SAVITR- Web service utilising OSM for disaster preparedness and relief operations

Ritam Dutt $14CS30041$, Kaustubh Hiware $14CS30011$,
Avijit Ghosh $14CH3FP18$, Rameshwar Bhaskaran $14CS30027$, Nishant Nikhil $14MA20021$

***Abstract*—We present in this paper, an early warning system, called SAVITR. It leverages the vast information hosted on online social media, namely Twitter to forsee, monitor and analyse different disaster and emergency sitautions. It employs natural language processing techniques to infer the location of a tweet from the text in an unsupervised fashion and display it on a map-based interface.**

I. Introduction

The onset of online social media and microblogging sites like Facebook and Twitter have revolutionized lives in recent years. It has played a pivotal in the case of emergencies and disaster-related operations like the devastating Nepal Earthquake [1], Chennai floods [2] and Paris terrorist attacks[3]. The ubiquity of smartphones and the vast popularity of OSM have enabled people to act like social sensors and convey situational information quickly. Consequently, it is not only imperative to process the vast amount of incoming datastream on a real-time basis but also accurately extract relevant information from the unstructured and noisy data.

The work proposes a novel method of extracting locations from the text of English tweets in an unsupervised fashion, in contrast to using the geo-tagged field. The following reasons justify our choice. Firstly, tweets having a valid geo-tagged field are very sparse, especially in a developing country as India. 0.36%. Secondly, geo-tagged tweets might not be an accurate representation of the situation, as we demonstrate shortly.

We put forward the following research questions in this paper.

RQ1: What is the performance of the location detection algorithm in terms of coverage and accuracy?

RQ2: Is our method accurately able to conform with the real-world scenario? Simply put, is our method reliable?

II. Previous work

A few Information Systems have already been implemented in the USA and other countries for emergency informatics. The efficacy of such systems have been demonstrated in a variety of situations. Previous work on real-time earthquake detection in Japan was deployed by [1] using Twitter users as social sensors. Simple systems like the Chennai Flood Map [2] during 2015 Chennai floods have demonstrated the need and utility of Information Systems during emergency events in India. It used a combination of crowdsourcing, open source mapping technologies and contributed to large-scale civic participation.

Likewise, Ushahidi,[3] a non-profit crisis-mapping software company utilises the concept of crowdsourcing for social activism and public accountability. Here local observers submit reports using their mobile phones or the internet, while simultaneously creating a temporal and geospatial archive of an ongoing event. Ushahidi has been deployed in situations such as earthquakes in Haiti, Chile, forest fires in Italy and Russia.

The potential of crowdsourcing can be acknowledged from AIDR [4],a platform designed for classifying crisis-related microblogs (tweets). It has been developed by the Qatar Computing Research Institute in collaboration with United Nations Office for Coordination of Humanitarian Affairs (OCHA) and the UN International Childrens Fund (UNICEF).

Our system also works on the same basic principle as the aforementioned ones, information extraction from crowdsourced data. However, unlike [2] and [3], it is not necessary for the user to explicitly specify the location as a separate field. We intend to infer it from the tweet text, without any prior manual labelling. Consequently the unsupervised algrithm assures that our system is capable of integrating news from different sources. Moreover, unlike other systems which are used during crisis analysis, ours intend to detect the onset of some calamity, by analysing past trends.

III. Dataset

We have used the Twitter Streaming API [4], to collect tweets from 12^{th} September, 2017 to 13^{th} October, 2017, and filtered those containing only dengue or flood. This produced a massive dataset of 317567 tweets collected over a period of 31 days. The tweets were preprocessed to remove duplicate entries and also tweets written in different languages, specified by the ”lang” field in the tweet-json object. This resulted in 239269 unique tweets for both categories, floods and dengue. These were further segregated into tagged and untagged based on the presence of a valid location.

Our main aim was to collect and display tweets located only within India's bounding box. Thus, we needed some lexicon to disambiguate whether a place is located inside India and what are it's geographical coordinates. To that end, we scraped the data publicly available from Geonames, [5] and made a

[1] https://en.wikipedia.org/wiki/April_2015_Nepal_earthquake

[2] https://en.wikipedia.org/wiki/2015_South_Indian_floods

[3] https://en.wikipedia.org/wiki/November_2015_Paris_attacks

[4] https://developer.twitter.com/en/docs

[5] http://www.geonames.org/

dictionary corresponding to different locations within India. The dictionary has the information of 449973 places within India. However, some places mentioned in this dictionary have high othographic similarity with common English nouns. For example, the word song is a place located in Sikkim, whose coordinates are $27.24641'N, 88.50622'E$.

IV. Methodology

We now present the methodology employed to infer locations from the tweet as proposed below.

- The tweet texts are first preprocessed to remove mentions, urls, hashtags, RTs, unknown characters like $\&,\ldots$ and numbers are converted to a NUM tag. Case-folding and stop word removal is not carried out however, since we wish to retain the original text.
- We first observe the tweet's corresponding geo-tagged field, if any, and accept only those tweets whose location corresponds to a region inside of India. However the fraction of such tweets are very less, approximately 0.36% of the entire dataset.
- We then obtain the POS tags of the tweet text. T_i denotes the POS tag of the i^{th} word of the tweet. If T_i corresponds to a proper noun, we keep on appending words that succeed it, provided they are also proper nouns, delimiters or adjectives. If the word i is followed by a noun which corresponds to village, town, road, hospital, street, etc, we consider the i^{th} word as a viable location. We also consider the tag immediately preceeding the word T_i and if it is a preposition which usually preceedes a place or location, like at, in, from, to, near etc, we take that word into consideration as well. Thus we infer from the text proper nouns which conform to locations from their syntactic structure.
- We ran a named entity recognizer on the given text and observed those words whose tags correspond to LOC, GPE or FACILITY.
- We also considered hastags as a possible source of inferring a tweets's location.
- Combining the aforementioned techniques, we chose only those words which occur within India's bounding box.

. The named entity analysis, and POS tag distribution was carried out using SpaCy [6] as opposed to the CMU TweeboParser[7], due to the heavy processing time of the latter. The TweeboParser was 1000 times slower as opposed to SpaCy. We considered the speed to be a viable trade-off for accuracy since our aim is to deploy a system which would work on a real-time basis and we observed the processing time would be a bottleneck in this regard.

V. Evaluation

A. RQ1

What is the performance of the proposed algorithm in terms of accuracy and coverage?

We define accuracy in terms of F1 score and coverage as the increase in the fraction of tagged tweets.

From the entire set of 239269 tweets, only 3493 were tagged, out of which 869 were from India. This corresponds to a minute proportion of approximately 0.36% of the entire dataset. On the other hand the number of tweets which were successfully tagged using our algorithm was approximately 11.05%. This increases the coverage drastically and was able to take into account niche and remote places like Ghatkopar, Pipra village and Kharagpur, besides metropolitan cities like Delhi, Kolkata and Mumbai.

TABLE I
Coverage of tweets

Stats	Dengue	Flood
Tagged	14044	13254
Untagged	65853	147785
%-wise tag	17.577	8.23

The relative proportion of tagged tweets for dengue is higher than that of floods as observed in Table I. This discrepancy arises since dengue is a niche and more specialised topic as opposed to the other. Consequently while streaming for floods, we obtained irrelevant tweets of this format [8]

We have shown a distribution of tagged tweets for the two situations across the timeframe of 31 days.

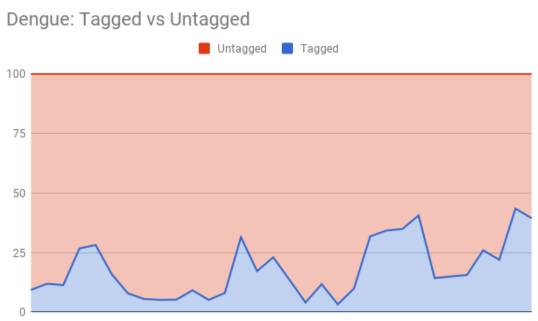


Fig. 1. Proportion of tagged dengue tweets

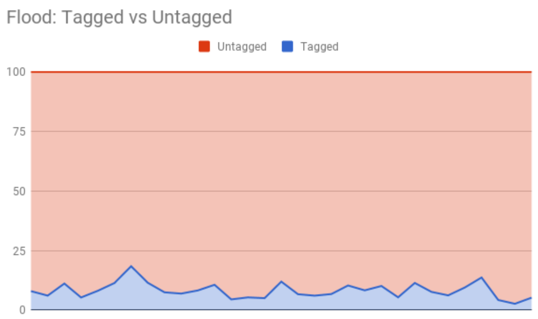


Fig. 2. Proportion of tagged flood tweets

[6] https://spacy.io/
[7] http://www.cs.cmu.edu/ ark/TweetNLP/

[8] RT @IndianMillenni1: @republic .@iHrithik has proved that hes a real life hero. I hope that love and support for Hrithik floods in afte

The important or the more frequently occuring locations are captured in the following word cloud However we demonstrate that the algorithm was not only able to capture well-known, popular metropolitan locations only but also some remote places inside India, via the Table II As observed from the frequency distribution plot, it is evident several rural and remote places occur within India have been identified by the algorithm. The ones in the Table II under the Infrequent category all have a frequency of 1.

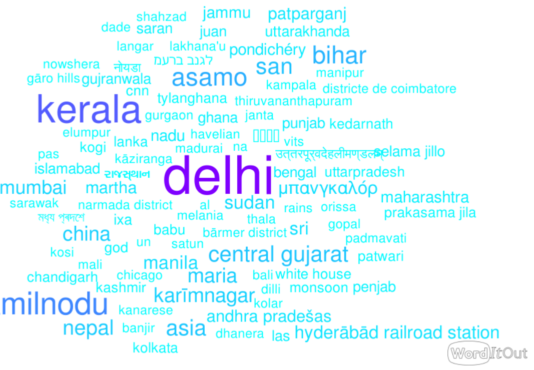


Fig. 3. Most frequently occuring cities

TABLE II
COVERAGE OF TWEETS

Type	Location name
Frequent	delhi, kerala, tamilnodu, mumbai assamo, bihar,central gujrat,
Infrequent	indirapuram, ramnagar, hosur, bandra, tripura,ghaziabad, yamunanagara jila, kharagpur

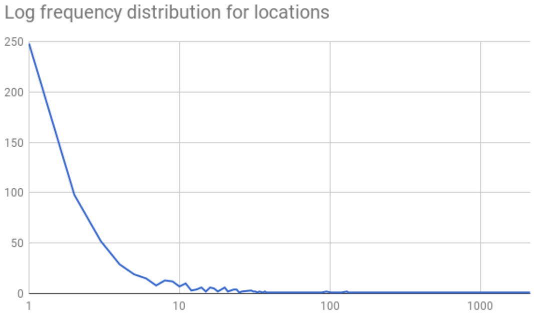


Fig. 4. Frequency distribution of the freuency of locations

It is important to note that there are some invalid locations as well such as potus, lol, god in the dictionary obtained from Geonames.

We needed to perform an evauation to see the correctness of our algorithm. As mentioned before, we aim to measure the accuracy of the system in terms of precision, recall and F1 score. To that end, we have randomly sampled 200 tagged tweets and another 200 untagged ones. We evaluate the precision of the system by observing the tagged tweets and assigning it's relevance if a location within India was mentioned in the tweet. Likewise if it mentions a location not within India, the tweet is deemed invalid. In case a tweet has more than one locations, any one of them is a valid mention. Likewise the recall of the system was observed from the untagged tweets. The table describes the results in this aspect.

TABLE III
EVALUATION METRICS

		Actual	
		Tag	No Tag
Predicted	Tag	152	48
	NoTag	13	187

The precision measured is 0.76, recall is 0.9212 and F1 score is 0.8475

B. *RQ2*

Is our method accurately able to forsee real-world disaster scenarios?

In order to analyse the reliability of the system we consider the massive dengue outbreak that plagued India in the fall of 2017. [9]. The report, published on 2^{nd} November mentions the plight of the Southern States in the wake of dengue, some of the affected ones being Kerala and Tamil Nadu. We highlight the findings from our data in this aspect.

The number of unique tweets mentioning Kerala and Tamil Nadu were very high. The unnatural reference to these names as opposed to other metropolitan areas like Bangalore or Kolkata is proof that people are referring to these places more. This is evident as we observe the following statistics about Kerala. There were 2204 tweets about Kerala out of which 1960 tweets contain the term dengue. This amounts to a 88.89% overlap. However most of the data is inferred from the tweets having Kerala mentioned in their texts. We see some examples of geo-tagged tweets below which talk about Kerala but are posted from other parts of India. This paints an inaccurate description of the situaion.

TABLE IV
COVERAGE OF TWEETS

Tweet Text	Location
Numerous death in Kerala from Dengue, Chicken guinea, Malaria @cpimspeak pushed Kerala into a money order econom...	. New Delhi
@Bhayankur Hmmm - not rosy in Kerala either ...	New Delhi
Dengue : 5 worst affected states. Scandinavian level HDI state Kerala tops the list ...	Bengaluru
45% of Dengue cases and nearly half the Dengue related deaths in India from Kerala. Too much filth or related to we..	Bengaluru
@Rameshnair101 @CNNnews18 Dengue cases reported: UP 302, Kerala 16530 .death due to dengue: UP 17, Kerala 28 mortal..	Kerala

VI. SYSTEM DESCRIPTION

We started off by developing upon TweetGeoViz, which was developed in react and nodejs. However, due to its

[9] https://www.telegraphindia.com/india/dengue-spurt-in-south-182846

bulky nature, the system did not display any results when the database contained over 3 lakh tweets.

Owing to ease of control, we decided to port the complete system to Python. The current system, live on savitr.herokuapp.com, works well at the same number of tweets. It has been built using flask and dash libraries.

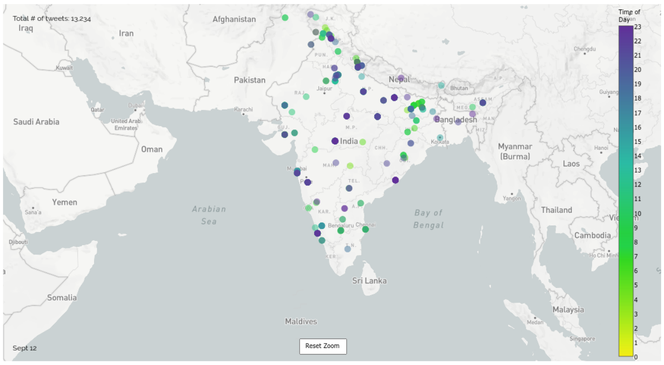
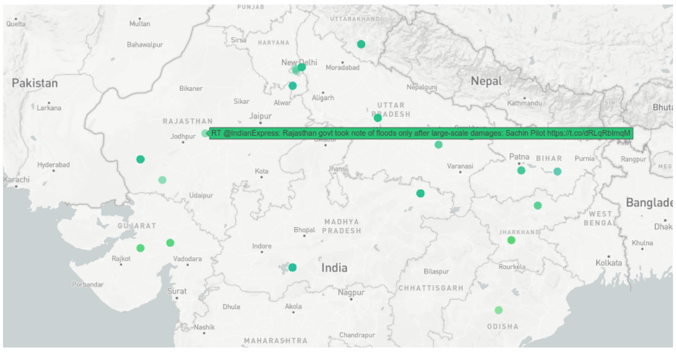


Fig. 5. Tweets visualized on Indias map



RT @IndianExpress: Rajasthan govt took note of floods only after large-scale damages: Sachin Pilot https://t.co/dRLqRbImqM

Fig. 6. Hovering over points reveal tweet text

The current system allows the user to search for tweets pertaining certain situations, like floods, dengue, etc. Hovering over a point reveals the tweet text. The possible situations were enlisted beforehand, but it will be possible to search for tweets containing any word.

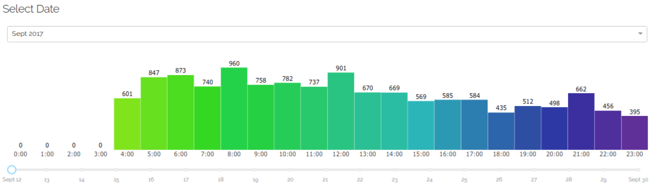


Fig. 7. Tweets sorted by time of posting

Untagged tweets are shown separately in tabular format at the very end. At any given instance, the user can see untagged tweets on a certain day. To reduce server strain, the user can only view data spanning 30 days, not more.

Further extensions to be added:

- Currently, the data is being displayed from a static file. We will soon be shifting to a database support.

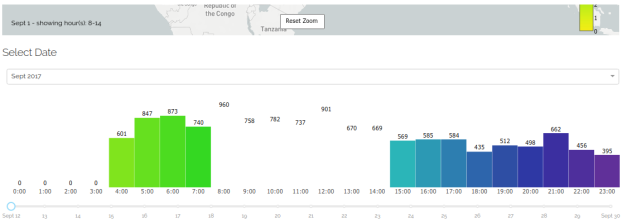


Fig. 8. Crosshair selection of time interval to visualise

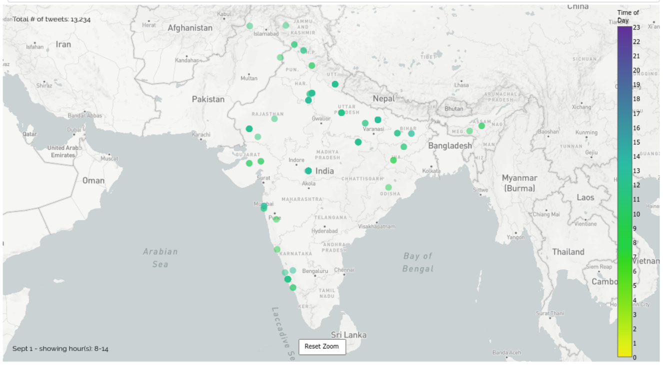
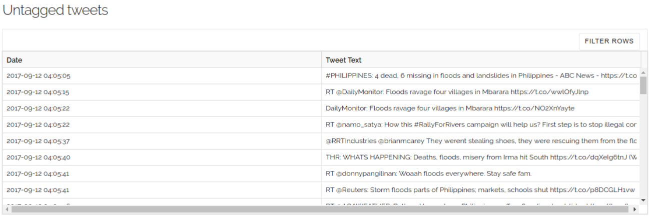


Fig. 9. Tweets for certain time window displayed



Date	Tweet Text
2017-09-12 04:05:05	#PHILIPPINES: 4 dead, 6 missing in floods and landslides in Philippines - ABC News - https://t.co
2017-09-12 04:05:15	RT @DailyMonitor: Floods ravage four villages in Mbarara https://t.co/wwlOfyJlnp
2017-09-12 04:05:22	DailyMonitor: Floods ravage four villages in Mbarara https://t.co/NO2XnYayte
2017-09-12 04:05:22	RT @namo_satya: How this #RallyForRivers campaign will help us? First step is to stop illegal cor
2017-09-12 04:05:37	@RRTIndustries @brianmcarey They werent stealing shoes, they were rescuing them from the flc
2017-09-12 04:05:40	THR: WHATS HAPPENING: Deaths, floods, misery from Irma hit South https://t.co/dqXelg6tnJ (W
2017-09-12 04:05:41	RT @donnypangilinan: Woaah floods everywhere. Stay safe fam.
2017-09-12 04:05:41	RT @Reuters: Storm floods parts of Philippines; markets, schools shut https://t.co/p8DCGLH1vw

Fig. 10. Untagged tweets displayed in tabular form at the bottom

- The system will soon be able to visualise tweets spanning multiple days. Currently, the longest duration of tweets visualised in 1 day.

VII. Conclusion

We have developed an unsupervised algorithm of extracting location information from tweets with a high F1-score of 84.05%. Moreover the obseravtion mirrored real-life occurences, like the massive dengue outbreak in South India, 2017, which were accurately inferred from the text rather than the geo-tagged location. We have also developed a system that allows for the visual representation and analysis of the potential disaster-scenarios.

References

[1] Sakaki, Takeshi, Okazaki. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. *Proceedings of the 19th International Conference on World Wide Web WWW*, pp. 851-860, 2010.

[2] A. Ganesh, S. Bhangar, and S. Aruna. https://osm-in.github.io/flood-map/chennai.html#11/13.0000/80.2000

[3] https://www.ushahidi.com/

[4] http://aidr.qcri.org/