGZ | new york | new york | 0 | 31 | 2013-02-23 23:01:43 |
| | HoursTracking | Why Track Your Hours in Real Time? via @sitepointdotcom http://t.co/OoVWs5iYUH | new york | new york | 0 | 0 | 2013-02-23 23:01:04 |

**Figure 16: The City and State Analytics Page - 2**

## 3. Topics Analytics Page

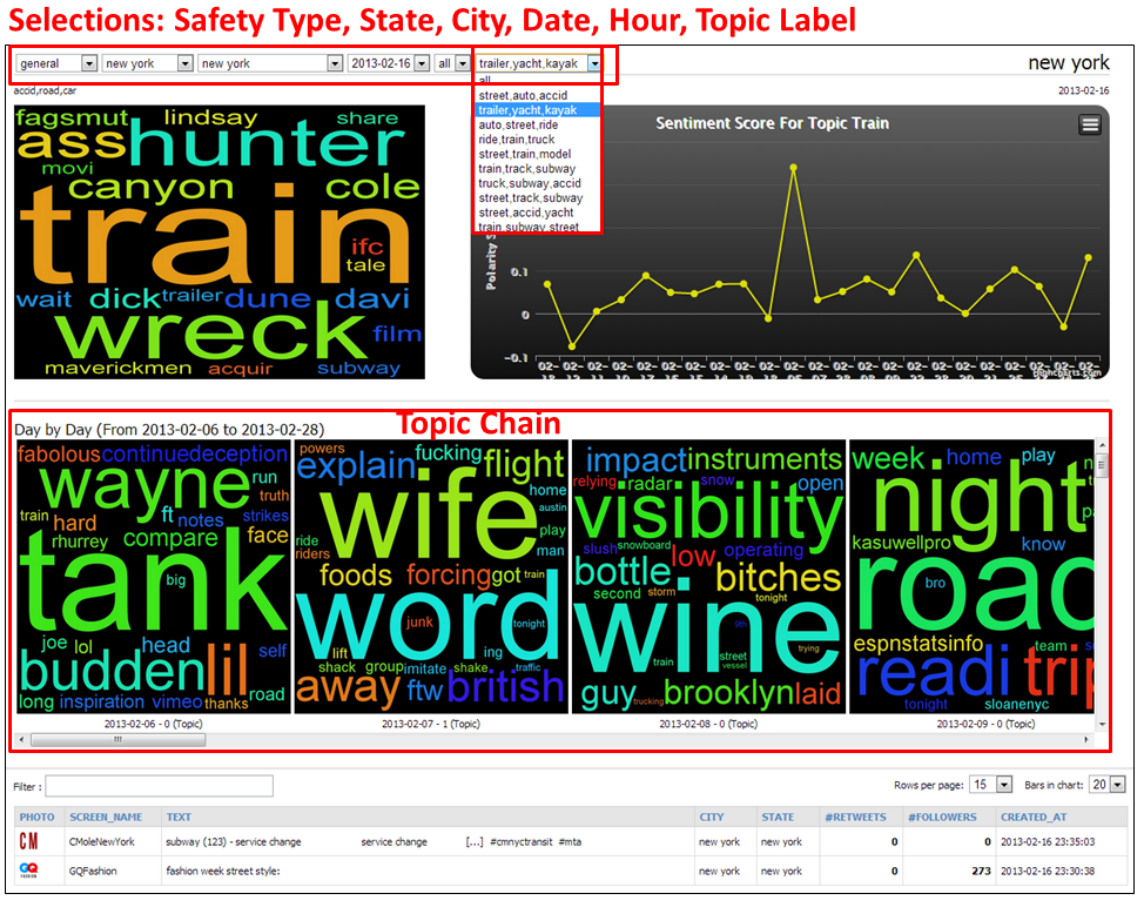
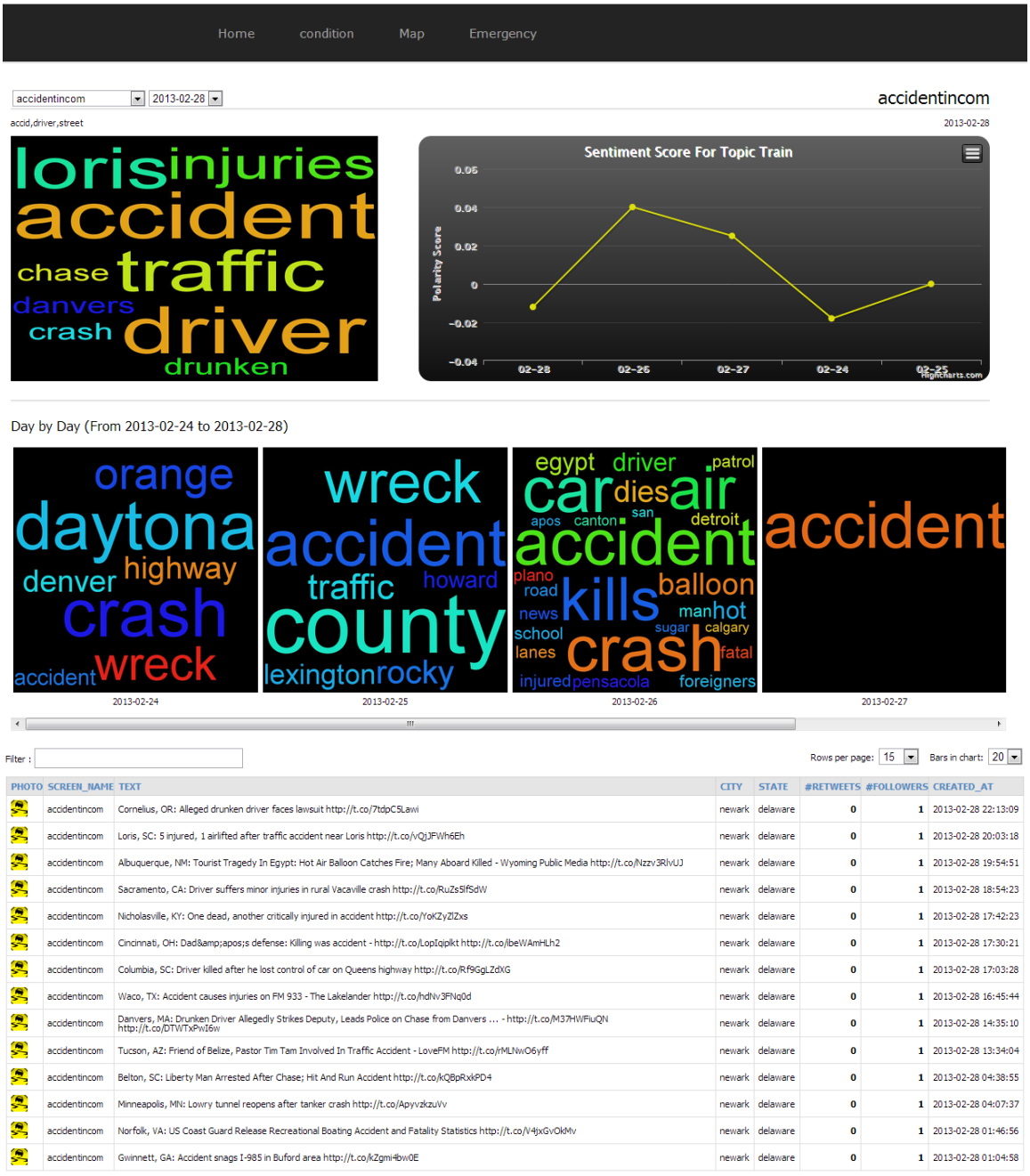



**Figure 17: The Topic Analytics Page**

The topics analytics page allows users to evaluate transportation and safety related topics that were automatically generated using topic modeling. The user needs to select a specific topic from the list box as marked using red rectangle in Figure 16. Then, the top left word cloud will show the word cloud of the tweets related to the selected topic. The right top plot provides sentiment scores related tweets in adjacent hours or days. As shown in this Figure, there is a clear peak point that indicates that people are excited about something related to the select topic. The middle layer shows word clouds of related topics in adjacent hours or days. These topics were discovered using topic chaining as discussed in Section 3.2. This function helps users evaluate the evolution of the selected topic including the changing of text content. The bottom layer provides the related raw tweets that help the user better understand the selected topic and quickly identify some potential traffic patterns.

## 4. Twitter User Analytics Page




| PHOTO | SCREEN_NAME | TEXT | CITY | STATE | #RETWEETS | #FOLLOWERS | CREATED_AT |
|---|---|---|---|---|---|---|---|
| | accidentincom | Cornelius, OR: Alleged drunken driver faces lawsuit http://t.co/7tdpC5Lawi | newark | delaware | 0 | 1 | 2013-02-28 22:13:09 |
| | accidentincom | Loris, SC: 5 injured, 1 airlifted after traffic accident near Loris http://t.co/vQjJFWh6Eh | newark | delaware | 0 | 1 | 2013-02-28 20:03:18 |
| | accidentincom | Albuquerque, NM: Tourist Tragedy In Egypt: Hot Air Balloon Catches Fire; Many Aboard Killed - Wyoming Public Media http://t.co/Nzzv3RlvUJ | newark | delaware | 0 | 1 | 2013-02-28 19:54:51 |
| | accidentincom | Sacramento, CA: Driver suffers minor injuries in rural Vacaville crash http://t.co/RuZs5fSdW | newark | delaware | 0 | 1 | 2013-02-28 18:54:23 |
| | accidentincom | Nicholasville, KY: One dead, another critically injured in accident http://t.co/YoKZyZlZxs | newark | delaware | 0 | 1 | 2013-02-28 17:42:23 |
| | accidentincom | Cincinnati, OH: Dad's defense: Killing was accident - http://t.co/LopIqiplkt http://t.co/lbeWAmHLh2 | newark | delaware | 0 | 1 | 2013-02-28 17:30:21 |
| | accidentincom | Columbia, SC: Driver killed after he lost control of car on Queens highway http://t.co/Rf9GgLZdXG | newark | delaware | 0 | 1 | 2013-02-28 17:03:28 |
| | accidentincom | Waco, TX: Accident causes injuries on FM 933 - The Lakelander http://t.co/hdNv3FNq0d | newark | delaware | 0 | 1 | 2013-02-28 16:45:44 |
| | accidentincom | Danvers, MA: Drunken Driver Allegedly Strikes Deputy, Leads Police on Chase from Danvers ... - http://t.co/M37HWFiuQN http://t.co/DTWTxPwI6w | newark | delaware | 0 | 1 | 2013-02-28 14:35:10 |
| | accidentincom | Tucson, AZ: Friend of Belize, Pastor Tim Tam Involved In Traffic Accident - LoveFM http://t.co/rMLNwO6yff | newark | delaware | 0 | 1 | 2013-02-28 13:34:04 |
| | accidentincom | Belton, SC: Liberty Man Arrested After Chase; Hit And Run Accident http://t.co/kQBpRxkPD4 | newark | delaware | 0 | 1 | 2013-02-28 04:38:55 |
| | accidentincom | Minneapolis, MN: Lowry tunnel reopens after tanker crash http://t.co/ApyvzkzuVv | newark | delaware | 0 | 1 | 2013-02-28 04:07:37 |
| | accidentincom | Norfolk, VA: US Coast Guard Release Recreational Boating Accident and Fatality Statistics http://t.co/V4jxGvOkMv | newark | delaware | 0 | 1 | 2013-02-28 01:46:56 |
| | accidentincom | Gwinnett, GA: Accident snags I-985 in Buford area http://t.co/kZgmi4bw0E | newark | delaware | 0 | 1 | 2013-02-28 01:04:58 |

**Figure 18: The Twitter User Analytics Page**

This Twitter user analytics page allows users to evaluate the behavior patterns of a specific Twitter user. The top list boxes allow users to select a specific user by his screen name and a specific date. For example, the user may be interested to check up to date traffic tweets of influential users. Suppose we know that a user posted important

information about an ongoing traffic accident, then we may be interested to check his follow-up tweets in order to know more about the progress of the traffic accident by considering the user as a "sensor" of the accident. As shown in Figure 17, the top left word cloud shows the distribution of words in the raw tweets posted by the user during the date 2013-02-28. The word cloud indicates that the user was talking about accidents that may involve crash and potential injuries of the drivers, and the impacts to the current traffic. The top right plot shows the sentiment cores of the user in adjacent days, which is helpful to infer the behavior and emotions of this Twitter user. The third layer shows day by day word clouds that help users to check the evolution of tweeting content by this Twitter user. The bottom panel shows the raw tweets for deep analysis.

## 5. Traffic Map Page

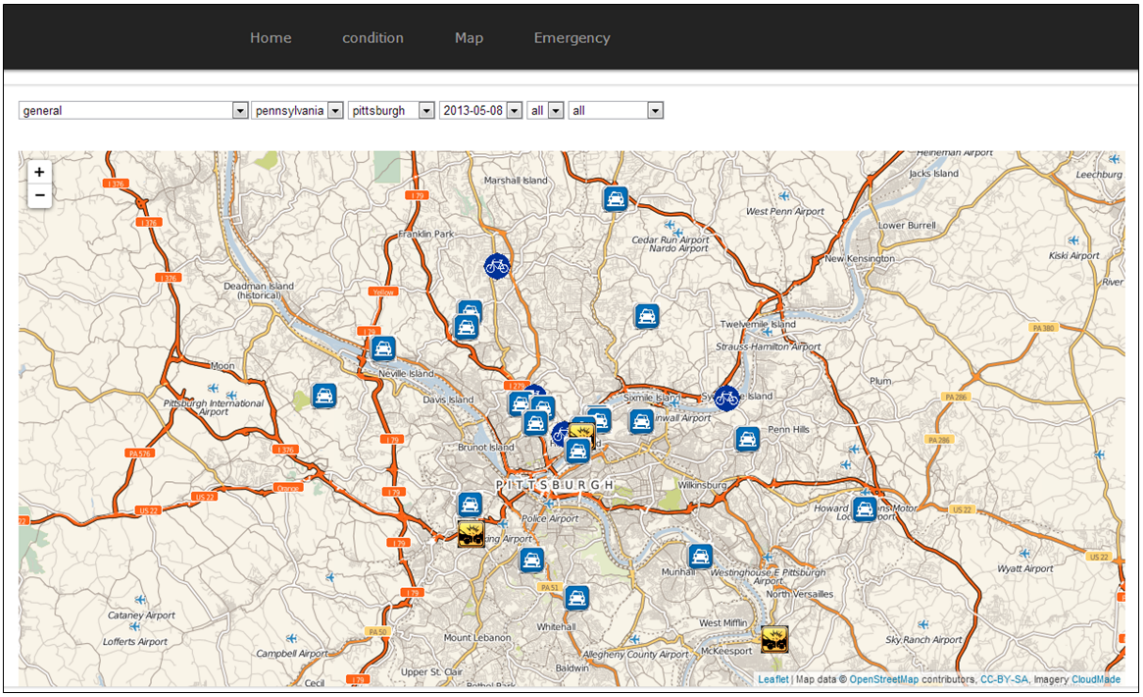



**Figure 19: The Traffic Map Page - 1**

The traffic map allows users to check the geographic regions of transportation and safety related tweets. The top list boxes allow users to specify a specific category (or "general" to include all categories), state, city, date, hour, and topic. Each category is displayed using a different icon. As shown in Figure 18, there are three categories of transportation tweets, including accident, biking, and others. The geolocation of each tweet was

predicted based on the geocoding techniques discussed in Task 2. Figure 19 shows the high resolution location of an example tweet. The tweet text is “Disabled vehicle: NB on I-95 at MM 159 in Prince William Co. 1 lane closed. Potential Delays 8:16AM”. Based on the map visualization, the tweet is accurately geo-located near I-95 MM 159.

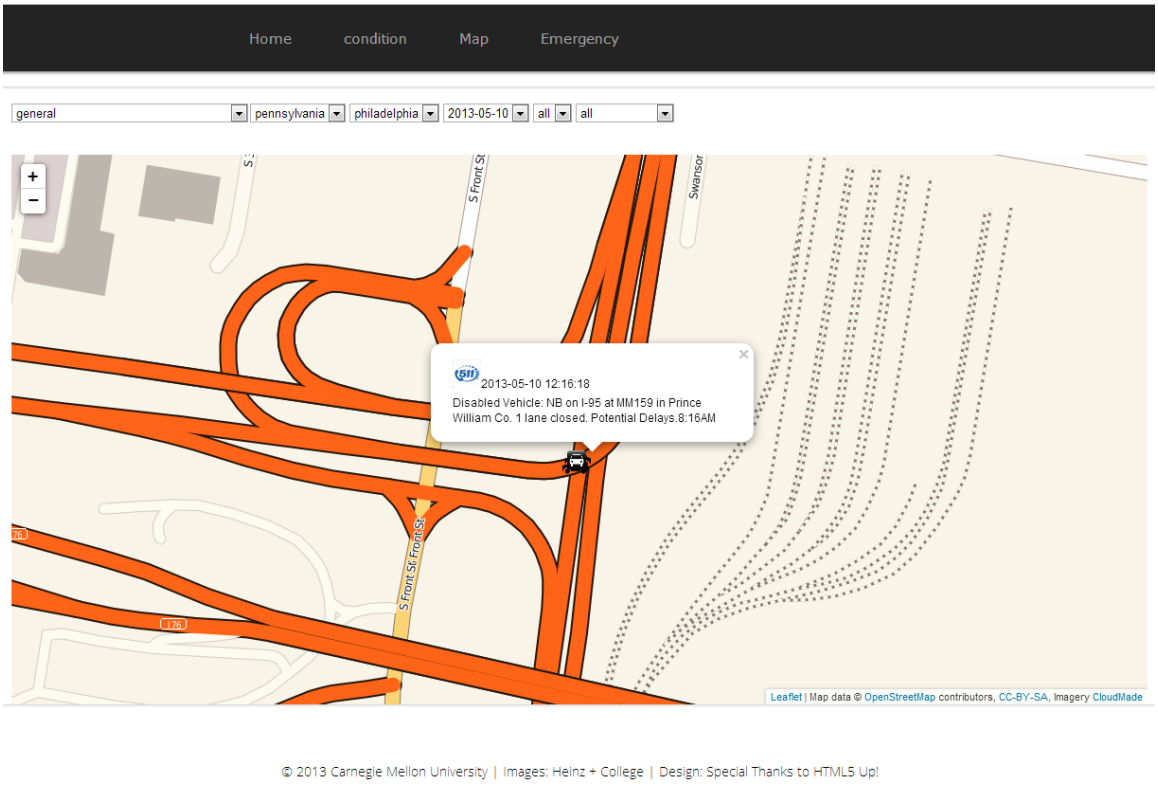



**Figure 20: The Traffic Map Page - 2**

## 6. Emergence Page

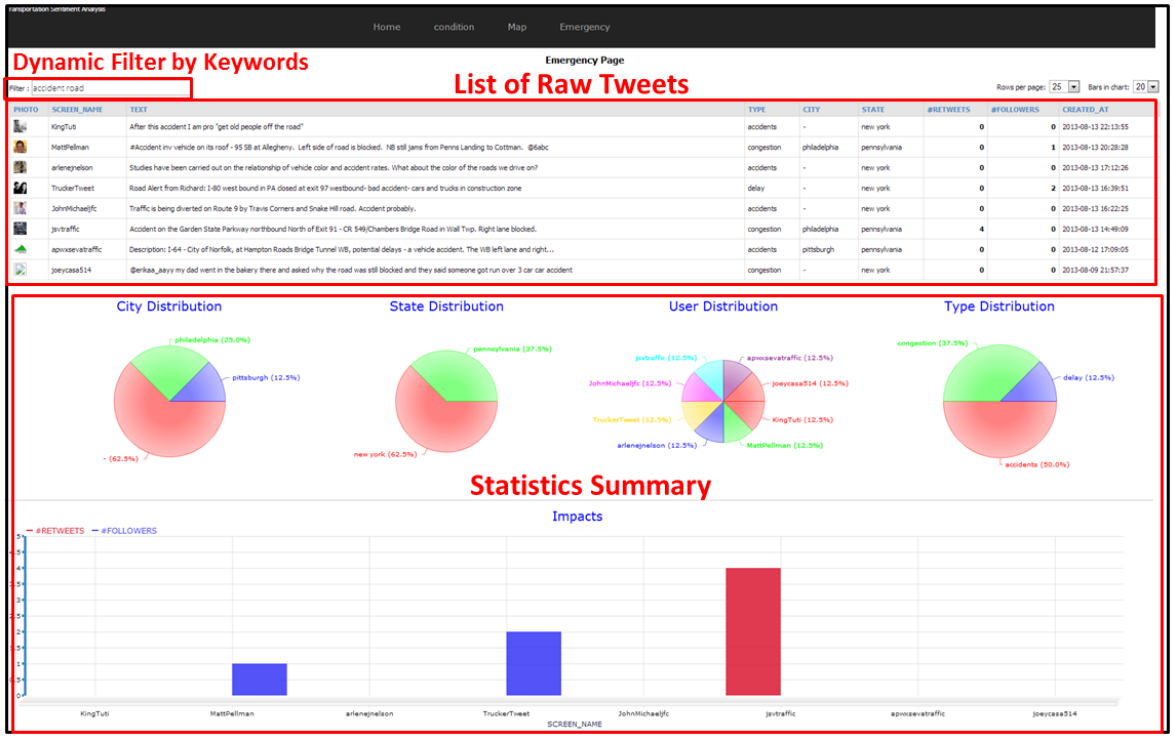



**Figure 21: The Emergency Page**

The emergence page was designed to display only emergent tweets that may require traffic engineers to take emergent actions. The emergent tweets were identified based on a set of predefined rules. For example, one rule is to identify tweets reporting accidents and injuries. Another rule is to identify tweets reporting road damages or flooding. This page provides the list of priority tweets for users to take emergent care. The left top box is a query search box that allows users to filter based on keywords, such as accidents, collisions, and traffic lights. The second layer shows a list of raw tweets. Several key features of each tweet are displayed, including the photo of the Twitter user, screen name, raw tweet text, category, city, state, number of retweets, number of followers, and post time. Several dynamic charting functions were provided, including the pie chart of city distribution of the emergent tweets, the pie chat of state distribution, the pie chart of user distribution, and the pie chart of categories. As illustrated in Figure 20, the user filtered tweets based on keywords “accident” and “road”. The charts show that around 40 percent of tweets are from the cities Philadelphia and Pittsburgh, and more than 50 percent tweets are from the state New York. The distribution users based on their volumes of tweets

seems close to uniform, which potentially indicate that most of tweets are from public users, instead of influential users. The pie chart of category distribution indicates that 50 percent of tweets talked about accident, around 37.5 percent tweets talked about congestion, and around 12.5 percent tweets talked about traffic delay. The bottom plot shows the number of retweets and the number of followers for each related Twitter user. The results indicate that the user "jsvtraffic" has a large number of retweets, but not many followers, and the "truckerTweet" has a large number of followers but no influential retweets. These two patterns are both interesting and may provide some values about the significance of the related users' tweets.

# CHAPTER 5: Conclusion and Future Work

This project presents the design and implementation of a real-time Twitter monitoring system for the detection and visualization of safety related patterns from Twitter data. All the major components have been discussed extensively, including the crawling of tweets that are related to transportation and safety; the automatic filtering of noise tweets; the geocoding of tweets at latitude/longitude, street, city, and state levels; the automatic topic discovery, topic chaining, and sentiment analysis; and the implementation of a web based prototype system.

For future work, we will focus on the following tasks: 1) deep text analysis to improve the geocoding quality. Currently, we can only accurately estimate the latitude/longitude coordinates for those tweets that have either geo-tags or mile marker information. For other tweets, we achieved good geocoding at city and state levels, but not on the level of latitude/longitude; 2) Estimation of real-time traffic flows by fusing Twitter, Foursquare, and traditional traffic sensors data, such as GPS, loop detector, and camera data; 3) Detection of traffic accidents and traffic congestions by fusing the preceding heterogeneous data sources; and 4) prediction of travel times or origination-destination times.

## Acknowledgments

Brendan O'Connor helped collect the 10 percent of public tweets from Gardenhose/Decahose stream.

## References

[1] Zhiyuan Cheng, James Caverlee, and Kyumin Lee, "A Content-Driven Framework for Geo-locating Microblog Users," Transactions on Intelligent Systems and Technology (ACM TIST), vol 4, issue 1, No 2, pages 1 to 27, 2013

[2] Mark Dredze, Michael J. Paul, Shane Bergsma, Hieu Tran, "Carmen: A Twitter Geolocation System with Applications to Public Health," AAAI Workshop on Expanding the Boundaries of Health Informatics Using AI (HIAI), Bellevue, WA, 2013.

[3] Edit Distance, http://en.wikipedia.org/wiki/Edit_distance

[4] David Jurgens, "That's what friends are for: Inferring location in online communities based on social relationships," Proceedings of the 7th International AAAI Conference on Weblogs and Social Media (ICWSM), 2013.

[5] Open StreepMap APIs, http://wiki.openstreetmap.org/wiki/API

[6] Angelika Kimmig, Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor, "A Short Introduction to Probabilistic Soft Logic," NIPS Workshop on Probabilistic Programming: Foundations and Applications – 2012.

[7] Jaccard index, http://en.wikipedia.org/wiki/Jaccard_index

[8] Twitter Search APIs, https://dev.twitter.com/docs/using-search

[9] Bigram, http://en.wikipedia.org/wiki/Bigram