Detecting Informative Messages from Twitter feed during Catastrophes

Akash Malla
Santa Clara University
amalla1@scu.edu

Samarth Mehta
Santa Clara University
smehta1@scu.edu

Table of Contents

Acknowledgements

We thank professor Ming-Hwa Wang, our family and friends for their support

Preface

The purpose of this project is to provide valid information regarding any catastrophe that occurs around the world. We do not want people to believe in fake donations or browse through spam information from social media.

Abstract

Social Media has dramatically transformed the way we communicate and gain knowledge from general public. It is very accessible via our smartphones and other devices we carry or use daily. Keeping this in mind, in order to communicate regarding crisis situations especially natural calamities, there is a big spike in continuous feed of real-time information shared on Twitter. People actively login to twitter to either share some an opinion or to find out the latest update regarding the disasters such as earthquakes, floods etc. However, along with rich credible information, there is a lot of unreliable, unvalidated tweets that are often trending and promote wrong facts. In recent well known catastrophes, for example, regarding Boston Marathon blasts there were about 7.8 million tweets collected through twitter API [3]. So, this paper provides information on how we will tackle this problem.

Introduction

Objective

To build a naive bayes classifier based hierarchical classification algorithm

What is the Problem?

There is a dire necessity to detect legitimate and useful tweets and distinguish from spams, rumors and other fake information. A fake tweet is clearly a misleading information and it can lead to emergency personnel addressing wrong issues. This is the foremost step in making sure that the information that can be acted on is true in nature.
Furthermore, it's important for emergency responders to attend to critical information first (for example, casualties or critical damage). For this, information needs to be categorized into type of content. Repeating information is also one of the roadblocks for information extraction. If a tweet is from a source, then it has to identified to be not acted upon multiple times.

Why is this project related to class?

Data mining is essentially to try to find some new information in a large data set. Similarly, through Twitter API data we get different types of content. Essentially, the informative content needs to be filtered out from the vast amount of tweets. This process requires classifying data into informative and noninformative sets. Informative tweets often have repetition, for example, tweets from news stations being picked up and retweeted. Our goal is to find these unique tweets before categorizing them, this approach saves a lot of processing time.

Why other approach is no good?

Generally when data is classified we use keywords such as n-grams. It works efficiently to find text sequences but our question is inclined towards categorizing content first and then providing valuable findings to a party to analyse. Our approach deals with essentially filtering of a topic related data. This can't be done with other algorithms that are commonly used. Naive Bayes Classifier remains one of the very popular choices for text categorization but seldom used to categorize tweets. We think that it can improve emergency response through carefully classifying abundance of data available and provide only the "informative" nuggets.

Why do we think our approach is better?

In this project, our aim is to provide emergency information that people can trust relevant to crisis situations. Whether it is a natural disaster such as earthquakes or some catastrophes such as terrorist attacks, we want to let actionable information pulled from Twitter APIs after

removal of non-legitimate tweets to help recover from these situations as quickly as possible. For example, we would want to know which organization is trustworthy to have general public donate money. Inspired by authors who proposed detection of fake and spam tweets [1], information nuggets [4] and Boston marathon blast fake twitter content article [3], we intend to collect, analyze and present legitimate information to the appropriate people as a tool for emergency service. Instead of manually putting effort to tag a tweet to a certain class, we would like to automate this process and build a model that learns from previous results. In order to classify malicious content from legitimate the tweets, we chose naive bayesian algorithm for two reasons: (i) it will save manual effort to build a large training set and (ii) Adding a two step process using hierarchical classification increases accuracy. Although other approaches have used hierarchical classification, they did not have a way to build training and test data set in a simpler and faster way. [5].

Area of scope of investigation

Twitter Stream

Theoretical bases and Literature review

Definition of the problem

To be able to model hierarchical classification from real time twitter data in order to retrieve helpful information for general public.

Theoretical background of the problem

One of the papers we read for finding the best approach focused on finding information nuggets. It talks about two main steps in finding the informative information through hierarchical classification. First, it tries to categorize tweets between personal, containing information and others. Once it is found that a tweet is informative, it further classifies into Caution, Source, Donation or Damage categories. This lets the emergency responders attend to only the required categories. [4]

In the Fake and Spam Messages paper, they used flat classification and hierarchical classification. Among the two approaches, hierarchical classification provided more accurate results. As a first step in hierarchical classification, authors classified legitimate and non-legitimate tweets. In the second step, they classified non-legitimate tweets into fake and spam tweet for which the authors chose FT, NBTree and Random Forest classifier based hierarchical classification.

Advantage/disadvantage of those research

Advantages:

Fake and Spam detection paper used decision tree related classifiers which provide very accurate results.
In the information nuggets paper, the approach used to classify in two steps leads to accurate results. It matches the hit ratio of human extractors in categories like casualties and source.

Disadvantages:

Usage of decision trees will involves having a bigger training set and for every new event you would need to rebuild the whole tree depending on features as the features keep changing depending on the catastrophe.
The information nuggets paper shows that categorizing tweets for caution and damaged objects is poor compared to human extractors. Even though it's able to extract the some information related to damage, the nature of text is highly changeable when describing a damage. This leads to poor precision and recall in this category.

Our solution

We focus to use hierarchical classification approach which is a supervised machine learning technique that can classify a tweet by providing user-defined classifiers. Since tweets mainly consists of textual data and hashtags along with pictures and links at times, implementing Naive Bayesian text classifier is a right solution. We try to hierarchically categorize tweets to lead us to informative-only tweets. Once a tweet is identified as Help or Caution, we can provide this information to two different target audience. Caution can be

How our solution is different?

We are using naive bayesian classification so that we can work with a smaller training set if we have to and we can control the accuracy of our solution we can do with this approach.

Why our solution is better?

In many application for classifying textual data especially email spam detection, naive bayesian algorithm has been a good approach. The fake and spam detection paper mentions to have used two human labelers who have independently classified around 1050 tweets to be either fake, spam, or legitimate for Hurricane Sandy and Moore Tornado events [1]. This has two problems, firstly, we do not have any clue of the background of these labelers in order to evaluate their way of labeling a tweet is trustworthy or not. Secondly, it is a time taking process to manually label each tweet, they have no choice but to label a large number of tweets as the

algorithm requires a bigger training set than naive bayesian would require. Also, rather than using a decision tree based classifier which other approaches have taken, we thought a simpler and faster solution which does not sacrifice accurate results is a more effective solution for this application as providing actionable information immediately after a natural disaster or catastrophe is very important.

Hypothesis

If significant accuracy is observed in our solution, it can be used to monitor live tweets as an aid to disaster recovery authorities. The output of our solution can be used as a feed to warn people (via a blog or website) once we classify them as Caution/Warning. As we have seen from results from previous papers, sources can be found accurately, which can provide valuable stream of local information tweets. The hit ratio of human extractors can be matched for casualties using our methodology. This can be used as a tracker for missing people.

Methodology

How to generate/collect input data

Our application assumes that there is a constant flow of incoming data regarding an event after it occurs. Twitter provides an interface called Search API to retrieve tweets in real time. This API allows to get user related data satisfying some conditions based on keywords, location and other such parameters in reference to a catastrophe. The data will also go through redundancy check by implementing regular expressions and picking a random sample of tweets to be compared to existing tweets. Once data is collected, we focus to apply the hierarchical classification algorithm by doing the following: (i) classify tweets to either informative or noninformative. (ii) Informative tweets are further classified to help required or caution and advice or other categories

How to solve the problem

Given numerous amounts of tweets, we categorize each tweet by its characteristics. Below is the definitions for two categories that is done in the first part of hierarchical classification:

- Noninformative: A tweet is considered noninformative if it consists of the following.
    - Opinion on the disaster. Message is not convey to others but its author only and immediate friend circle
    - Lack of source or incorrect location, time, etc
    - Non-english or repetitive tweets if any

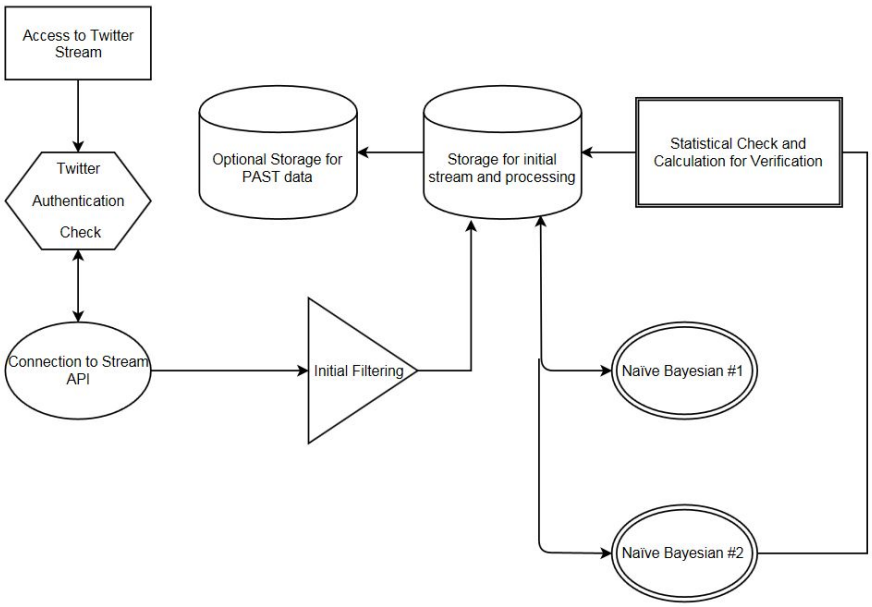
- Informative: A tweet that has a message conveyed to everyone and consists of the following.
    - Has direct eyewitness information of what is being taken place
    - Message specifies a source such as seen or heard from online, television, radio, or other sources of information.

Once the tweets are categorized into the above two categories, we further categorize informative tweets into two categories that we believe are the most important out of informative tweets that needs people's attention. Although non informative tweets are not classified further, we ensure to carefully work on it because we do not want incorrect information to be shown. In order to take care of that, we will check the number of existing tweets for the same location of an event, if that does not match then we will discard that tweet to be noninformative. Once we have retrieved informative tweets with certain confidence threshold set in the naive bayesian classifier, below are the two categories of informative tweets we classify:

- Help Required: An informative tweet which specifies the following.
    - Donation, goods and services
- Caution and Advice: An informative tweet that refers to be cautious.

GeoNames API is an alternative in identifying the location from tweets lets us filter out tweets that were sent from unusual location, which indicated that the person who posted is not a trusted source on the ground. This significantly helps in taking only the local information into account primarily. [4]

Implementation



The flowchart above explains the overall architecture of our application. The following steps explain the implementation:

Step 1 : Created Twitter Application for Authenticated use of Twitter Feed

- Owner : samarth128, Owner ID : 28956455

Step 2 : Authentication and Connection to API

- Consumer + Authentication Key & tokens passed and checked at runtime by the Twitter API

Step 3 : Initial Filtering

- Retweets : Checking for String “RT” present in the received tweet, after making sure that original tweet exist
- Difflib Library : Computing Delta to check if “similar” tweet already exists
- Similarity threshold for Difflib : 0.7

Step 5 : Parsing Twitter JSON

Sample JSON:


```
"created_at": "Fri Jan 23 23:57:36 +0000 2015",
"id": 558775589612437504,
"id_str": "558775589612437504",
"text": "Thanks, polar vortex: Attendance dips at major Chicago museums in 2014 http:\/\/t.co\/jQlLEurk9s http:\/\/t.co\/3bss2nGemx",
"source": "\u003ca href=\"http:\/\/twitter.com\" rel=\"nofollow\"\u003eTwitter Web Client\u003c\/a\u003e",
"truncated": false,
"in_reply_to_status_id": null,
"in_reply_to_status_id_str": null,
"in_reply_to_user_id": null,
"in_reply_to_user_id_str": null,
"in_reply_to_screen_name": null,
"user": {
  "id": 7313362,
  "id_str": "7313362",
  "name": "Chicago Tribune",
  "screen_name": "chicagotribune",
  "location": "Chicago, IL",
  "url": "http:\/\/www.chicagotribune.com\/",
  "description": "Chicago Tribune news, features and so much more live from our newsroom. A part of your life since 1847.",
  "protected": false,
  "verified": true,
  "followers_count": 321548,
  "friends_count": 523,
  "listed_count": 8074,
  "favourites_count": 34,
  "statuses_count": 47367,
  "created_at": "Sat Jul 07 14:10:07 +0000 2007",
  "utc_offset": -21600,
  "time_zone": "Central Time (US & Canada)",
  "geo_enabled": false,
  "lang": "en",
```


- Fields extracted from JSON :
    - Text, URL, Hashtags, Retweet Count, Follower Count, Geo Enabled, Created at
- Storing Retrieval of Data (.csv)

Step 6 : Naive Bayesian run#1 and run#2

- Used sklearn module's term frequency and inverse document frequency API to compute importance of every word in all tweets.
- First run predicted tweets to be classified as informative or noninformative.
- Second run predicted tweets to be donation/help related or caution/advice related

Data Analysis

Example #1 : Taking 1 Alberta Dataset for training and testing, we get the following Bayes accuracies,

Related vs **Non Related** : 73%

Train Data : 983

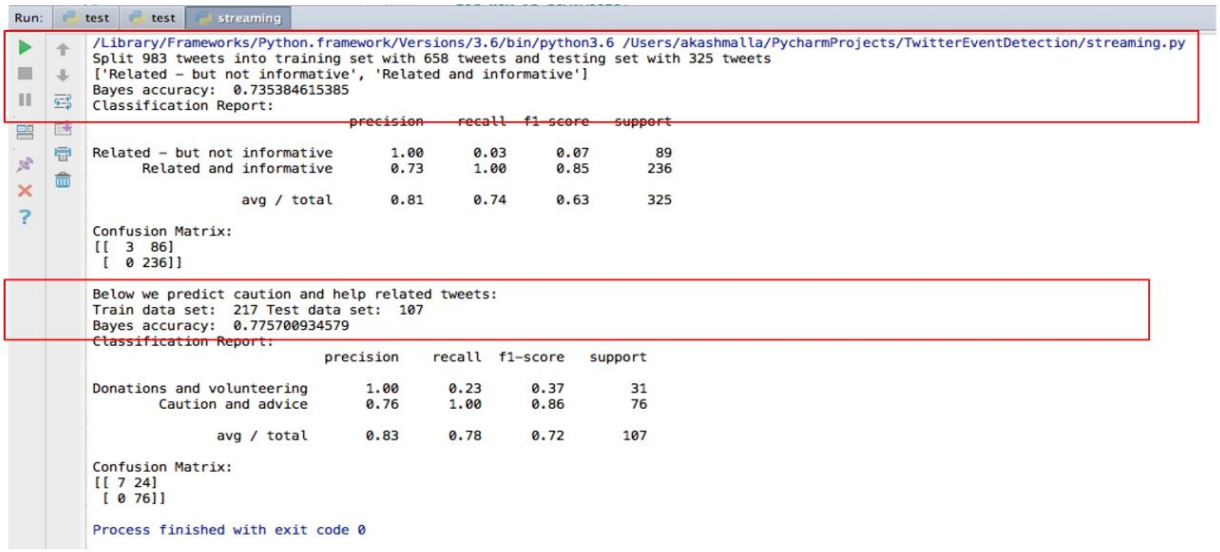
Test Data : 658

Donations and Volunteering vs **Caution and Advice** : 77%

Train Data : 217

Test Data : 107

Analysis : Accuracy is acceptable for the first pass of Naive Bayes Algorithm to categorize informative and non-informative tweets. The second pass showed higher accuracy



Example #2 : Taking 5 past Calamities Datasets for training and testing, we get the following Bayes accuracies,

Related vs **Non Related** :79%

Train Data : 5580

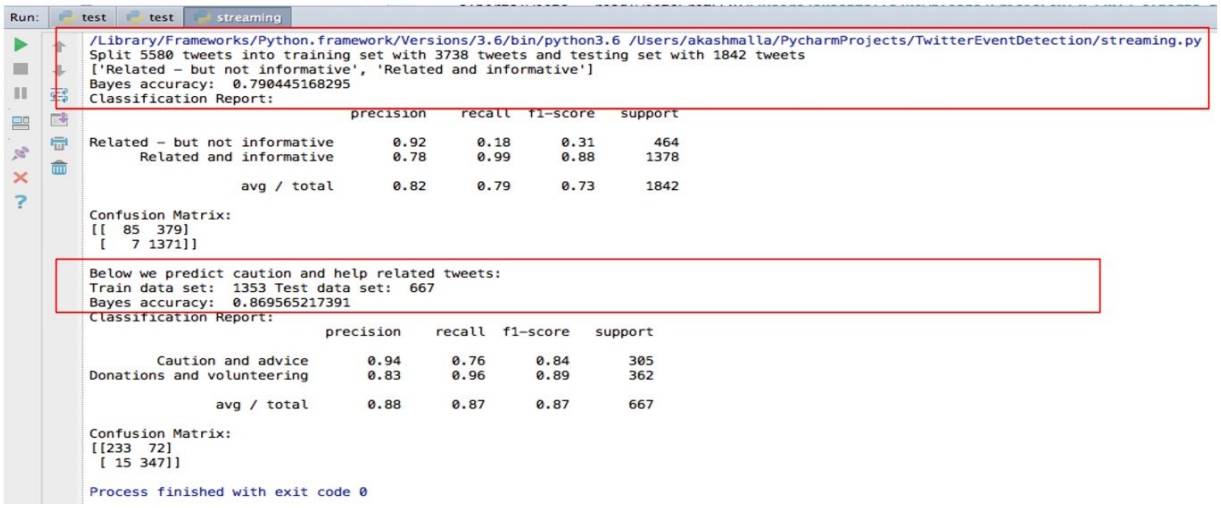
Test Data : 3738

Donations and Volunteering vs **Caution and Advice** : 87%

Train Data : 1353

Test Data : 667

Analysis : Accuracy improves significantly when we take multiple datasets are used for training.



Example #3 Changing the number of categories,

Analysis : Instead of using 2 categories, if we use 4 categories, we get lower precision for each. However, logically that makes sense, because when taking only two category, it's mixing up multiple category tweets into one.

	precision	recall	f1-score	support
Caution and advice	0.94	0.76	0.84	305
Donations and volunteering	0.83	0.96	0.89	362
avg / total	0.88	0.87	0.87	667

	precision	recall	f1-score	support
Donations and volunteering	0.84	0.66	0.74	250
Infrastructure and utilities	0.70	0.71	0.70	287
Affected individuals	0.67	0.91	0.77	389
Caution and advice	0.74	0.41	0.53	210
avg / total	0.73	0.71	0.70	1136

Example #4 San Jose Flood Tweets Statistics, tweets Observed over past 7 days:

Limit : 100
Query Results containing "San Jose Floods" : 68
After 1st Filtering (RT, Duplicates) : 36

Category Results:

- Donation - 15, Caution - 6, Other - 13, Sympathy - 2

Tweet: Don't stop finger-pointing: "The Coyote Creek flood is too significant a failure to worry about hurt feelings." https://t.co/tDXkfVGUHI . Prediction: Sympathy and support

Tweet: Come out and donate this weekend at Olinder Elementary!! Donation Drive! Coyote Creek Flood Relief Fundraising... https://t.co/kRN4pCNJFG . Prediction: Donations and volunteering

Tweet: Public Hearing Held In San Jose To Discuss Response To Coyote Creek Flooding https://t.co/A3RVB26Hgh #sanfrancisco . Prediction: Other Useful Information

Tweet: VIDEO: San Jose declares shelter crisis amid devastating Coyote Creek flood https://t.co/NevAb8Z0Up via @kron4news . Prediction: Caution and advice

Conclusion

Positives:

- Naive Bayes is very fast for text classification
- With a large enough dataset, accuracy can be as high as,
  - Informative - 79%
  - Caution vs Help - 86%
- Upon wise selection of Categories for classification results precision is as high as
  - Donations - 94%
  - Caution - 83%
- Multiple CSVs to ensure higher accuracy

Negatives:

The accuracy of our solution is dependent on the tweets that we receive. Each classification label has a corresponding confidence score, if the confidence score is not at the right balance then we may have either lot of false positives or false negatives. Also, some additional challenges that we faced were,

- 7-10 Days limit for Tweets History:
  - Twitter doesn't let you extract topic-based tweets after this duration
  - Storing data older than 7 days becomes crucial
- Continuous Data Storage :
  - Bigger Storage and Processing requirements to use it as a Live Application
- Intended Spam to go past filtering:

        - Eg. Victoria Beckham's winter protection styling choice during New York Snow Storm
        - Use of Mirror Links so duplicate checker fails
    - Twitter Error 420 & 429 : Overuse of Set Limit of Twitter Resources
        - Eg. Trying to fetch thousands of 'past' tweets continuously breaks connection from Twitter's side
    - Tweepy API :
        - Often timeouts and needs to be reset when the stream is longer than a few hours

Future Works

- Parallel Processing:
    - Can reduce overhead of storing to disk and reading it back for filtering
    - 2 processes in parallel:
        - Handler for incoming stream (Initial Filter + Storage)
        - Hierarchical Classification to categorize
- Geo-based additional filtering to extract tweets from hotspots only:
    - Using GeoNames origin of the source can be found and filtered which let's us focus more on local tweets as they might have more recent and eyewitness news.
- Web-based application that outputs the result of Classification
    - Emergency Forces
    - Authenticated Stream of Live Events
- Connecting it to a Blog/Page on social media

Bibliography

1. M. Rajdev and K. Lee, "Fake and Spam Messages: Detecting Misinformation During Natural Disasters on Social Media," *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Singapore, 2015, pp. 17-20.
2. Himanshu Shekhar, Shankar Setty, Uma Mudenagudi, "Vehicular traffic analysis from social media data", *Advances in Computing Communications and Informatics (ICACCI) 2016 International Conference on*, pp. 1628-1634, 2016.
3. Vishal Kumar, Kunwar Singh Vaisla, Jaydeep Kishore, "Analyzing Email Account Creation: Expectations vs Reality", *Communication Systems and Network Technologies (CSNT) 2014 Fourth International Conference on*, pp. 597-600, 2014.
4. Muhammad Imran, Shady Elbassuoni, Carlos, Fernando Diaz, Patrick Meiser, “Extracting Information Nuggets from Disaster-Related Messages in Social Media”, *ICIS - International Conference on Information Systems, 2013*