Detecting Rumors in Disaster Related Tweets

Authors:

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

Table of contents

Introduction

Social media has proven to be a powerful enabler of human connection but it has also demonstrated serious pitfalls. Like any technology, it is neither inherently good nor evil, but rather its final impact depends on how it is used and regulated. There is an easy analogy to nuclear power, which on the one hand could be harnessed to satisfy the world’s energy needs, while on the other it can be used to create devastating weapons that threaten our very existence. It may seem a dramatic way to frame it, but in recent years the proliferation of social media has presented similar stakes. At its best, social media empowers us. It allows rapid dissemination of information that can facilitate bottom-up social coordination and promote democratic action. At worst, the very same mechanisms spread rumors and misinformation that erode standards of truth and leave us polarized and disconnected.

While social media does present risks and challenges, its crowd-sourced data stream has many potential applications, one being as a source of up-to-the-minute information during disaster events. We use the term ‘disaster’ in a broad sense, referring to an acute event that poses a serious threat to human lives and livelihoods. This includes natural disasters such as hurricanes and earthquakes but also includes human-caused incidents such as wildfires, plane crashes, or mass shootings. In all such cases, there can be a scarcity of information as the events unfold in real-time, producing uncertainty for emergency respondents and for those affected. Live information from platforms like Twitter could therefore help inform disaster response but, given the volume of data, it is difficult to verify tweets to ensure that people are acting on valid information.

Machine learning (ML) – another technology with its own set of benefits and pitfalls – may be useful for processing the raw tweet stream to determine which tweets might contain information that is accurate and pertinent to disaster events. The purpose of this report is to evaluate the potential for ML towards this application. In particular, we evaluate the following **Research Question**:

Can we produce an ML model to accurately assess tweets’ relevance to disaster events, as well as their credibility?

This is an inherently difficult problem to solve due to a dearth of labeled data that can be used for model training – while fake news identification has been well-studied and is of particular interest to social media platforms themselves, there are few available datasets specifically pertaining to rumors about disasters. As a result, we divide this analysis into two separate components. First, we build a disaster relevance model that can assess a raw tweet and classify it as disaster-relevant or disaster-irrelevant. Second, we tackle the question of rumor identification, training a rumor classification model on general misinformation datasets, not particularly about disasters. Our hypothesis is that this model will be generalized well enough to be useful in evaluating the veracity of disaster tweets.

Background

Many studies have attempted to use machine learning to identify fake news and misinformation but very few have considered this question in the context of disaster response. One study from Hunt, Agarwal & Zhuang (2020) attempted to answer a research question very similar to our own but with mixed results. They analyzed tweets related to six crisis events that were known to have rumors associated with them and used an ML framework to classify tweets as rumor or not-rumor. Their models performed well when trained using data that contained items from the same events that were present in testing data, but their results did not generalize well to unseen data from new events. In one component of their analysis, they trained models using tweets from five events and testing models on tweets relating to the sixth event, achieving an F1-score of 50.6%. They then repeated this process, gradually including tweets from the sixth event into their training dataset, and eventually reached an F1-score of 73.4% using an SVM model when 40% of sixth-event tweets were included. These results demonstrate the difficulty of creating a general tool for rumor detection, suggesting that, to evaluate veracity of tweets from a new, unseen event, examples from that event need to be included in training data.

This question of generalizability is examined by Buntain & Golbeck (2017), who trained models using a large crowd-labeled dataset called CREDBANK (more details on this are provided in our data section) and evaluated predictions on the entirely independent and professionally fact-checked BuzzFeed News Fact-Checking Dataset. Their analysis yielded a ROC-AUC of 73.80% and an accuracy of 65.29% when testing the CREDBANK-trained model against the BuzzFeed testing model, which is similar to the model's performance against unseen CREDBANK testing data.

This would suggest that generalizability is possible, however, Buntain & Golbeck trained their model using features that may not be available for the real-time evaluation of a raw tweet stream during a crisis event. The authors divide their features into four types: (1) Structural Features, (2) User Features, (3) Content Features, and (4) Temporal Features. Features of types (2) and (3) would be available in real-time as they relate to the text-content of a tweet, as well as characteristics of the tweet's author, such as account age, the number of followers, and the difference between when an account was created and when the relevant tweet was posted, which would help weed out spambots (Temporal Features inform about how some of these change over time, e.g. trend in a user's follower count). Features of type (1), however, would not be available in real-time. These include variables such as thread length and average tweet length per thread. They carry information on structured groups of tweets, which could still be active and evolving during a live event. In order to address this, our analysis focuses only on tweet content (text data), and attempts to produce a general model trained on CREDBANK data that can accurately classify unseen disaster tweets.

Data

Disaster Relevance

The following table contains the data used for production and testing of the disaster relevance model:

Name	Description	Use
Kaggle - Disaster Relevance	Collection of tweets, hand classified as relevant or irrelevant to disasters of numerous types	Model Training
Google News Word2Vec	Pre-trained Google News corpus (3 billion running words) word vector model	Embedding Weights
Kaggle - NLP Competition	Collection of tweets, hand classified as relevant or irrelevant to disasters of numerous types	Model Testing

Credibility Assessment

The following table contains the data used for production and testing of the disaster relevance model:

Name	Description	Use
CREDBANK	Streaming tweets collected between mid Oct 2014 and end of Feb 2015: topics are classified as events or non events, events annotated with credibility ratings	Model Training
Google News Word2Vec	Pre-trained Google News corpus (3 billion running words) word vector model	Embedding Weights
California Wildfire Data	Collection of 500 Tweets about California, divided into 4 groups: 25% are rumors related to wildfires; 25% are non-rumors related to wildfire; 25% are rumors unrelated to wildfires; 25% are non-rumors unrelated to wildfires. Data were collected and labeled by Sean Atkinson.	Model Testing
PHEME Augmented Data	Collection of Twitter rumours and non-rumours during six real-world events	Model Training (Not used in final analysis)

LIAR	Publicly available dataset for fake news detection that can be used for fact-checking research. The dataset includes 12.8K human labeled short statements from POLITIFACT.COM's API	Model Testing and Training (Not used in final analysis)

The CREDBANK dataset is a large collection of tweets relating to events that occurred between mid October 2014 and end of February 2015, which were assessed for veracity by crowdworkers through Mechanical Turk (Turkers). The data are publicly available here, along with a detailed description of the data.

The dataset was developed through 4 steps, each with a corresponding CSV file, available for download:

1. ***Streaming Tweet File - stream_tweets_byTimestamp.data:***
   169 million tweets collected from within the aforementioned time interval.
2. ***Topic File - eventNonEvent_annotations.data:***
   62,000 tweet topics, which were generated from the above raw tweet stream using automated topic modeling (LDA). Each topic is characterized by a list of 3 topic terms. Each topic is also rated by Turkers as being an event (class 1) or non-event (class 0).
3. ***Credibility Annotation File - cred_event_TurkRatings.data:***
   1,378 tweet topics that were categorized as events through the process described in #2, above. Each topic was evaluated by 10 Turkers, and was counted as an event if a majority (6/10) assigned it a value of 1. Each of these event topics were then evaluated for credibility by 30 Turkers, each scoring it on a scale ranging from -2 (least credible) to +2 (most credible).
4. ***Searched Tweet File - cred_event_SearchTweets.data:***
   80 million tweets grouped by the 1,378 event topics from #3, above. Tweets corresponding to each event topic were extracted using the 3 topic terms to form an 'AND' query.

For our analysis we use only files #3 and #4, to produce a dataset of roughly 13 million individual tweets, that are grouped into event topics and rated for credibility by 30 Turkers. That is, each tweet is given the credibility score corresponding to its event topic. We roughly follow the data preparation process described by Buntain & Golbeck (2017), whereby each tweet is given an aggregate credibility score that is the mean of 30 individual Turker scores. We then use only the top and bottom deciles of scores, to help ensure that the event topics we include have greater Turker consensus, and are therefore more likely to truly belong to the negative or positive class. We also take further measures to select a sample of these data for modeling as we faced time and computational constraints. The details of this process are described in parts one and two of the associated technical document.

Methodology

Neural Network

Through trial and error of numerous neural network implementations, a low-loss, low-variance model was developed with an accuracy of 83% for disaster relevance classification:

- Training results: loss: 0.3918 - acc: 0.8354 - val_loss: 0.4055 - val_acc: 0.8310
- Model composition:
  - Stochastic Gradient Descent (SGD) optimizer
    - Adam resulted in very overfit models with high variance
  - Embedding input layer
  - 2 Conv1D layers (32 & 64 filters respectively)
  - 1 Dense layer (32 filters)
  - Dense sigmoid output layer
    - Batchnomalization + Dropout applied to all layers except output
    - AveragePooling1D and GlobalAveragePooling1D (similar to AveragePooling+Flatten) applied to all Conv1D layers

A CNN + Bi-LSTM model yielded slightly lower variance and loss, however was not selected due to the significant decrease in memory and computational efficiency. The trade-off in efficiency for very minimal improvements was unreasonable.
Although a similar model with 98% accuracy was developed for credibility classification, it was not utilized due to its excess size causing complications with the web application integration.

Support Vector Classifier

Support Vector Classifiers (SVC) is useful for a supervised learning method for classification problems. It is memory efficient and works well in high dimensional spaces[1]. From the preliminary investigation of model performances, it outperformed the following models:

- Random Forest Classifier
- Bagging Classifier
- Ada Boost Classifier
- K Nearest Neighbors Classifier
- Decision Tree Classifier

After SVC was chosen, the final model was a result of hyperparameter tuning. A bagging model was also developed to compare the training cost and trade off with variance, however, in the end the SVC model worked the best. The specifics of the model performances can be read in the Results section.

[1] https://scikit-learn.org/stable/modules/svm.html

Web App

We chose to use Streamlit to host our webapp for its design and the ease of organizing its display:

- The user input will be tested against our model - it should be a tweet either related or not to a disaster
- Our two layer model is applied to this tweet:
  - First, our disaster model returns how confident it is that the tweet is related to a disaster
  - Second, the credibility model returns if the tweet is likely or not to be credible

We chose to use the SVC model based on available space and computing resources. In order to be as efficient as possible, our Streamlit code is calling multiple parts of the original code (cleaning, tokens and model) as different files. In a user-friendly perspective, we added visual highlights for the user to quickly identify if the result is relevant/accurate or not using green, orange and red colors.

Results

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
MultiNB	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

Conclusion

The results of our analysis are largely consistent with Hunt, Agarwal & Zhuang (2020). Each of our ML models achieves a high degree of accuracy when classifying disaster-related tweets as rumors or non-rumours, but only when evaluating tweets relating to events that are represented in training data. When we attempt to generalize our model, applying it to tweets relating to an unseen event, it shows poor performance with an accuracy similar to our baseline.

There are several reasons why this may be the case, not the least of which is data quality. Even simple image labeling can be expensive and time-consuming, but credibility data needs to be labeled as true or false, which can require fact-checking and sometimes domain knowledge. Humans are also subject to biases and interpretation, which can result in incorrect or imprecise data labels. This latter concern is certainly an issue for the CREDBANK data, having been labeled by crowdworkers on Mechanical Turk. Upon scoring an event for credibility, each evaluator also provided a short note explaining their reasoning. Inspection of a sample of low-credibility topics shows that in many cases the low score is not due to its being a rumor or misinformation, but rather that it wasn’t precisely an event and so the scorers were unsure about how to classify it as having actually occurred or not. For example, one of the negative-class topics in the CREDBANK dataset was identified with the keywords “merry”, “everyone”, and “guys”, and contains tweets about Christmas well-wishing. This topic received a low credibility score because many of the reviewers didn’t consider the holiday to be an event and so rated the topic as 0 or -2. Many of the lowest ratings in CREDBANK were due to differing interpretations of the word “event”, rather than being reflective of actual veracity.

Buntain & Golbeck (2017) were able to generalize their CREDBANK-trained model by onto other datasets by including non-textual features based on user attributes and the structure of tweet threads, suggesting that NLP may not by itself be a viable method for validating natural disaster tweets. That said, the issues with the CREDBANK leave the question open, as it is possible that a similar dataset that is rigorously cleaned and vetted could perform well. Moreover, a robust dataset that is specifically about disaster tweets could also have potential as a training set for ML algorithms. For future research into verification of disaster tweets we'd recommend one of the following approaches:

1. Use a purpose-built dataset comprised of disaster-related tweets that are manually labelled as rumour or not-rumour.
2. Use a broader feature set that includes textual and non-textual features in evaluating the veracity of disaster-tweets.
3. Select a web app host that can support a neural net.
4. Explore if transformers could perform better.

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

Hunt K, Agarwal P, Zhuang J. Monitoring Misinformation on Twitter During Crisis Events: A Machine Learning Approach. Risk Anal. 2020 Nov 14. doi: 10.1111/risa.13634. Epub ahead of print. PMID: 33190276.

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.