A SOCIAL COMPUTING SOLUTION TO DISASTER RELIEF

An Undergraduate Research Scholars Thesis

by

YANG YANG

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. Xia Hu

May 2018

Major: Computer Science

TABLE OF CONTENTS

ABSTRACT

A Social Computing Solution to Disaster Relief

Yang Yang
Department of Computer Science and Engineering
Texas A&M University

Research Advisor: Dr. Xia Hu
Department of Computer Science and Engineering
Texas A&M University

Disaster relief has chronically been a major issue, and various solutions have been presented, attempting to provide the best relief. Currently, disaster rescue teams are facing the problem of lack of valid information at the rescuing scene, resulting in worse relief and more casualties. Data analytics on social network has received success in multiple other application like spam filtering and trend prediction, showing its potential in the field of disaster relief. Previous attempts like TweetTracker has shown the potential of data analytics in disaster relief, with a few potential improvements like expanding the size of the dataset and including a more detailed map. The purpose of this research is to expand on previous applications and use social media data to generate a detailed disaster situation map for first responders. With validated information about the disaster, both survivors and rescuers can pinpoint hazardous areas and avoid further damage.

ACKNOWLEDGEMENTS

I would like thank Dr. Xia Hu for being my undergraduate research advisor and offering guidance throughout this research. I would also like thank Dr. Ali Mostafavi and his student Aayush Gupta and Chao Fan for their inspiration and knowledge, as well as my teammates(Micheal Peterson, Ehab Abo Deeb, Sriram Natarajan, Nandan Gade, Nicolas Carvajal, Uyen Pham, Audi Putera, Justin Nguyen, Trey Shaffer and Tiffany Zhang) in the Aggie Challenge team. I would also like to thank Amazon Web Services for their support.

CHAPTER I

INTRODUCTION

Motivation

This research aims to provide a solution to the problem of restore and improve urban infrastructure, a grand challenge of engineering posted by the National Academy of Engineering [1]. Disaster relief has become a major issue in research of urban infrastructure, where rescuers are often restrained from conducting rescues more efficiently by the lack of detailed information of the scene. As a response to the situation above, our research attempts to use social network data provide more information for first responders and civilians during disaster.

Previous Researches

Using social sensing in disaster relief is not entirely new. TweetTracker, a research conducted in Arizona State University, presented a sophisticated structure of a disaster awareness software based on Twitter data, which divided the software into a crawler, a database, and a visualization interface [2]. Another notable research at Louisiana State University used a similar structure, but with the addition of frequency analysis and content analysis to provide a visualization of tweeting intensity and global/local trend [3].

Improvement

Compare to previous research projects, our research aims provide a more detailed live representation of disasters. With help of machine learning methods, our project aims to implement a prediction-based analyzer to recover the disaster scene from social network data. With hazards clearly labeled on the map, first responders can carry out their rescuer more effectively, while civilian can understand the situation better and avoid further casualties.

Challenges

Main challenges for this project are data collection, analytics and performance. For data collection, not only do we need to collect as much data from social networks and media in a reasonable rate, but we also need to verify and validate these data. After we have the data, what to do with it is another problem. Many previous projects only provided visualization of the social network data but did not include analytics on that data. Figuring out what analytics can help the rescuers and the general public the most is one of our top priorities. Performance when processing the data and running analytics is another concern as well, and the bottleneck is probably going to be the analytics module. The software requires inputs to be feed in and results to be outputted in real time, hence building data processing and analytics modules that are well optimized is another big challenge.

CHAPTER II

METHODS

Structure

This project currently uses an architecture similar to TweetTracker, which has a crawler-database-analyzer/visualization. The crawler mainly uses Twitter stream API to obtain live tweets and upload them to the database [3]. For limitation of the Twitter stream API [4], only a fraction of the real-time tweets is returned by the API, but this issue can be resolved by running multiple instances of the crawler and removing overlapped data during preprocessing.

For the database, we chose MongoDB, an open-source non-relational database, mainly for its non-relational nature that can store various kind of data without much effort in maintaining schemas. Users can send queries from the interface to fetch tweets related to the current disaster from the database and prediction of hazards from the analyzer, which will be visualized by the visualization tool on the interface.

We are using React to implement our website, which features a map with geolocated tweets being mark on it, a keyword search bar and a timeline slider that allows users to narrow down tweets displayed by date. We also plan to implement the connection between front-end and crawler, to allow our crawler start crawling in at areas not covered when users are looking at them.

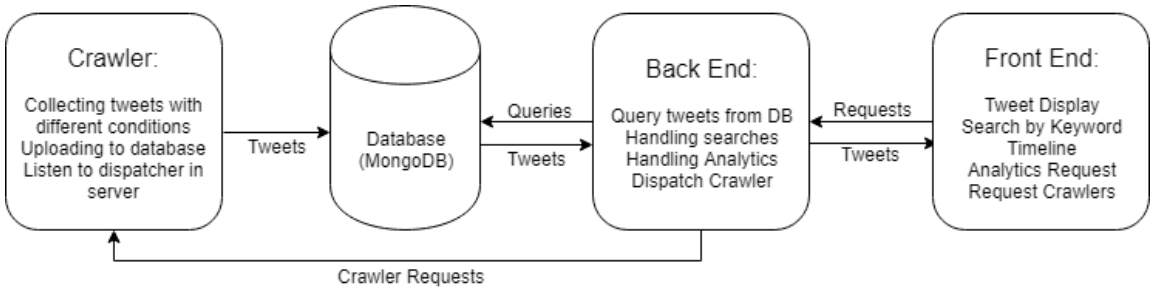

Figure 1. Software Structure

Analytics Module

Being able to give reasonable and interpretable conclusions on the data, beyond just marking the tweets on the map, is one of our aims of the project. In operation, the stream of tweets is feed by the server to the analyzer constantly, while the analyzer tries to detect locations of hazards from the tweet stream. Currently, we are aiming to implement a model that can detect trending topics within a group of tweets.

Supervised learning is one optional method. Learning from the comparison of performance of unsupervised and supervised learning methods in a research paper by CMU [5] and the fact that our project only focuses on disaster related topics, we can use supervised learning methods for our model. The supervised learning model can use long documents generated from aggregating many tweets as our training data, labeled with disaster related topics by methods described in the research paper mentioned earlier. However, due to the scale of the project, we want to try using unsupervised method first, as the researchers in the CMU research paper did.

For unsupervised learning, our approach to detect hazards in the disaster scene is clustering tweets together and conclude the trending topics from summaries of each cluster. For clustering algorithm, we attempted KMeans and later Gaussian Mixture Model(GMM). Before applying clustering, we first used truncated SVD to find and remove the principle components of the tweets, to filter out words that are uncorrelated. During clustering, in order to find the parameters that gives the best estimation, we used grid search to iterate through the number of clusters from 1 to 5 to find the best configuration. Scikit-Learn's grid search implementation also allows us to run it in parallel, reducing the runtime.

After the clusters are created, we also need to give summaries of each cluster for users to interpret easily. At first, we attempted to use LDA to extract the topic of each cluster, but we realized it is unpractically slow. We went for a more simplistic result and used NLTK's[6] pos_tag package to filter out nouns (with NN, NNS, NNPS or NNP pos tags) and verbs (with VB, VBP, VBG, VBD, VBN and VBZ pos tags). After filtering, we listed the most frequent nouns and verbs to give us a summary of the cluster.

We also plan to refine our training data by verifying it with the presence of spatial temporal data of hazards from previous disasters. With more detailed data during disasters obtained, like power outages and flooding data, better performance can be expected.

CHAPTER III

RESULTS

Overall Structure

The main framework of crawler-database-interface is set up, setting the foundation for future works. A multithreaded crawler is implemented with Python and Twitter stream API to simultaneously collect tweets with different keyword or in different locations and upload them to the database. The system also allows users to enter their Twitter Developer Token and start clusters of their own, reinforcing the crowdsourcing aspect of this project. The database is setup on MongoDB cluster, with the ability to scale storage and bandwidth based on usage. The website is built with Flask framework and React, capable of displaying tweets within the users view, requesting new tweets when users scrolls around or zoom out and display word frequencies of the tweets displayed. Currently, a detailed map with markers of geolocated tweets on it and list of all tweets displayed are available on the website. We are also working on displaying the result of trending topics by marking tweets in the same cluster with the same color and displaying a list of most frequent noun and verbs in each cluster. With each component developed separated, the system can be easily modified in the future.

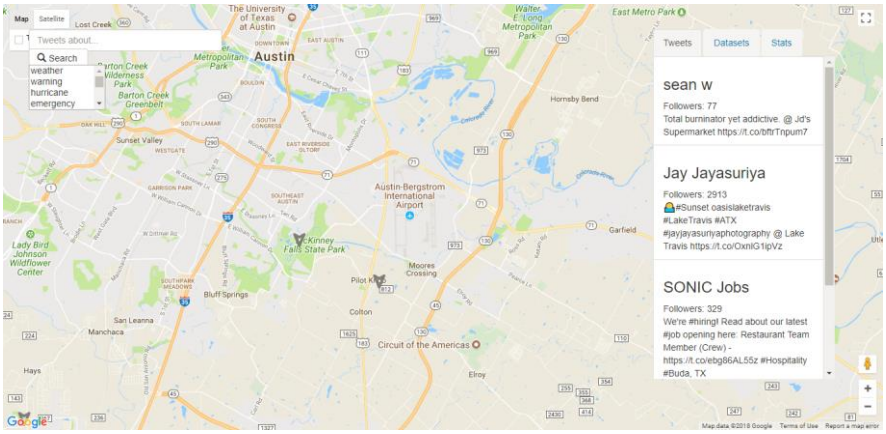

Figure 2. a screenshot of the current website

Analytics Module

We tested two clustering algorithms, KMeans and Gaussian Mixture Model(GMM), for performance in clustering tweets. Currently, both of these clustering algorithms give inconsistent results.

For a dataset collected in Houston during Hurricane Harvey and filtered with the keyword “barker”, the analytics module is able to come up with meaning topics that match real-life events on certain dates. Below are a few notable dates where we can identify distinct topics.

```
####################
2017-08-26 00:00:00 4
####################
Cluster 0 Number of tweets in each cluster:- 144
verbs [('spotted', 7), ('touched', 7), ('reported', 7), ('barker', 4), ('forming', 4), ('take', 4), ('confirmed', 3), ('hit', 3)]
nouns [('barker', 94), ('cypress', 77), ('tornado', 45), ('houston', 12), ('thabarkers', 11), ('harvey', 9), ('video', 8), ('area', 8)]
Cluster 1 Number of tweets in each cluster:- 1
verbs [('made', 1), ('absolved', 1)]
nouns [('atrocities', 1), ('barker', 1), ('carnival', 1), ('deal', 1)]
Cluster 2 Number of tweets in each cluster:- 1
verbs [('cavs', 1), ('acquired', 1)]
nouns [('yardbarker', 1), ('deal', 1), ('yardbark', 1), ('george', 1), ('report', 1)]
Cluster 3 Number of tweets in each cluster:- 25
verbs [('barker', 2), ('looking', 1), ('cypres', 1), ('raining', 1), ('amp', 1), ('khou', 1)]
nouns [('barker', 21), ('cypress', 12), ('tornado', 5), ('west', 3), ('khou', 3), ('bark', 2), ('video', 1), ('dam', 1)]
```

Figure 3. Summary of the cluster about the tornado spotted at Cypress on August 26th

```
####################
2017-08-27 00:00:00 4
####################
Cluster 0 Number of tweets in each cluster:- 138
verbs [('barker', 21), ('released', 15), ('res', 11), ('reservoir', 10), ('reservoirs', 8), ('addicks', 8), ('breaking', 6), ('know', 5)]
nouns [('barker', 100), ('addicks', 48), ('cypress', 33), ('water', 32), ('reservoir', 21), ('reservoirs', 18), ('release', 14), ('rain', 14)]
Cluster 1 Number of tweets in each cluster:- 2
verbs [('flooded', 1), ('complain', 1), ('see', 1)]
nouns [('park', 1), ('business', 1), ('marketing', 1), ('reservoir', 1), ('saintrph', 1), ('parts', 1), ('tho', 1), ('barker', 1)]
Cluster 2 Number of tweets in each cluster:- 17
verbs [('amp', 8), ('impacted', 1), ('evacuate', 1), ('flooded', 1), ('reservoir', 1), ('says', 1), ('spill', 1), ('happens', 1)]
nouns [('barker', 14), ('cypress', 6), ('amp', 6), ('addicks', 5), ('clay', 3), ('army', 3), ('corps', 3), ('water', 3)]
Cluster 3 Number of tweets in each cluster:- 2
verbs [('address', 2), ('reservoirs', 1), ('reservoir', 1)]
nouns [('banks', 2), ('levels', 1), ('harvey', 1), ('herzogweather', 1), ('issues', 1), ('barkers', 1), ('addicks', 1)]
####################
2017-08-28 00:00:00 4
####################
Cluster 0 Number of tweets in each cluster:- 636
verbs [('barker', 88), ('released', 87), ('reservoirs', 57), ('controlled', 45), ('addicks', 43), ('reservoir', 35), ('amp', 32), ('going', 25)]
nouns [('barker', 471), ('addicks', 301), ('water', 183), ('release', 149), ('reservoirs', 121), ('reservoir', 114), ('cypress', 90), ('houston', 59)]
Cluster 1 Number of tweets in each cluster:- 2
verbs [('get', 2), ('take', 1), ('needed', 1)]
nouns [('cypress', 2), ('barker', 2), ('road', 1), ('route', 1), ('galleria', 1), ('toll', 1), ('need', 1), ('amp', 1)]
Cluster 2 Number of tweets in each cluster:- 1
verbs [('affected', 1)]
nouns [('openings', 1), ('aches', 1), ('barker', 1), ('heart', 1), ('addicks', 1)]
Cluster 3 Number of tweets in each cluster:- 1
verbs [('largas', 1)]
nouns [('hay', 1), ('ramdalls', 1), ('lineas', 1), ('los', 1), ('por', 1), ('quedando', 1), ('esta', 1), ('vacia', 1)]
```

Figure 4. Summary of the cluster about the release of Addicks and Barker reservoir on August 27th.

While on a few datasets collected from southern Florida during Hurricane Irma, the analytics module would cluster most tweets into one cluster, which does not have a clear focus or topic.

CHAPTER IV

CONCLUSION

In this work, we provide a tool for rescuer and civilians during disasters, with a detailed representation of hazards from social network data. In the future we can improve this project in aspects below.

1. **More Social Network Sources**: The abundance of social network data like tweets enable us to recover the scene of disaster more precisely. In the future, inclusion of other social networks like Facebook, Nextdoor and addition of more categories of hazards can enable us the create a more detailed map. With the basic framework implemented, these features and hazards can be added conveniently.
2. **Media Sources:** Being able to provide media's coverage on the disaster can help users to understand the situation quickly. Currently, we only have Google News as our media source. In the future, we could work on more news sources, and even extracted metadata from these new articles to map them onto our map.
3. **More Analytics Options:** Currently, analytics module only provides basic searching, filtering by time and trending topic detection module. In the future, we would like to implement more analytics options
4. **Different Methods in Trending Topic:** As discussed in the optional methods above, we can try using supervised learning methods for trending topic detection, given we have enough labelled tweets.
5. **Optimization of Analytics Module:** Grid search is essentially running clustering algorithms with different parameters in different iterations, and hence can be improved greatly by running the iterations in parallel. Currently, multithreading in grid search only

provided minimal performance increase. In the future, we can use clustering methods and grid searches with more optimized implementations, for example PySpark, to increase the performance of the analytics module.

REFERENCES

[1] Grand Challenges – Restore and Improve Urban Infrastructure, http://www.engineeringchallenges.org/9136.aspx

[2] Kumar, Shamanth, et al. "TweetTracker: An Analysis Tool for Humanitarian and Disaster Relief." *ICWSM*. 2011.

[3] Lei Zou, Nina S.-N. Lam, Heng Cai, Yi Qiang, "Social Sensing of Disaster Resilience through Twitter: a Case Study of Hurricane Sandy"

[4] Streaming APIs – Twitter Developers, https://dev.twitter.com/streaming/overview

[5] Rosa, Kevin Dela, et al. "Topical clustering of tweets." *Proceedings of the ACM SIGIR: SWSM* (2011).

[6] Natural Language Toolkit – NLTK 3.2.5 documentation, https://www.nltk.org/