AquaScale: Social Media Based Data Collection and Water Pipe Leakage Reporting & Querying System

Huan Chen, Wei Wang, Zhichun Ning (Group 3)
Donald Bren School of Information and Computer Science
University of California, Irvine
Email: huanc3@uci.edu, wwang16@uci.edu, zhichunn@uci.edu

I. Assignment Description

A. Demo Video

Demo video: https://youtu.be/DmhD0gRGB8c

B. Role Description

- Huan Chen: built the tweets analyzer, classifier and collector, developed the web query interface to search for historical and real-time tweets at given time and location, classify real-time user reports, vote for tweets and messages)
  Code: https://github.com/bjtuhfz/LeakageQuerySystem
- Wei Wang: developed the web app (user register, sign in, handle user report submitted via web & iOS, display historical messages, vote for messages, send notifications to user)
  Code: https://github.com/bjtuhfz/LeakageReportWebApp
- Zhichun Ning: developed the iOS app (user register, sign in, send messages to server and receive response)
  Code: https://github.com/bjtuhfz/LeakageReport-iOS
  Code

II. Introduction

Water is a critical resource to and lifeline service to communities worldwide. The water infrastructure including water generation, treatment, storage and distribution. However, underground pipelines damage is often hidden. The statistics shows that it is close to 237,600 breaks per year in the US which leads to drought and revenue lost. Smart phones and mobile network technology develop fast in recent years. The number of smart phones grows exponentially. Meanwhile, there are millions of phone apps that help people on daily problems. Thus we aim to encourage and facilitate people to receive and report such information via mobile phone apps or web apps.

III. Objective

Develop a middleware system that involves a web app and an iOS app to help people receive and send water pipe leakage events information in a real-time manner and query historical leakage events in different locations from historical Twitter data and extract real-time tweets.
Analyze association between historical temperature, precipitation statistics within Los Angeles, CA (LA) and Montgomery County, MD (MC) and relevant social media posts variation (number of relevant tweets, Flickr posts, etc.) during the last 15 months. We are going to analyze how twitter data can reveal water pipe leaks and how we can utilize this information to improve speed and accuracy of water pipe leaks detection, thus reducing service restoration time.
Functionalities requirement

- Provide interface to clients to report water pipe leakage events
- Notify and encourage clients to report events if a relevant message is detected
- Handle client report
- Build Twitter analyzer: analyze correlation between temperature, precipitation and historical relevant tweets
- Classify and filter real-time related tweets automatically
- Display and query real-time and historical water pipe leakage related tweets at different locations

IV. Related Work

iRain [1] is the first global real-time crowd-sourced rainfall observation system developed through a joint research project developed by UCI and USC. It consists of a database server, an application server and mobile apps for smartphones and tablets using iOS or Android. It supports both public rain observation and on-demand requests through tasks. It aims at improving the quality of real-time satellite precipitation observation.

A. Publish-subscribe Systems

With the development of Internet services, the complexity of distributed network is rising fast. More and more entities appearing makes it an urge demand for a more flexible as well as efficient communication pattern to be applied. Therefore, publish/subscribe technology is then invented to deal with versatile needs of events delivery from publishers to subscribers. The inspiration of the publish-subscribe pattern is from the real world activities. And the main expectations of the pub/sub pattern are to monitor events activities such as pick particular events and configure new incoming events. And the receiving side will not be aware of who is sending it the message. Many obstacles are notified to build this system. First the whole environment is very complicated. Plus, the requirements of different unities are various. Thus many researches are there to provide support on no matter algorithm or structure. Now, pub/sub pattern is a very popular system in web application and mobile application industry practices.

The basic structure of Pub/Sub system is as follows. First, all the events in the system can be classified by their different features. And the primary elements in a pub/sub system are the publishers and subscribers. And an event message is sent when a publisher finds a new event. When an event is published by a publisher, the pub/sub system is capable to deliver the event to the subscribers, who are deemed interested in this kind of event. And the interests of a particular subscriber are gathered and used to define what kind of message should be pushed to the subscriber. The message delivery can be broadcasting in the network, then fetched and processed by pub/sub system on the host side. And also this can be done by group sending to those who have already informed that theyre interested in some typical kind of events [2].

The pub/sub system can be deployed either centralized or distributed. However, due to the high variety of entities and high demands of events, the centralized deployment is not very efficient. Therefore, an approach of using an overlay structure composed by a set of brokers over the current Internet routing system. The vital duty of the brokers is to set up the routing states and keep it running dynamically. Still, a complete pub/sub system should have compatibility to own the capability for different kinds of applications within the web service like various of protocols. Another key part is the event loop, which is responsible for monitoring new event and control the message receiving and processing. Therere three main features of pub/sub system. Space decoupling is to fetch the event message from the publisher and transfer it to the subscribers. Time decoupling is to use a message buffer to orderly deliver. And synchronization decoupling is to separate the publishing and delivery so that they could accomplish without synchronization.

B. Data Analysis

In our project, we are going to utilize sentiment analysis and machine learning techniques to extract water pipe leakage as well as extreme weather events from historical dataset crawled from Twitter over the last four months. Then we can use the models trained from historical data to prevent future extreme weather events and classify if an input tweet is related to water pipe leakage or cracks.

1) Sentiment Analysis: Sentiment analysis (opinion mining) [4], is a computational study of opinions, sentiments, subjectivity, evaluations, attitudes, appraisal, affects, views, emotions, etc., expressed in text. Twitter provides researchers an important information source to predict future events as well as mining popular topic. Businesses spend a huge amount of money to find consumer opinions using consultants, surveys and focus groups. Sentiment analysis is considered to be a popular research topic in NLP, text mining and Web mining in recent years [5]. The natural language text (mainly English in this paper) is often regarded as unstructured data. There are two primary types of opinions, regular opinions and comparative opinions, the former states or expresses opinions or facts about certain target entities, the latter mainly compares multiple entities. The sentiment analysis part, which is extract and classify relevant tweets from input MongoDB source is under the category of regular opinions mining. An opinion can include several components: target entity, entity feature, sentiment value (can be positive, negative or neutral), opinion holder and timestamp when the opinions is expressed. Sometimes due to the complexity and skewness of social media text dataset, it is not an easy task to identify the previous five components clearly and correctly. Aside from these five components, other attributes associated with a tweet are user id, time zone, language set, number of retweets, etc., depending on the application scenario. Sentiment analysis includes the following key techniques: named entity extraction, information extraction, sentiment identification.

2) Machine Learning Based Classification: [6] combines rule-based classification, supervised learning and machine learning into a new combined method, a semi-automatic, complementary approach in which each classifier can contribute to other classifiers to achieve good performance. In machine learning based classification, training set and testing set are required. The machine learning based classification approach focus on optimizing either parameters associated with a model or induced rules with aspect to a set of attribute-value pairs. As the case in the Support Vector Machines based method, it focuses on finding a hyperplane that separates positive form negative samples by learning and optimizing the weights of features. A major limitation of these methods is that it requires a large amount of time to assign significant features and a class to each document in the training set, sometimes labelling the documents by rules or patterns are difficult and not accurate enough to produce good performance for the classifiers. In general, there are four existing methods: Natural Language Processing (NLP) and patterned based methods focus on using NLP tools such as parsers or N-grams to assign a sentiment; unsupervised learning utilizes the search engine corpus to determine the sentiment of an expression; machine learning methods such as the widely used Support Vector Machines (SVM); hybrid classification combines several methods of the previous three categories.

In rule-based classifiers, the set of target words (tokens) is the crucial factor in determining the sentiment of an antecedent, which is a part of a rule, the sentiment is represented by consequent. Thus, a sentence by be represented by a target term and multiple tokens. A set of rules are needed to label a sentiment of a document, in this process, negation such as no, not or never should be taken into account as well.

The SVM finds a hyper plane that separates the two sets with maximum margin (or the least positive distance from both sets). In advance, each training sample should be converted to a real vector. Ideally, the positive samples and negative ones should reside in different sides of the hyper plane. However, in real world datasets, this clear-cut scenario is not possible. To deal with this problem, SVM uses a penalty C to a certain sample contains some features that fall into the wrong side and these features should not dominate the classification decision. If the training set and testing set is too ambiguous or sparse, SVM classifiers may fail. Thus to improve the performance

of SVM classifiers, feature frequencies within each document should be treated as binary and then normalized.

C. Web Frameworks

Python is widely used on web application with lots of web development framework including: Django, Flask, Web2py, tornado and etc. Django is a full stack web application development framework which means it provide comprehensive solution on web development that help improve the development efficiency. Compared to Django, Flask is a micro web framework which means it aims to keep the core simple but extensible. Tornado is not only a web framework but also a web server which means other web application can be deployed on tornado.

Django is developed from real-world project which is a news web application of Lawrence Journal-World in 2003. It provides a programming infrastructure for web application. Django uses MVC design pattern and this theory was published in 1978. The object of this design pattern is to help simplify the modification and extension on program and reuse for specific part of code. The code for defining and accessing data (Model) is separate from request routing logic (Control), which in turn is separate from the user interface (View). A key advantage of this pattern is that the components are loosely coupled which causing each distinct piece of a Django-powered Web application has a single key purpose and can be changed independently without affecting the other pieces. Model can access and process on data, when changes happened it would tell view to show with the pattern defined. Generally, view does not contain complex logic. Control provides user with the ability to handle user request and it helps modify the model, so user can access to view and model. There are four key philosophies of Django. First is loose coupling, it is based on the MVC model that provides clear interfaces between different layers of the framework. Second is that it helps on less code especially by utilizing Pythons dynamic capabilities. Third one is quick web development because it focuses on outcome, not on the details. The last is single replacement for every distinct concept and data which means the code would not repeat. The key features for ORM can be summarized as: 1. Support object-relational mapper (ORM) for dynamic data-access API. 2.polished administration interface for end-users. 3. Elegant URL design for parameter-free URI hiding the technology. 4. Template system providing means to separate design, content and code. 5. Cache support 6. Built-in test framework for doc test and unit tests.

V. Implementation Details

A. System Architecture

Our system has three components, the client, the server and the analyzer/publisher. It provides two interfaces to the clients: the iOS app and web app. The back-end server could process requests from both iOS app and web app. Twitter data is used to build classifier to label incoming tweets as well as user generated messages. The models trained in this process will

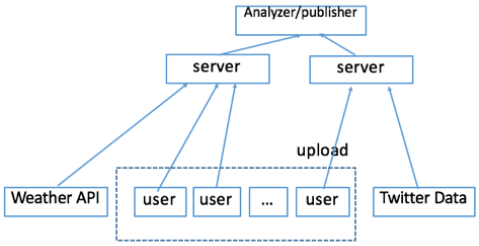


Fig. 1. System Architecture

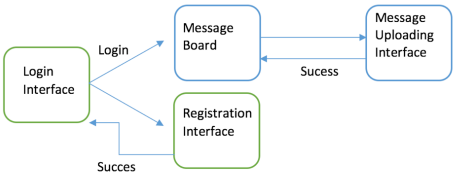


Fig. 2. iOS Architecture

be saved and loaded into analyzer, where the requests can be handled and results will be returned.

B. iOS app functionalities

- User registration, sign in, upload water leak information to server, receive response info

With username and password fields, users can either login or register a new account to access the server. The message board shows the messages sent. The ”Mine” button at the bottom is used to enter the message and send it to the server. Message uploading interface for user to send their message like what they observe to the server. Users can just type text in the texting field or just simply click on the leakage button at the bottom to make a quick report with location.Fill the message, enter the location and then click the ”send” button on the top right, and the users report is sent to the server. When the user clicks on the ”low” button on the Leakage column, the text field becomes Low Leakage immediately to facilite user reporting. Below are the coresponding UI of iOS app.

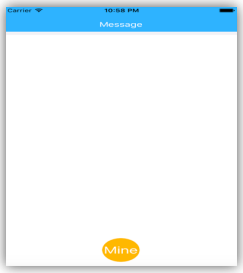
C. Web app functionalities

- User registration, sign in, view/vote for historical messages, handle messages uploaded via Web & iOS app
- Display and search real-time historical tweets and user generated posts
- Google map API is used, a marker will be added to the query location, an information window will pop up displaying the number of relevant tweets found.
- Tweets can be viewed as list and table, by default, 20 most recent tweets will be shown when index page is loaded.

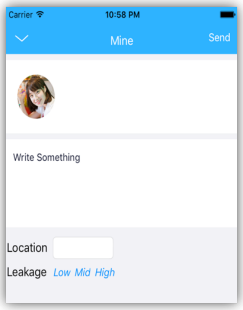
Our web interface uses Bootstrap template and web server is based on Django. Users can use the web app to send message

(a) iOS register

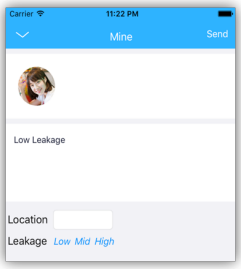
(b) iOS login



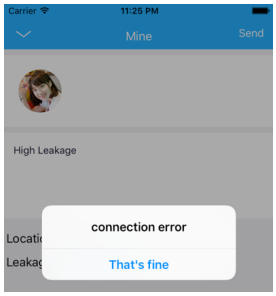
(c) iOS message board



(d) iOS message upload

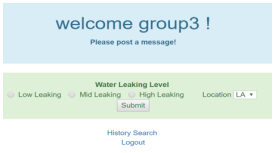


(e) iOS upload button

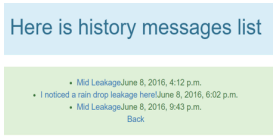


(f) iOS error checking

Fig. 3. UI of iOS App

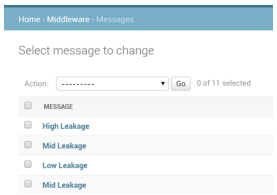


(a) web upload message



(b) web message history

(c) web vote for message



(d) web admin

Fig. 4. UI of Web App

to server by Http request packet, then server will manage this message, update the database and send Http response

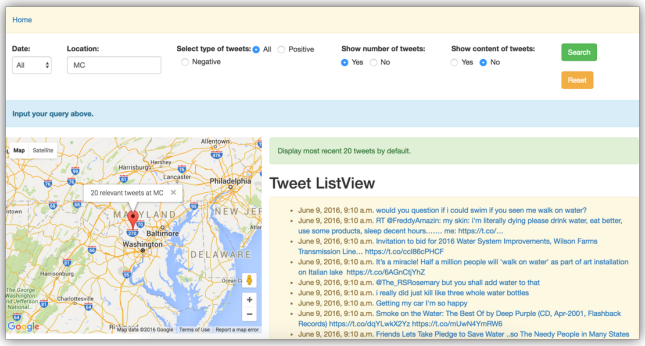
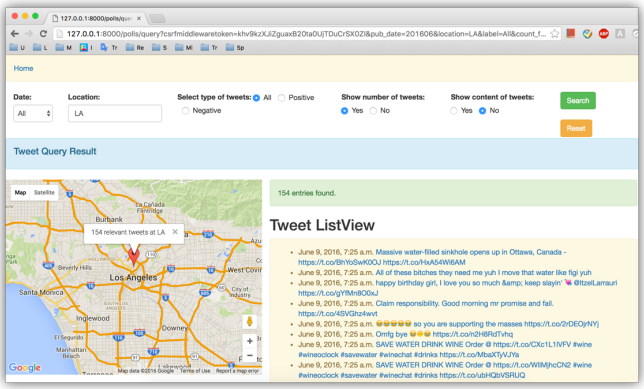


Fig. 5. UI of Web Query



(a) web query result list view

Tweet TableView

tweet_id	tweet_text	pub_date	location	label
3773	Massive water-filled sinkhole opens up in Ottawa, Canada - https://t.co/BhYoSwK0OJ https://t.co/HxA54Wi6AM	June 9, 2016, 7:25 a.m.	LA	Negative
3774	All of these bitches they need me yuh I move that water like figi yuh	June 9, 2016, 7:25 a.m.	LA	Negative
3775	happy birthday girl, I love you so much & keep slayin' https://t.co/gYfMn8O0xJ	June 9, 2016, 7:25 a.m.	LA	Negative
3776	Claim responsibility. Good morning mr promise and fail. https://t.co/4SVGhz4wvt	June 9, 2016, 7:25 a.m.	LA	Negative
3777	😂😂😂😂 so you are supporting the masses https://t.co/2rDEOjrNYj	June 9, 2016, 7:25 a.m.	LA	Negative
3778	Omfg bye 😭😭😭 https://t.co/n2H8RdTvhq	June 9, 2016, 7:25 a.m.	LA	Negative
3779	SAVE WATER DRINK WINE Order @ https://t.co/CXc1L1IVFV #wine #wineoclock #savewater #winechat #drinks https://t.co/MbaXTyVJYs	June 9, 2016, 7:25 a.m.	LA	Negative

(b) web query result table view

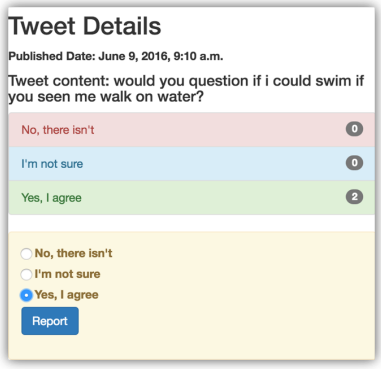


Fig. 6. UI of Web Vote for Tweet

back to user. For different system users, the system should achieve their goals individually. This function area assembles different users with different authorities. The administration user obviously has totally all permissions, for simple reason that he can manage the web site. The clerk in the shop is the second largest authority owners of the system. Then the user is also the system users, they can send and messages. Database contains three tables, namely users table, the message table, tweets table and votes of tweets table. The users table manages user registration and verification. The message table is associated with the users, deals with the messages generated by users. The tweets table stores the historical and real-time tweets as well as the labelling result of each newly inserted tweet. The votes table records the number of votes each tweet or message receives during reports of users. Every user has the authority to read message history and vote for suspected messages/tweets to add credibility to an event. They can see the final voting results and then respondents take advantage of these information and deal with them on time.

After the combination of the *Message*, *User* module and *Tweet*, *Vote* module, our integrated system is able to receive messages sent from both the web and iOS app by HTTP request. Utilizing the tweet classifier built based on historical Twitter dataset, we can display the newly submitted user messages and its labelling result, which is relevant or irrelevant to water pipe leakage event. Test cases are also written to test the query engine, upload module and synchronize message module.

D. Twitter collector and analyzer

- Analyze correlation between temperature, precipitation and historical relevant tweets

We use Twitter Stream API, Search API and REST API to get real-time, historical tweets respectively, on one hand, we fetch tweets from these APIs, on the other hand, the fetched tweets are inserted into the *tweets* table of the sqlite database. Before we begin to analyze tweets, we follow the text mining rules step by step to perform data cleaning, tokenizing, word vectoring, etc. During the process of accessing real-time tweets, we add "water" as keyword and Los Angeles and Montgomery county as location ranges to filter the real-time tweets, otherwise, too many irrelevant twees will be returned and take up a lot space unnecessarily. The Search API is used to search for past tweets given a set of keywords and location and time range constraints, since past tweets older than a week cannot be loaded from REST API. After this, we can proceed to build classifier.

Filter the US district Twitter dataset from Qings MongoDB tweets collection, according to Mehdi, his Twitter Acquisition System (TAS) [3] collects English tweets relevant to a input topic all over the web, all possible regions around the world, based on the number of English words detected within a tweet. It will extract input keywords and compare the pattern with current tweet then label it as relevant or irrelevant. As the data collection process goes, the pool of topic related word features will change and update, taking advantage of auxiliary information from Wikipedia, to decide if a certain word feature should be associated with the input topic or not.

I set the utcOffset field value of a document in the MongoDB to be in the range of -32400 and -18000, which spans the Alaska Time Zone, Pacific Time Zone, Mountain Time Zone, Central Time Zone and Eastern Time Zone. The time range is from Jan. 6th to Mar. 31st, 2016, collected by TAS over the last 3 months period. Also, I used a combination of rule based method and machine learning method to filter the relevant tweets from the downloaded dataset. The rules I have used include water p̂ipe, water p̂ipe l̂eak, since water pipe leakage is a very specific topic, with such words in a tweet, the tweet has a higher rate of being labelled as relevant. Then three ML classifiers are applied respectively to train the manually labelled datasets and one of them is chosen based on the best performance (accuracy, recall and F1 score).

A correlational analysis is conducted based the on the number of relevants tweets filtered via the keywords set extracted from the SVM model (with linear kernel) and a set of target topics, including the precipitation amount (rainfall) over the period from January to March 2016 in Los Angeles area, the temperature data (extreme cold weather) overee the period from January 2015 to March 2016, the reported number of breaks in Montgomery County (MC) from January 2015 to March 2016, the time interval is set to be one month. More details will be covered in section *Evalution Results*.

E. Twitter classifier

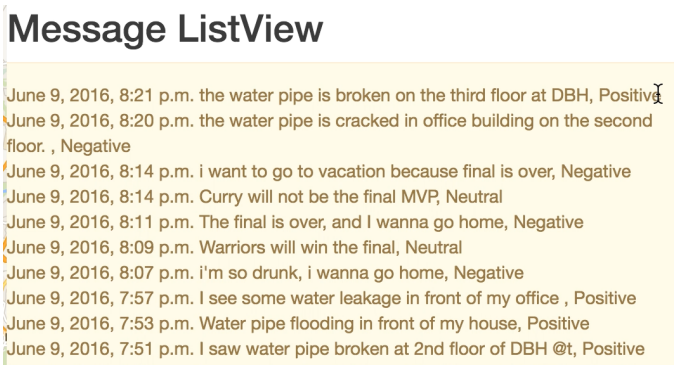
- Label real-time tweets and user reports automatically with machine learning model (Nave Bayes, SVM, etc.)

We trained a supervised machine learning model based on SVM. We use the last three months tweets data (labelled as relevant or irrelevant to water pipe leakage each, assigned positive/negative values in the "label" field of each tweet entry in the table) to train our classifier. We also used other models including Naive Bayes and Maximum Entropy, here we choose SVM because it gives the highest accuracy and recall. The Python libraries we have used in this module include *nltk*, *sklearn*, etc. Building classifier starts from teh creation of training and testing sets based on historical dataset, then tuning parameters according to the output accuracy and precision. Finally we do evaluations based on accuracy and recall and select SVM as the classifier for the incoming real-time tweets from Los Angeles and Montgomery county.

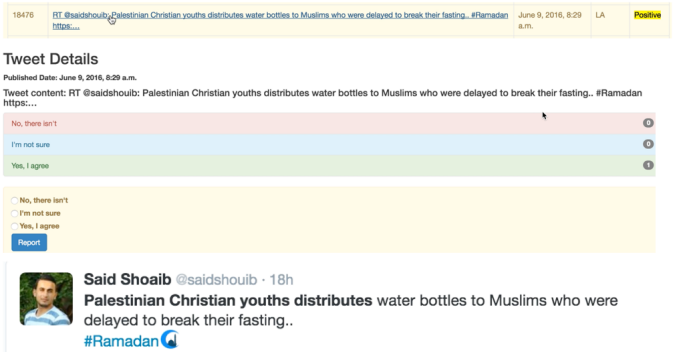
VI. Evaluation Results

A. Correlation Analysis

We use correlation coefficient to measure the association between the target variables. In the training model process, accuray and recall is used to determine the classifier with best performance. Thus we have calculated a series of correlation coefficients between number of relevant posts (from Twitter or Flickr) and target topics, including temperature (extreme cold weather), precipitation (rainfall), repored leaks (ground truth), etc. The following tables show the result we obtain from the above analysis.



(a) web synchronize messages from iOS app



(b) web real-time tweets classifying

Fig. 7. UI of Web Real-time Display

A series of analysis were conducted about the association between the amount of social activities like tweeting and water pipe leakage events as well as cold weather and rainfall in LA and Montgomery County areas from Jan. to Mar. 2016.

We dive into one example here, the anlysis of *Rainfall in LA in January 2016 and Relevant Tweets*, results are shown in Table I. Due to the el nino effect, the west coast of US has much more precipitation in January than February and March in 2016. We collected the tweets in LA area near 30 miles during the January period and partition the number of relevant tweets in four weeks, got a correlation coefficient of 0.99 between the number of rainfall related tweets per week and the amount (in inch) of rainfall in LA. For example, at the 1st weekend of January, heavy rainfall (a total of 2.72 inchs precipitation) was reported. Correspondingly, a surge of relevant tweets (755 tweets) were found too. Then in the following three weeks, rainfall is much less, average number of relevant tweets also drops to less than 100. Based on the dicussion above, the number of relevent tweet or Flickr posts are both highly relevant to the temperature trend over the last 15 months' period in Los Ageles (LA) and Montgomery County (MC) regions.

The results of cold weather analysis in MC, 2016 is shown in Table II and all the other correlation analysis result is shown in Table V.

TABLE I
RAINFALL IN LA, JAN. TO MAR. 2016

LA	Jan	Feb	Mar
Rel. Tweets	938	364	615
Temperature/F	44.6	51.8	52.5
Precipitation/"	6.4	1.2	4.88

TABLE II
COLD WEATHER IN MONTGOMERY, JAN. TO MAR. 2016

Relevant Tweets	Jan	Feb	Mar
US	331403	50081	2268
LA	1922	323	12
MC	57988	7758	355

TABLE III
PIPE LEAKS IN MC, 2015

MC	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec
Rel. Tweets	397	459	225	227	184	181	191	248	170	193	224	190
Rep. Leaks	410	412	121	55	97	91	84	85	88	114	300	240
Avg. Temp.	30.7	25.3	39.4	54.1	67.6	73	75.8	73.9	70.7	55.5	50.5	48.6

TABLE IV
PIPE LEAKS IN MC, 2016 AND FLICKR

MC	Jan	Feb	Mar
Relevant Posts	817	509	351
Temperature	32.1	36.9	49.4
Precipitation	3.13	4.09	1.82

TABLE V
CORRELATION COEFFICIENTS

Source	Variable 1	Variable 2	Correlation
Twitter	MC Temp. 2016	Relevant Posts	-0.79
Twitter	LA Temp. 2016	Relevant Posts	-0.81
Twitter	MC Temp. 2015	Relevant Posts	-0.77
Twitter	MC Reported Breaks 2015	Relevant Posts	0.82
Twitter	LA Precip. 2016	Relevant Posts	0.95
Flickr	MC Temp. 2016	Relevant Posts	-0.91
Flickr	MC Precip. 2016	Relevant Posts	0.42

B. Classifier Performance

In terms of classifier performance, the Nave Bayes and SVM classifiers can function normally when classifying a single input tweet. Due to the limited number of positive tweets (strictly talking about water pipe leakage or cracking event, gained from the dataset, there are only about 600 samples in the training set, positive and negative samples take up half respectively). I classify water pipe protection or water conservation related projects with hash tag as negative. According to previous experiments, the performance of Max Entropy is not very good, due to the high imbalance of training

TABLE VI
TWEETS CLASSIFICATION RESULTS (GLOBAL)

All (positive/negative)	Ground Truth	NB	SVM
01/06-01/31	530/14588	1595/13523	496/14622
02/01-02/28	100/2356	395/2061	97/2359
03/01-03/30	92/1481	245/1328	89/1484

TABLE VII
TWEETS CLASSIFICATION RESULTS (US)

All (positive/negative)	Ground Truth	NB	SVM
01/06-01/31	210/6043	529/5724	208/6045
02/01-02/28	37/912	125/824	37/912
03/01-03/30	42/412	72/382	42/412

TABLE VIII
TWEETS CLASSIFICATION RESULTS (MC)

All (positive/negative)	Ground Truth	NB	SVM
01/06-01/31	94/2750	243/2601	94/2750
02/01-02/28	22/432	69/385	37/912
03/01-03/30	42/412	72/382	42/412

set (too many negative samples than positive ones). SVM achieves the best performance of an accuracy of 0.99, which can well capture the features of water pipe leakage related tweets. Detailed results can be found in Table VI, VII, VIII.

VII. CONCLUSION

Within the context of our social media based data collection and leakage alerting and querying app, before any of the previously described classifiers are applied, the input dataset should be tokenized and cleaned, duplicates should be removed due to the numerous retweets. The set of parameters used in our Nave Beyes, Max Entropy and SVM based models should be tuned to gain the maximum F1 score, since the tweets classification results will have a great influence on our future water pipe leakage prediction task. Asides from the classification and prediction script written in Python, Django framework is also used to accomplish the goal of providing our iOS app users the efficient way to get classification results as well as alerts sent by our server.

We analyzed historical tweets during last 15 months at Los Angeles and Montgomery county as well as these two locations' temperature and precipitation statistics data, then found that people social media activity are highly relevant to our target events including rainfall, extreme cold weather and water pipe leakage, with a correlation factor over 0.8. Based analysis of the relationship between the above topics and number of related tweets, we can safely draw the conclusion that we can utilize historical and real-time tweets to analyze water pipe leakage problem in depth and extract exact locations provided by user to better serve the repondents to handle such events.

In general, people may only care about their own house and working places water facility, excessive alerts or notifications within a higher level of region only ends up with annoying subscribed users. More personalized notification system should be built to satisfy the real needs of users as well as promote social media and IoT awareness.

VIII. FUTURE WORK

Integrate information from weather API to better detect real-time water leakage events. Build an Android app to enlarge users group. Add one click to report functionality to take users geolocation automatically and take pictures and upload image.

ACKNOWLEDGMENT

The authors would like to thank Prof. Nalini Venkatasubramanian's ongoing guidance and useful advice during the development of our project and Qing Han's historical Twitter dataset from Jan. 2016 to Mar. 2016.

REFERENCES

[1] https://play.google.com/store/apps/details?id=irain.app
[2] Publish/subscribe systems: design and principles, Tarkoma, Sasu, John Wiley & Sons, 2012.
[3] https://github.com/mehdiesadri/tweet-acquisition
[4] https://www.cs.uic.edu/l̃iub/FBS/Sentiment-Analysis-tutorial-AAAI-2011.pdf
[5] Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." *Foundations and trends in information retrieval* 2.1-2 (2008): 1-135.
[6] Prabowo, Rudy, and Mike Thelwall. "Sentiment analysis: A combined approach." *Journal of Informetrics* 3.2 (2009): 143-157.