Detecting Rumors in Disaster Related Tweets

Claire Cateland, Kevin Luu, Kirsty Hawke,
Richard Stassen, Saud Nasri, Sean Atkinson

Contents

The dangers of rumors spread by social media in a time of crisis…

The Boston Marathon Bombings

Image courtesy of KPCC

Sunil Tripathi

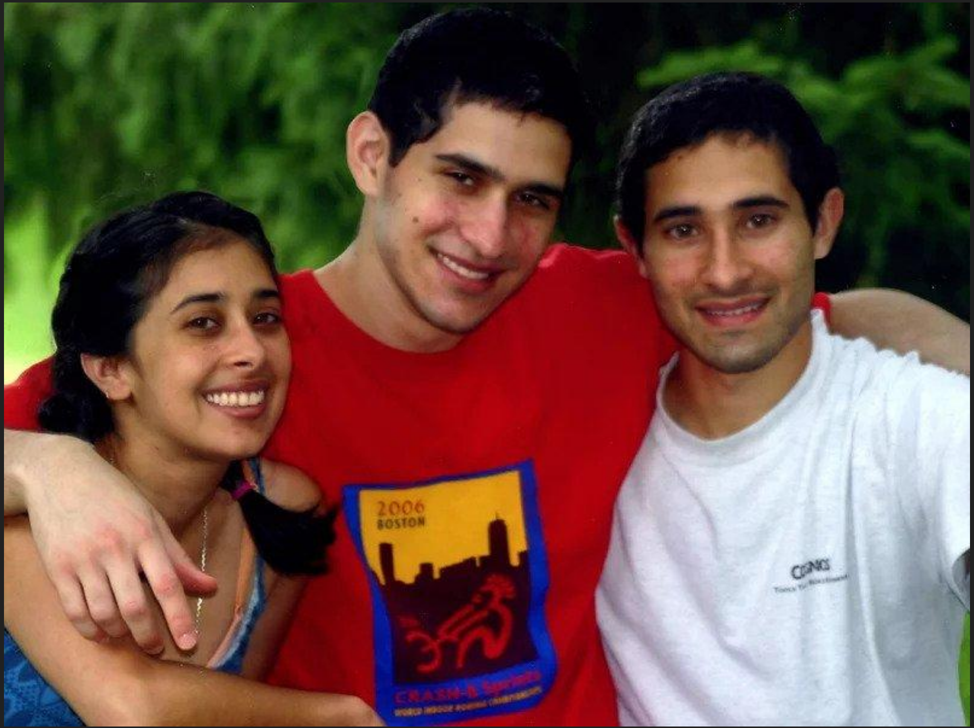


Image courtesy of NPR

Image courtesy of NBC News

Unfortunately, Sunil's story isn't the only example of the danger of rumors in a time of crisis or disaster:

- During Hurricane Harvey, there were rumors that undocumented immigrants could not go to shelters because they would be reported to ICE

- During Hurricane Ida, there were rumors that people needed to show proof of Covid-19 vaccination to stay in shelters

- Also during Hurricane Ida, there was a rumor that the Louisiana Department of Disaster Assistance had designed a program to provide anyone in need with $8,500

All the above rumors proved to be false.

Rumors also take resources away from relief efforts

“Conspiracy theories and misinformation take valuable resources away local fire and police agencies working around the clock to bring these fires under control. Please help our entire community by only sharing validated information from official sources.”

- Federal Bureau of Investigation

Problem Statement

Can we produce a machine learning model to identify whether tweets are related to disaster events and to assess their credibility?

Background

- **Hunt, Agarwal & Zhuang (2020)**
  Monitoring Misinformation on Twitter During Crisis Events: A Machine Learning Approach

- **Buntain & Golbeck (2017)**
  Automatically Identifying Fake News in Popular Twitter Threads

A Difficult Problem…

- Limited data

- Breaking problem into two manageable components
  - Disaster identification
  - Veracity assessment

- Hypothesis: the model will generalize well enough to be useful for evaluating the veracity of disaster tweets

Data Preparation

Kaggle - Disaster Relevance:

10,860 tweets

CREDBANK process:

169 million raw tweets → 62,000 topics → 1,378 events → 80 million scored tweets

Data Preparation

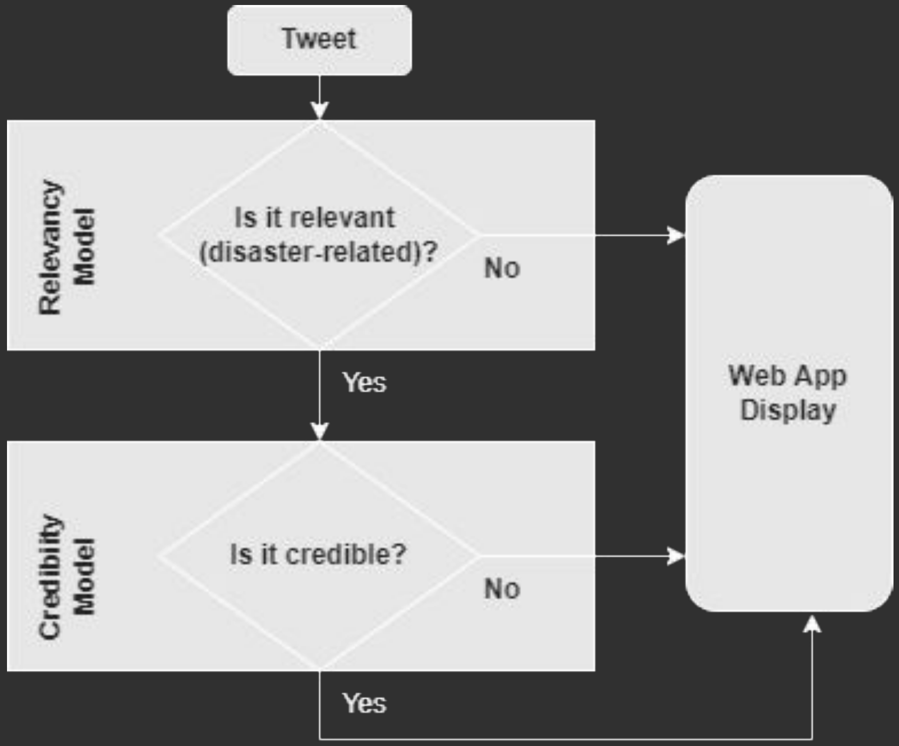
Our Process:

- Each event scored by 30 different evaluators

- To classify our training data we follow the same process as Buntain & Golbeck (2017)

- Take the average score for each event, and select only events with scores in the top and bottom decile

Methodology



Models Considered - Relevance

Convolutional Neural Network

Logistic regression

Naive Bayes

Random Forest

Decision Tree

Models Considered - Credibility

- Random Forest Classifier
- Bagging Classifier
- AdaBoost Classifier
- K Nearest Neighbors Classifier
- Decision Tree Classifier
- Support Vector Classifier

Support Vector Classifier

- Applies a linear kernel function to perform classification
- <u>Advantage:</u> low bias and low variance without much tuning
- <u>Disadvantage:</u> very slow to train, loss of interpretability

Bagging Classifier

- Fits base classifiers each on random subsets then aggregates their predictions
- <u>Advantage:</u> relatively fast training
- <u>Disadvantage:</u> higher variance, loss of interpretability

Modeling - Neural Net

- Embedding input + 2 Conv1D + 1 Dense + Dense sigmoid output
    - Google News Word2Vec embedding weight
    - All layers (except output) — Batchnomalization & Dropout
    - Conv1D layers — AveragePooling1D & GlobalAveragePooling1D
- Stochastic Gradient Descent (SGD) optimizer

Neural Net - Initial Scores

Data	Loss	Accuracy	Validation Loss	Validation Accuracy
Relevance	0.3918	0.8354	0.4055	0.8310
Credibility	0.0431	0.9831	0.0436	0.9827

Relevance Model Results

Model	Training Accuracy	Validation Accuracy
Logistic Regression	0.883	0.795
Naive Bayes	0.901	0.801
Decision Tree	0.981	0.728
Random Forest	0.988	0.797
Convolutional Neural Network	**0.835**	**0.831**

Credibility Model Results

Model	**Training Accuracy**	**Validation Accuracy**
Multinomial Naive Bayes	0.954	0.938
Random Forest	0.996	0.978
Bagging	0.995	0.973
ADABoost	0.954	0.952
KNN	0.881	0.862
Decision tree	0.997	0.969
SVC	**0.992**	**0.983**
Convolutional Neural Network	**0.983**	**0.982**

Web App

Streamlit

- Host: Streamlit
- User input : tweet (string)
- Two layer results:
  - Disaster model confidence based on relevancy
  - Credibility model returns likelihood of being truthful or not

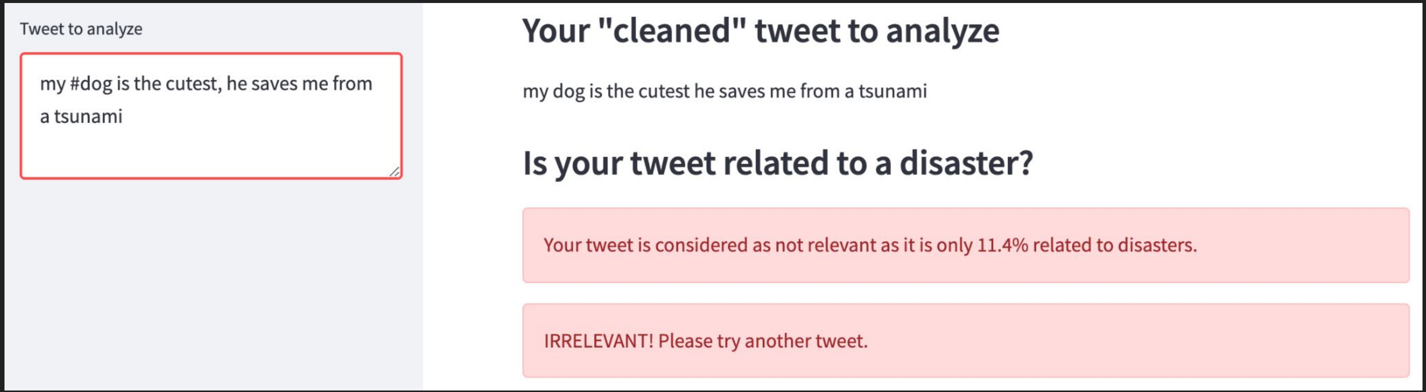
Web App Considerations

- Visual highlights for the user to quickly identify if the result is relevant/accurate or not
- Using SVC model for available space and computer resources
  - CNN model too large
- Calling multiple part of the original code (cleaning, tokens and model)

Modeling - Web App Implementation

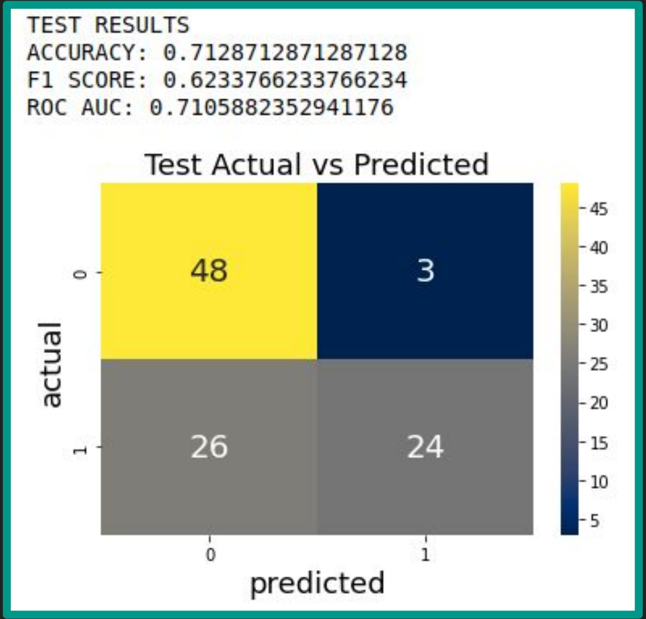
- Using Streamlit to host our app
- User input : testing our model with a tweet of its choice
- Two layer results:
  - Disaster model returns how confident it is that the tweet is related to a natural disaster
  - Accuracy model return how confident it is that the tweet is not a rumor
- Adding visual highlights for the user to quickly identify if the result is relevant/accurate or not
- Using SVC model for available space and computer resources
  - CNN tried also
- Calling multiple part of the original code (cleaning, tokens and model)

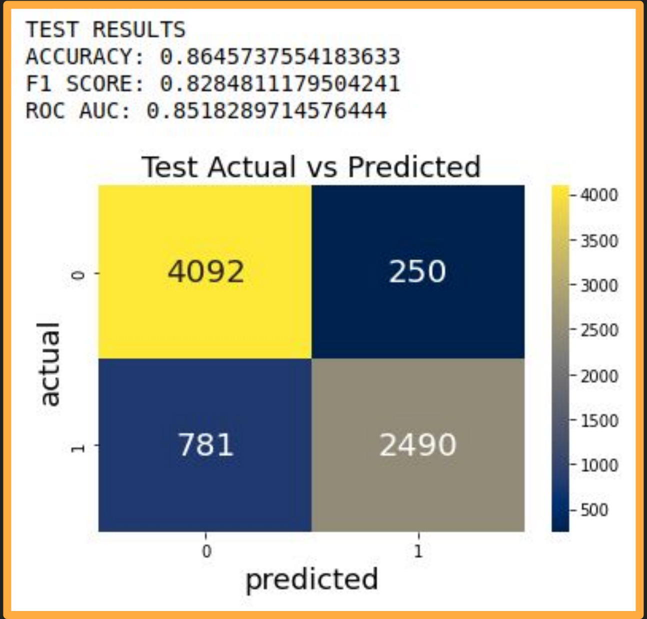
Examples/Screenshots



CNN Testing Results - Relevance

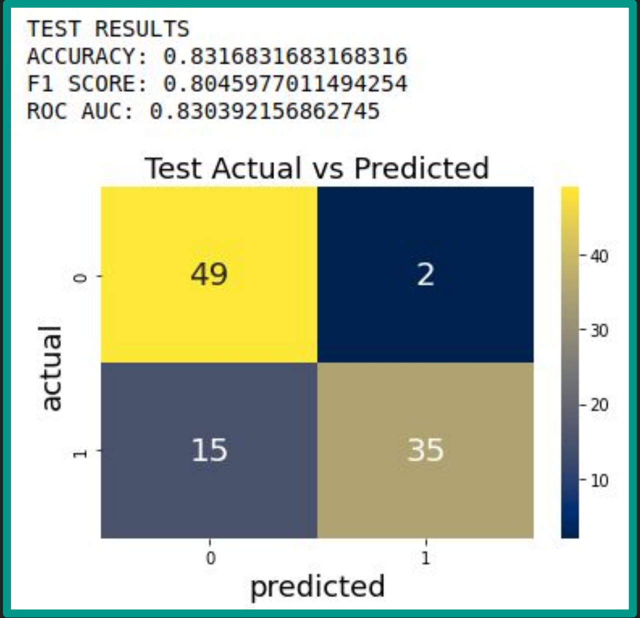
- Testing results on hand-selected subset and Kaggle competition dataset:

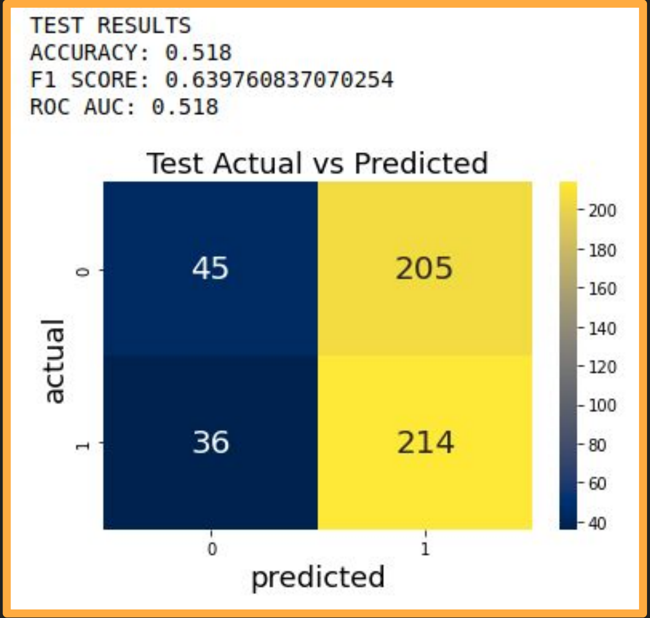




SVC Testing Results - Credibility

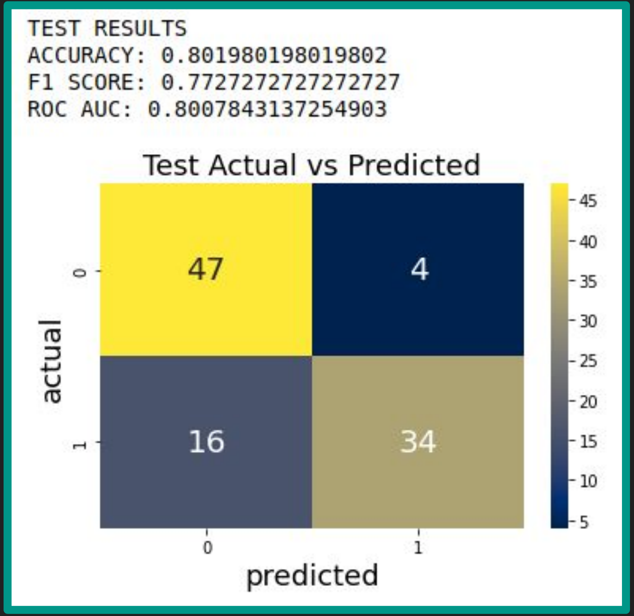
- Testing results on hand-selected subset and wildfire dataset:

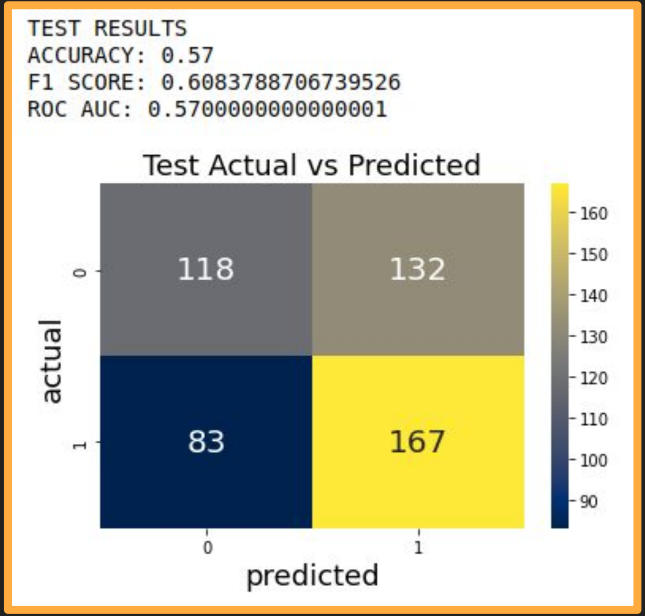




CNN Testing Results - Credibility

- Mirrored structure to the relevance model, unused due to web-app limitations
- Testing results on hand-selected subset and wildfire dataset:





Future Improvements

- Improve upon existing datasets
  - Add features (followers, thread depth, account age)
  - Extract more tweets
- Select a web app service that can host the neural net
- Create a better dataset
  - More reliable labelling
  - Specific disaster relevance
- Explore transformers

Thank you!

References

C. Buntain and J. Golbeck, "Automatically Identifying Fake News in Popular Twitter Threads," 2017 IEEE International Conference on Smart Cloud (SmartCloud), 2017, pp. 208-215, doi: 10.1109/SmartCloud.2017.40.

Han S., Gao, J., Ciravegna, F. (2019). "Neural Language Model Based Training Data Augmentation for Weakly Supervised Early Rumor Detection", The 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2019), Vancouver, Canada, 27-30 August, 2019

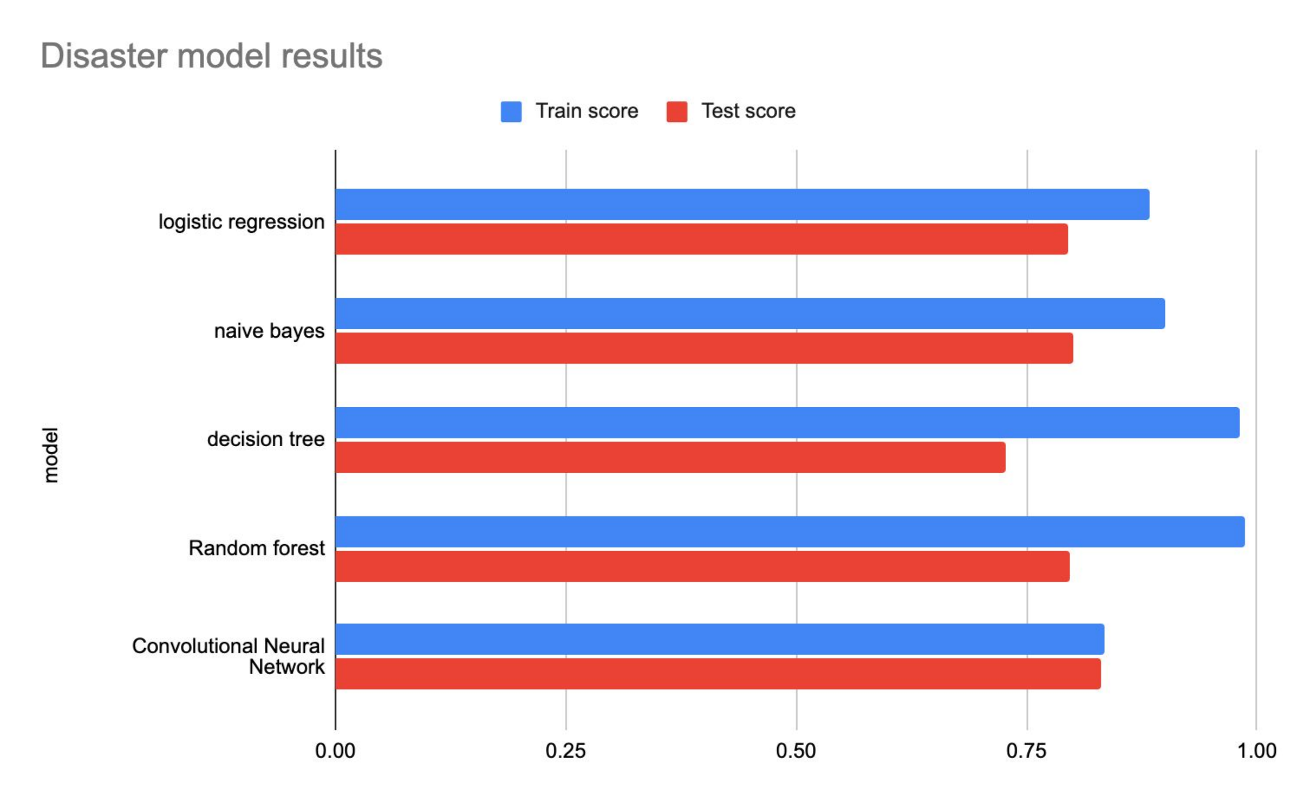
Mitra, T., & Gilbert, E. (2021). CREDBANK: A Large-Scale Social Media Corpus With Associated Credibility Annotations. Proceedings of the International AAAI Conference on Web and Social Media, 9(1), 258-267. Retrieved from https://ojs.aaai.org/index.php/ICWSM/article/view/14625

Murayama, T., Wakamiya, S., Aramaki, E., & Kobayashi, R. (2021). Modeling the spread of fake news on Twitter. PLOS ONE, 16(4). https://doi.org/10.1371/journal.pone.0250419

Sharma, K., Qian, F., Jiang, H., Ruchansky, N., Zhang, M., & Liu, Y. (2019). Combating fake news. ACM Transactions on Intelligent Systems and Technology, 10(3), 1–42. https://doi.org/10.1145/3305260

Shu K, Mahudeswaran D, Wang S, Lee D, Liu H (2020) FakeNewsNet: a data repository with news content, social context, and spatiotemporal information for studying fake news on social media. Big Data 8:3, 171–188, DOI: 10.1089/big.2020.0062.

Relevance Model Results



Credibility Model Results