Fighting for Information Credibility: An End-to-End Framework to Identify Fake News during Natural Disasters

Dipak Singh
Louisiana State University
dsingh8@lsu.edu

Shayan Shams
University of Texas
Shayan.Shams@uth.tmc.edu

Joohyun Kim
Louisiana State University
jhkim@cct.lsu.edu

Seung-jong Park
Louisiana State University
sjpark@lsu.edu

Seungwon Yang[*]
Louisiana State University
seungwonyang@lsu.edu [†]

ABSTRACT

Fast-spreading fake news has become an epidemic in the post-truth world of politics, the stock market, or even during natural disasters. A large amount of unverified information may reach a vast audience quickly via social media. The effect of misinformation (false) and disinformation (deliberately false) is more severe during the critical time of natural disasters such as flooding, hurricanes, or earthquakes. This can lead to disruptions in rescue missions and recovery activities, costing human lives and delaying the time needed for affected communities to return to normal. In this paper, we designed a comprehensive framework which is capable of developing a training set and trains a deep learning model for detecting fake news events occurring during disasters. Our proposed framework includes infrastructure to collect Twitter posts which spread false information. In our model implementation, we utilized the Transfer Learning scheme to transfer knowledge gained from a large and general fake news dataset to relatively smaller fake news events occurring during disasters as a means of overcoming the limited size of our training dataset. Our detection model was able to achieve an accuracy of 91.47% and F1 score of 90.89 when it was trained with the first 28 hours of Twitter data. Our vision for this study is to help emergency managers during disaster response with our framework so that they may perform their rescue and recovery actions effectively and efficiently without being distracted by false information.

Keywords

Neural Networks, Social Network, Natural Disaster, Fake News, Deep Learning.

INTRODUCTION

Social media platforms have become an integral part of the life styles of many individuals since such platforms allow easy access to diverse information and rapid interactions among their users. Unequivocally, the growing impact of fake news has been increasingly affecting the integrity of the society in a harmful way. Therefore, detecting fake news and understanding its characteristics and mechanisms of dissemination constitutes the community-wide responsibility for the stability and the sustainability of the society in a modern and open internet ecosystem.

In spite of many prior studies conducted to detect fake news primarily relied upon social media contents, due to the non-trivial task of semantic modeling dealing with both misinformation and disinformation (Rubin et al.

[*] corresponding author
[†] https://www.lsu.edu/chse/slis/about_us/bios/yang.php

2015), a successful strategy is highly likely to be achieved only when we understand the complexity on how they are engaged and disseminated (Vosoughi et al. 2018; Conroy et al. 2015; Qazvinian et al. 2011; Shu et al. 2017). Consequently, it is a holistic approach that takes into account multiple characteristics of fake news namely, user engagement, user profile, and the dissemination of entire fake news event, along with the fake news texts or the content-based information (Ruchansky et al. 2017). In this work, we introduce an end-to-end solution by leveraging recent advances in Deep Learning, which is intrinsically suitable for combining diverse sets of features without hand-crafted feature engineering and it is also inherently scalable for large datasets (Ruchansky et al. 2017).

In order to address the complexity of requirements for an effective solution to deal with malignant fake news spread in the time of disasters, we introduce a pipeline framework for detecting fake news events which delivers false information to a large population quickly and broadly on Twitter. It is worth mentioning that our framework is capable of dealing with other types of fake news events (i.e., non-disaster domains) as well. Our framework is streamlined with the Twitter data collection component and the Deep Learning-based data analysis component. Notably, the data analysis, which models a classification task, incorporating multiple features (such as user features) as input along with the temporal sequence of fake news propagation. This analysis is designed to be effective in cases with an insufficient size of the training dataset, which is commonly faced during a specific disaster event.

In the following section, we introduce several studies which are most relevant to our study, followed by the methodology, experiments, and the results and discussion sections. Finally, we conclude with a brief summary of our study and plans for future work.

RELATED STUDIES

Pierri and Ceri (2019) present a comprehensive survey of publications on recent advances in fake news studies in three categories: the detection of false news; the characterization of fake news spreading in social media; and the mitigation approaches to alleviate the impact of such news on communities (Pierri and Ceri 2019). For research contributions on detecting false news, the authors further group the publications into three categories based on the features analyzed: the content of fake news, (social) context of fake news which propagate in social media, and combination of both content and context. Content-based approaches identify linguistic features from the textual content of fake news (W. Y. Wang 2017; Popat et al. 2018; Pérez-Rosas et al. 2018; Horne and Adali 2017; Potthast et al. 2018; Hosseinimotlagh and Papalexakis 2018). Manually-engineered feature sets are often used along with traditional machine learning and deep neural network models. One of the early studies for fake news detection, which focused on classifying the relative stances of news content to its title, also goes into this category (Hanselowski et al. 2018). Social context-based approaches aim to distinguish fake news cascades (which are a group of social media posts sharing the same fake news) by analyzing the social aspect of the fake news which diffuse in social media (Tacchini et al. 2017; Volkova, Shaffer, et al. 2017; Y. Wang et al. 2018; L. Wu and H. Liu 2018; Y. Liu and Y. B. Wu 2018). The profiles of users involved in fake news activities, interactions between users, and interactions between users and fake news are all important features to consider in identifying fake news cascades. Context-based approaches may also analyze the spatiotemporal features (e.g., time stamps and geolocation data of Twitter posts) as long as these features are provided as part of the dataset. Finally, the authors introduce the combined approach, which mixes both content-based and social context-based approaches, as the latest and most effective methodology for fake news detection (Ruchansky et al. 2017; Volkova and Jang 2018; Shu et al. 2017).

Considering that the combined approach is the latest and the most effective one, we have also based our proposed detection model on this approach. In the following paragraphs, we further introduce selected studies, which have incorporated the combined approach in their methodologies. These studies use datasets which have multiple dimensions (e.g., texts, user profiles/interactions, spatiotemporal data), and also apply deep neural network models (e.g., Convolutional Neural Net, Recurrent Neural Net) to detect fake news and to facilitate the study of fake news detection in social media.

Ma et al. (2016) developed an approach which converts Twitter data into variable-length time series and then trains four recurrent neural network (RNN)-based models (tanh-RNN, long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997; Graves 2013), gated recurrent unit (GRU) (Cho et al. 2014) with 1-layer hidden units, and GRU with an extra hidden layer) for rumor detection problem (Ma et al. 2016). Deep neural network models, including RNN models (Rumelhart et al. 1988), demonstrated advantages over traditional machine learning models. The authors shared their own training datasets after developing them by using both manual and automatic processes based on Twitter and Weibo (Chinese social media) platforms. Their algorithm converts incoming microblog streams into a variable length time series data. This approach captures and represents the densely populated regions in the diffusion stream for analysis. The model performances were compared to those of traditional machine learning models, and it showed outstanding performance for the Weibo dataset in terms of the accuracy, precision, recall, and F measures. Although their GRU models performed well for the Twitter data in terms of the accuracy and F measure, traditional support vector machine (SVM) models showed slightly better performance for precision and recall measures. Our study also adopts an RNN model and use Ma et al.'s Twitter dataset (i.e., Rumdect dataset) as part of the training model. However, the difference is that our model includes a user module component to address the social aspect of the fake news propagation. In addition, our model is further fine-tuned using our own disaster fake news Twitter dataset applying the transfer learning (TL) scheme.

The hybrid model for detecting fake news diffusion in social media, presented by Ruchansky et al. (2017) (Ruchansky et al. 2017), focuses on three characteristics of fake news: textual content of the fake news, user responses the fake news receives, and the sources of the fake news (e.g., structure of URLs, publisher, users). Their model consists of three components, namely Capture, Score, and Integrate. The textual content of fake news and temporal aspect of user-to-news interactions are analyzed by the Capture module using a paragraph embedding technique (e.g., doc2vec) (Le and Mikolov 2014) and the volume of fake news posts per unit time, respectively. In order to uncover the source characteristics, their Score module operates on all users and then depicts frequencies of the user-to-user engagement mediated by fake news articles. Finally, the results from both Capture and Score modules are concatenated in the Integrate module for predicting whether each article is fake or real. Ruchansky et al. adopts the same social media data (Twitter and Weibo) collected by Ma et al. However, their approaches for preparing training datasets are not identical. For Ma et al., they produce variable-length time series data for training their models out of the collected social media data whereas Ruchansky et al. partition the social media data into 1-hour segments and then each of which becomes an input to a cell for more efficient training of the model.

One of potential limitations of Ruchansky et al.'s model is that their model depends heavily on user-to-user interactions in the Score module, which presides over all users in the training dataset (Ruchansky et al. 2017). This design consideration may make the entire model less flexible and perform less effectively when the model has to predict labels for an unseen set of users and articles with which those users interacted. Our model design is similar to that of Ruchansky et al.'s in that our model has the Content Aware module, which corresponds to their Capture module. Our model also has the Context Aware module, which corresponds to their Score module. However, the differences are that: (1) we attempt to resolve the issue of their model's dependency on user-to-user interactions by representing users with their Twitter profile data; and (2) we adopt the TL scheme to overcome the insufficient size of our training dataset, which is derived from fake news events (tweets) in disaster domain. We hope that this study will help the emergency organizations, institutions such as libraries and schools, and the affected public to filter out false information during the critical times of natural disasters. This will increase the community resilience as well.

METHODOLOGY

Data Collection

As part of our framework, we have a distributed tweet collection system infrastructure. We developed this to allow for convenient collection, storage, and filtering of Twitter data. The system collects requested tweets based on keywords and locations of the tweet posts, filters the tweets, and then stores them into the MongoDB database as shown in Figure 1.

Ground Truth Label: To collect the ground truth labels for the fake news, we rely on the fact-checking website such as Snopes.com. Snopes.com is a well-regarded evidence based source for filtering out myths and rumors and misinformation on the Internet (*Snopes is the internet's definitive fact-checking resource* 2020). Snopes.com classifies a news article into broader spectrum of ratings (e.g., false, mostly false, mixed, mostly true, unproven, miscaptioned). However, we only consider those articles which are labeled as 'false' and 'mostly false' as fake, and 'true' as real for the ground truth labels. We use web automation tool, Selenium (*SeleniumHQ Browser Automation* 2019), to periodically collect the relevant articles to create our ground truth labels.

Social Context: To ensure better search results, the keywords to collect relevant Twitter posts are generated manually based on the headlines of articles in Snopes.com. Related Twitter posts are then collected by using the Search and Streaming APIs, provided by the Twitter's Advanced Search API. Twitter's Streaming API helps us to define a bounding box of geolocations for the tweets that need to be collected, whereas Search API allows the hashtag- or keyword-based collection of tweets. We utilize both of these features to collect the disaster-related tweets. We also used the Twitter's Search box to search and collect the fake news-related tweets explicitly. Once the tweets are collected, we apply de-duplication and filtering processes to remove duplicate tweets that will inevitably be collected from both types of APIs and filter the noise (irrelevant tweets) respectively, based on the text, time interval, and geolocation to study our fake news spatio-temporal patterns. After we obtain social media tweets, which are directly relevant to the fake news, we further fetch metadata such as user profiles, replies, likes, retweets, and social network information from those tweets to create our training dataset to train our proposed detection model described in Detection Model section.

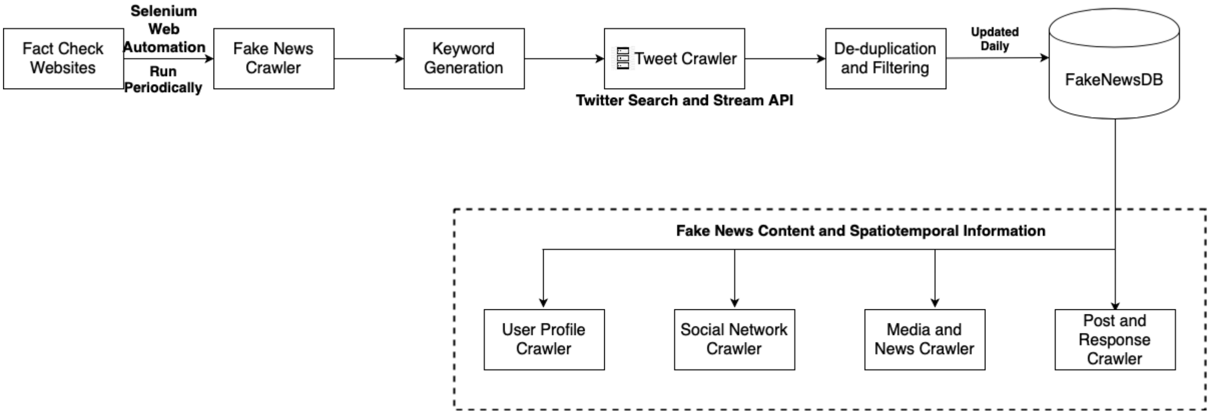

Figure 1. Twitter Data Collection System Infrastructure

Data Preparation

Our model is comprised of two modules-Content Aware and Context Aware modules and the input data is prepared accordingly. Following the model proposed by Ruchansky et al. (Ruchansky et al. 2017), the input data to our Content Aware module is generated. Each Twitter event e is considered as a temporal sequence of sets of tweets within t time interval. Each engagement between a user u_i and an event e_i at time t is represented as a tuple of $X = (\epsilon, \Delta T, X_u, X_t)$, where ϵ is the number of tweets engaged in a particular time interval ΔT. X_u is the row in the adjacency matrix between the event e_i and user u_i involved in ΔT, and X_t is the numeric vector representation of the tweet texts within ΔT. For example, let there be $e_1, e_2, ..., e_n$ events. Each event e_i is comprised of a series of tweets $t_1, t_2, ..., t_k$ by different users $u_1, u_2, ..., u_l$. Furthermore, we partition the series of tweets into different sets $s_1, s_2, ..., s_m$ based on ΔT time interval. Therefore, X is the vector of tuples $(\epsilon, \Delta T, X_u, X_t)_1, (\epsilon, \Delta T, X_u, X_t)_2, ..., (\epsilon, \Delta T, X_u, X_t)_m$, where ϵ is the total number of tweets in a set s_i. To calculate X_u, we first create an adjacency matrix of all the users and events, where each row represents the number of times a user u_i involved in each event e_1 to e_n. Moreover, we take the mean of the collection of rows (or users involved in a set s_i) of the adjacency matrix and represent it as X_u. In the end, we reduce the dimensionality of the adjacency matrix using PCA (with dimension 20). Similarly, in order to calculate X_t, we consider the collection of tweets in a set s_i and then use doc2vec to produce the vector embedding.

For the Context Aware module, various metadata from the Twitter user profiles are collected based on those twitter users involvement in a particular event e_i using our data collection system described in Data Collection section . Each user's number of tweets shared, number of followers, number of following, number of likes, the links or media shared, the age of their profile, if their account is private or verified, and their geocode locations are all taken into account. The user profile collected and processed in this way may capture user characteristics based on their activities in the network. Thus, our Context Aware module should be able to distinguish users who may have different levels of susceptibility for fake news. In Ruchansky et al.'s study, user features are constructed based on the user-to-user interactions within the training dataset. Such features are dependent on the dataset and if the test set encounters new unseen users involved in the fake news, then such model would fail to capture new fake users.

Detection Model

In this section, we go over the deep learning model utilized for disaster-related fake news detection. Any news article contains three important features: the information an article carries (e.g., tweet text), the response to the article by various clients engaged with that article (e.g., replies to the tweet, the number of likes, the number of followers, and the temporal information), and the source of the article (e.g., twitter account which posts that tweet and it's behavior on social site). Therefore, by incorporating these three features in our deep learning model, our model could learn the underlying data distributions for fake news behaviors in social media in a more comprehensive way to distinguish between a real and fake news. Inspired by Ruchansky et al. (Ruchansky et al. 2017), we constructed the content aware module to capture the temporal and textual fake news information. However, we added a context aware module to capture user features based on the individual twitter user profile. As previously mentioned, the main drawback of Ruchansky et al.'s model is the dependency on user-to-user interactions in the Score module. This design consideration may make the entire model less flexible and perform less effectively when the model has to predict labels for an unseen set of users and/or articles. We tackled this limitation by designing the content aware module to characterize each user's information based on it's profile. The second implementation constraint that we handled is the fact that the disaster-related dataset is relatively small in comparison with fake news dataset used in previous works. Therefore, we utilized TL to overcome this limitation. We will go over the implementation in detail in the rest of this section.

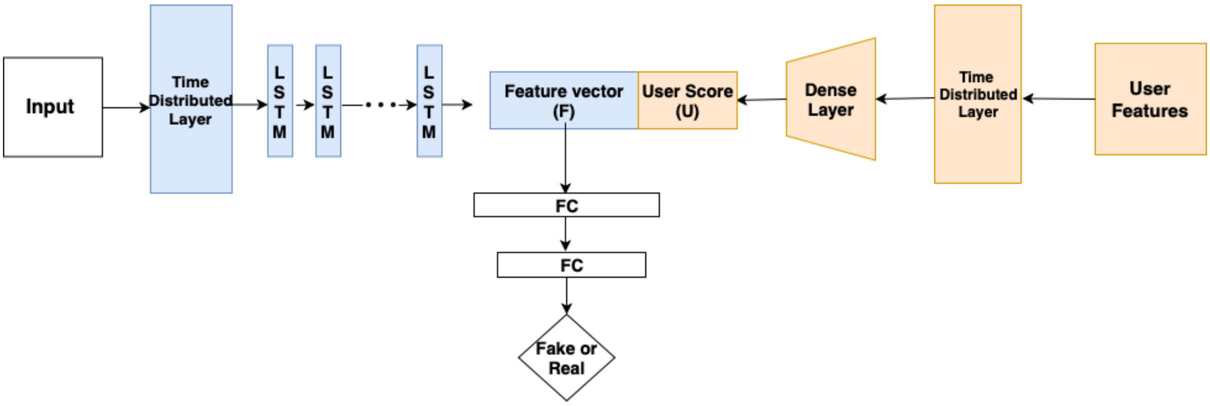

Figure 2. Model Detection Specification

Content Aware Module

The goal of this module is to capture the temporal and textual features of the fake/real events. We incorporated a time distributed embedding layer to standardize the input X before feeding it to the RNN. To capture the temporal characteristics, we used the Long Short-Term Memory (LSTM) model, since LSTM models are good at capturing long-term dependencies and can also handle variable length inputs. The final hidden layer of LSTM is then passed through a fully connected layer to get a low-dimensional feature vector F which represents the temporal and textual characteristics as shown in Figure 2.

Context Aware Module

This module captures the user features based on the Twitter user profile information. The user feature matrix is constructed based on their user profile and then Principal Component Analysis (PCA) is applied. PCA not only helps in reducing the dimensionality, but also enables us to capture the most important variables in the original feature space. Therefore, the dimension of the user feature matrix is reduced from eight to four (based on the elbow in the curve as a function between the dimensions and the explained variance, with no rotation). Given the set of low-dimensional user features (of dimension 4), we then apply dense layers to extract the user feature vector U and apply masking to select only those users that are involved in a particular batch. The output of both the modules are concatenated and the whole model is trained jointly.

Training with Transfer Learning

In the disaster domain we face the limitation of the sufficient amount of data, due to the limited time window of the disaster event. TL is a traditional novel approach to handle training on small datasets and achieve a robust model, given we have similar larger data distribution available from different domains. Thus, in our TL, we have pre-trained our model using a bigger dataset and then transfer the learning on a specific (but relatively smaller) natural disaster dataset (Pan and Yang 2010). Therefore, the pre-trained model have the same architecture and was trained with an existing larger fake news event dataset (Rumdect). In this first step, a rich set of generic features were extracted from the data in diverse topical areas. Then, the fine-tuning with our disaster data allows our model to be trained in spite of the smaller disaster dataset (Table 1).

Note that the TL scheme is only applied to the Content Aware module. The Content Aware module is then jointly trained with Context Aware model for the end-to-end supervised task. The Loss function for training our model is composed of the cross entropy loss ($\mathcal{L}_{\text{ce}}$) and the L_2 regularization as follows.

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda \|W\|_2^2$$

Table 1. Overview of the datasets used for training.

	Rumdect Dataset	Disaster Dataset
No. of Events	991	91
No. of Fake Events	497	46
No. of Real Events	494	45
No. of Users	226,791	31,305
Total No. of Tweets	569,912	37,975
Avg. Event Period (hrs)	1,961	3,924
Avg. No. of Tweets/event	575	417
Max No. of tweets	37,475	4,041
Min No. of tweets	4	6
Avg. No. of tweets/user	2	1

The weights for Dense Layer and Time distributed layers are randomly dropped out for training to reduce overfitting. Therefore, we have created an end-to-end framework to collect the fake news during a disaster and then prepared the dataset to feed it to the model. The model is then trained via TL to make predictions. We trained the Context Aware module initially on the larger Rumdect dataset, then the weights of the model were transferred to the natural disaster dataset to handle the small dataset limitations. All the RNN models are trained using Tensorflow 1.8 and tested with Nvidia GeForce GTX 1080Ti. The Rumdect dataset is divided into 80% training, 5% validation and 15% testing dataset, whereas the natural disaster dataset accuracy is measured using the 5-fold cross validation.

EXPERIMENTS

Datasets

Since our natural disaster dataset is generated based on the Twitter data, we considered only the Twitter data part from the publicly-available Rumdect dataset for pre-training step as discussed in Transfer Learning section. This dataset originally contained more than 5,000 events with five million relevant tweet IDs (Ma et al. 2016; *Twitter Ids (snowflakes)* 2010). However, when we hydrated those tweet IDs to utilize their full tweet texts and metadata, some of the tweets were found to be no longer existing due to account suspension or deleted posts. Thus, the Rumdect dataset that we hydrated has 991 events and approximately 570,000 relevant tweets. Each event comprises a news story, tweet IDs associated to that news, and the label (real or fake). Upon hydration of the tweet IDs, we get each event with the set of engagements (tweets) made by a user u_i at time t. For our disaster dataset, we have collected 91 instances of fake/real news events along with the profiles of users engaged in the events following the methodology in Data Collection section . The statistics of both the dataset is listed in Table 1.

Model Setup

We partition each engagement of the dataset into segments of ΔT, time interval such that all the engagements within ΔT time are taken as one input to the LSTM cell.

Hyperparameters: We used cross-validation to set the regularization loss parameter to $\lambda = 0.01$, the dropout probability as 0.4, the learning rate to 0.001, and used the Adam optimizer. Furthermore, we considered 1 hour granularity as the partition unit ΔT. For training the doc2vec model, we used the window size of 10 and a vector size of 100. The weights used in the Context and Content Aware modules are of dimension 100.

RESULTS AND DISCUSSION

Fake News Event Detection Performance

In Figure 3, the detection accuracies and the F1-score (the harmonic mean of the precision and recall) obtained with our model for different event periods are presented. The different time lengths of event periods from 6 to 48 hours with one hour increment are used and indicated with x-axis labels. Regarding the partition scheme, which is one hour interval, if there is no tweet activity in a certain partition in an event, we skipped it and considered only non-empty partitions in the dataset.

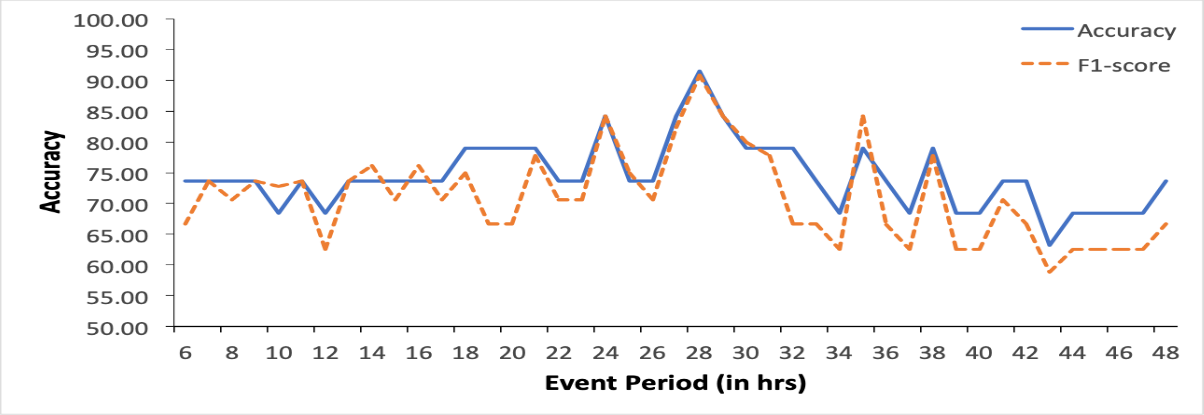

Figure 3. Accuracy and F1 score of our model trained with different sizes of event periods.

The best accuracy of 91.47 % was achieved when we used the first 28 hours of tweets as an event period in our disaster dataset. The same condition resulted also in the best F1 score of 90.89. Since in natural disaster domain only classifying the tweets is not enough, rather classifying them within smallest amount of time is also crucial. Therefore, we assessed different time intervals on which we can have the best trade-off between the sensitivity analysis (the overall best accuracy) and the threshold assessment (the minimum number of hours required to achieve the best performance), and found the 28hrs of training to be the right interval which gives us the best accuracy. With this, our model demonstrated the capacity of capturing the textual, temporal, and user engagement/profile aspects of the fake news events in social media successfully.

In spite of reasonably good performance, thanks to TL-based training, the model architecture, and datasets we collected, there exist a challenge to understand the overall results shown in Figure 3. For example, more data with a longer event period do not seem to improve the accuracy if the event period becomes longer than 28 hours. There might be two potential reasons for such degradation of the overall model performance: (1) the influx of noisy data and (2) a transition to a more complex data distribution. For (1), it might be related to the fact that noisy and unrelated tweet data were introduced, especially after around 28 hours since the beginning of the fake news occurrences in social media. For example, noisy tweets could have been added to our disaster dataset when social media marketers abused popular hashtags in fake news tweets for their marketing purposes. Another potential explanation might be (2) the transition to different model distribution. After a certain period, different mechanisms to generate fake news events may start to emerge, departing from the data distribution of an early event period (up to 28-hour period) readily detected by the current model. An understanding of whether such mechanisms could turn out to be difficult to model or could be captured by more sophisticated models would be beyond the scope of this work. However, unraveling such a time-dependent behavior we found in this work would be an interesting future work for the nature of disaster fake news detection.

Transfer Learning

Table 2. Accuracies for different model setup and with/without TL.

Model Setup	Without TL	With TL
Content Aware Only	78.94	84.21
Content + Context Aware	86.15	91.47

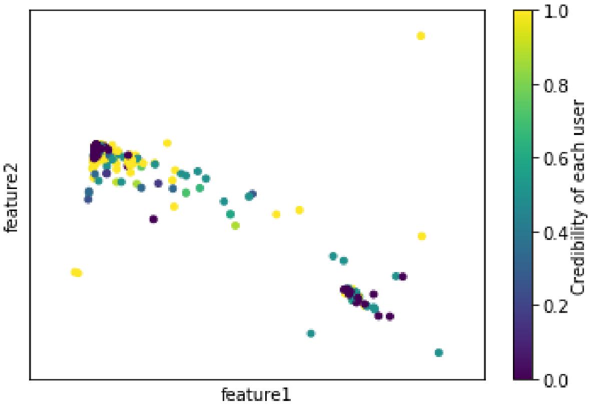

Figure 4. User characteristics (Credibility score 0: users involved only in real news events; Credibility score of 1: users involved only in fake news events).

As described in Detection Model section , our detection model mainly consisted of two parts: Content Aware and Context Aware modules. The Content Aware portion of the model was trained with TL and the Context Aware module captured the user profile information from social networks that they participated. In order to examine the effects of these different model setup and the role of TL, we computed accuracies using the first 28 hours of event data with different settings.

Combining the Context Aware module along with the Content Aware module had a significant effect on the accuracy with 7.24 % increase in average regardless of the application of TL. This tells us that our Context Aware module could successfully extract useful information from the profiles of users participating in social networks to distinguish false from real information activities. Applying TL to the Content Aware portion also positively affected the accuracy with a 5.3% increase in average regardless of adding the Context Aware module to the detection model. Thus, the issue of insufficient size of training dataset in the disaster domain could be addressed successfully.

If we compare the Content Aware only model without TL (78.94 %) and the Content + Context Aware model with TL (91.47 %), the accuracy difference is much more significant (12.53 %), and it clearly shows the benefits of using the user profile information, as well as TL for the false information detection in disaster domain.

Representing User Characteristics

We calculated the credibility score of each user based on their average participation in the fake or real news events in social media. If a user was involved in fake events only, his or her credibility score became 1. If the user was only involved in real events, his or her credibility score became 0. Hence, the credibility of each individual user was $C_i \epsilon [0, 1]$. We considered the 20% random samples of the data and extracted the user features on our trained model. We projected the first two scalar values of the user vector obtained after training, namely feature1 and feature2, with respect to the credibility score of the users. Figure 4 shows that the users of high credibility scores (i.e., black and navy dots) were clustered together into two groups located on the top left and on the bottom right. Also the users of low credibility scores (i.e., green and yellow dots), who have a higher tendency to share fake news content, are presented between the two groups of users with high credibility. This clustering result indicated the effectiveness of separating the users involved either in fake or real news events, by considering the features extracted from their user profiles.

Existing Challenges and Our Contributions for the Disaster Domain

Compared to other domains such as politics, stock market, or celebrities, the size of generated fake news events in disaster domain was relatively smaller. Insufficient size of training datasets in deep learning and machine learning may lead to a creation of an overfitting model. Thus, this insufficient data size was one of main challenges in this study, which we could overcome by applying TL for the Content Aware module in our model (See Table 2). Considering that we can make a bigger dataset by collecting fake news events from the forth-coming natural disasters, our model has a potential to be further fine-tuned continuously.

Another challenge was to identify what the optimal length of the event period should be for our dataset to achieve the best performance. We could find that the model trained with the event period length of 28 partitions showed the best performance (See Figure 3). Training with the event period longer than that had an adverse effect on the performance.

Our main contributions for the disaster response domain are two-fold: (1) applying the TL methodology and refining a prior fake news detection model by Ruchansky et al. to address the problem of fake news events occurring during disaster domains and (2) identifying the optimal length of the event period to train and detect fake news events which occur during natural disasters. This may mean that emergency managers performing rescue/recovery efforts could be benefited by identifying devastating false information in social media during disasters within a fairly short time period of 28 hours with a high accuracy (91.47%). Especially, (2) also implies that such early and accurate detection of fake news events could lead to a reduced costs for disaster response (e.g., dispatch rescuers to correct locations and save lives on time), as well as recovery actions (e.g., mobilize food, water, shelter, etc. to the hands of victims and affected on time).

CONCLUSION AND FUTURE WORK

In this study, we aimed to develop a comprehensive framework which was capable of creating a training dataset by collecting and processing fake news-related Twitter data, training an RNN-based detection algorithm, and then making decisions whether a given cascade of Twitter data might be a fake news event or not. Our application domain was natural disasters, and it gave us challenges considering that the number of fake news events occurring during the time of crises, although potentially critical for the safety of the victims and affected, were relatively smaller compared to other domains. To overcome this challenge, we incorporated the TL scheme, which pre-trained our detection algorithm with a larger Rumdect dataset and then fine-tuned the algorithm with our Disaster dataset (See Table 1).

We computed the model accuracies and F1 scores by increasing the length of the hourly partitions for the event period (1-hour increments). We could observe that the best performance was achieved when the model was trained with the first 28 hours of tweet event data (Figure 3) and increasing the event length only degraded the performance. Further investigations might be necessary to understand the potential causes of this decreased performance in the presence of longer events (tweet activity data). It is also important to note that since our goal in this paper is to develop a detection algorithm, investigating the motivation behind the fake news behavior is beyond the scope of this paper. As a future work, we plan to create a man-made disaster fake news dataset as well (e.g., mass shooting, train crashes, chemical pollutant leakage, etc.) to further fine-tune our detection model so that our model could be capable of dealing with diverse types of both natural and man-made disaster fake news situations. In our baseline study, we considered only the textual data of Twitter posts. We plan to expand our study to cover multimedia (such as images and videos) and other embedded URLs in Twitter posts as well. We also plan to experiment with several ideas such as constructing ego networks of Twitter users for the Context Aware module, using different text embedding approaches (e.g., BERT) for tweet texts, etc. to improve our model.

ACKNOWLEDGEMENTS

This work has been partially supported by a grant from National Science Foundation IBSS-L 1620451.

REFERENCES

Cho, K., Merrienboer, B. van, Bahdanau, D., and Bengio, Y. (2014). “On the Properties of Neural Machine Translation: Encoder-Decoder Approaches”. In: *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pp. 103–111.

Conroy, N. J., Rubin, V. L., and Chen, Y. (2015). “Automatic deception detection: Methods for finding fake news”. In: *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*. American Society for Information Science, p. 82.

Graves, A. (2013). “Generating Sequences With Recurrent Neural Networks”. In: *CoRR* abs/1308.0850. arXiv: 1308.0850.

Hanselowski, A., S., A. P. V., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C. M., and Gurevych, I. (2018). “A Retrospective Analysis of the Fake News Challenge Stance Detection Task”. In: *CoRR* abs/1806.05180. arXiv: 1806.05180.

Hochreiter, S. and Schmidhuber, J. (Nov. 1997). “Long Short-Term Memory”. In: *Neural Comput.* 9.8, pp. 1735–1780.

Horne, B. D. and Adali, S. (2017). “This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News”. In: *CoRR* abs/1703.09398. arXiv: 1703.09398.

Hosseinimotlagh, S. and Papalexakis, E. E. (2018). “Unsupervised content-based identification of fake news articles with tensor decomposition ensembles”. In: *MIS2, Marina Del Rey, CA, USA*.

Le, Q. V. and Mikolov, T. (2014). “Distributed Representations of Sentences and Documents”. In: *CoRR* abs/1405.4053. arXiv: 1405.4053.

Liu, Y. and Wu, Y. B. (2018). “Early Detection of Fake News on Social Media Through Propagation Path Classification with Recurrent and Convolutional Networks”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 354–361.

Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K., and Cha, M. (2016). “Detecting Rumors from Microblogs with Recurrent Neural Networks”. In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pp. 3818–3824.

Pan, S. J. and Yang, Q. (2010). “A Survey on Transfer Learning”. In: *IEEE Transactions on Knowledge and Data Engineering*.

Pérez-Rosas, V., Kleinberg, B., Lefevre, A., and Mihalcea, R. (2018). "Automatic Detection of Fake News". In: *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pp. 3391–3401.

Pierri, F. and Ceri, S. (2019). "False News On Social Media: A Data-Driven Survey". In: *CoRR* abs/1902.07539. arXiv: 1902.07539.

Popat, K., Mukherjee, S., Yates, A., and Weikum, G. (2018). "DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 22–32.

Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., and Stein, B. (2018). "A Stylometric Inquiry into Hyperpartisan and Fake News". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 231–240.

Qazvinian, V., Rosengren, E., Radev, D. R., and Mei, Q. (2011). "Rumor has it: Identifying misinformation in microblogs". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 1589–1599.

Rubin, V. L., Chen, Y., and Conroy, N. J. (2015). "Deception detection for news: three types of fakes". In: *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*. American Society for Information Science, p. 83.

Ruchansky, N., Seo, S., and Liu, Y. (2017). "Csi: A hybrid deep model for fake news detection". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, pp. 797–806.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). "Learning representations by back-propagating errors". In: *Cognitive modeling* 5.3, p. 1.

SeleniumHQ Browser Automation (2019). URL: https://www.seleniumhq.org/ (visited on 10/11/2019).

Shu, K., Sliva, A., Wang, S., Tang, J., and Liu, H. (2017). "Fake news detection on social media: A data mining perspective". In: *ACM SIGKDD Explorations Newsletter* 19.1, pp. 22–36.

Snopes is the internet's definitive fact-checking resource (2020). URL: https://www.snopes.com/about-snopes/ (visited on 10/01/2020).

Tacchini, E., Ballarin, G., Vedova, M. L. D., Moret, S., and Alfaro, L. de (2017). "Some Like it Hoax: Automated Fake News Detection in Social Networks". In: *CoRR* abs/1704.07506. arXiv: 1704.07506.

Twitter Ids (snowflakes) (2010). URL: https://developer.twitter.com/en/docs/basics/twitter-ids (visited on 09/30/2010).

Volkova, S. and Jang, J. Y. (2018). "Misleading or Falsification: Inferring Deceptive Strategies and Types in Online News and Social Media". In: *Companion Proceedings of the The Web Conference 2018*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, pp. 575–583.

Volkova, S., Shaffer, K., Jang, J. Y., and Hodas, N. O. (2017). "Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pp. 647–653.

Vosoughi, S., Roy, D., and Aral, S. (2018). "The spread of true and false news online". In: *Science* 359.6380, pp. 1146–1151.

Wang, W. Y. (2017). ""Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pp. 422–426.

Wang, Y., Ma, F., Jin, Z., Yuan, Y., Xun, G., Jha, K., Su, L., and Gao, J. (2018). "EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pp. 849–857.

Wu, L. and Liu, H. (2018). "Tracing Fake-News Footprints: Characterizing Social Media Messages by How They Propagate". In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pp. 637–645.