ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE - EPFL

MASTER'S PROJECT

DISTRIBUTED INFORMATION SYSTEMS LABORATORY

Classifying Informative Tweets

Author:
GERALD SULA

Supervisor:
TUGRULCAN ELMAS

Responsible:
PROF. KARL ABERER

January 15, 2020

Abstract

Existing studies have shown the usefulness of systems such as supervised machine learning for detecting classes of messages in social platforms like Twitter in terms of informativeness. However there exist very recently made available Natural Processing tools, which have not been fully taken advantage of in this context. Furthermore previous studies have focused on limited numbers of sampled data, which was typically related only to natural-disaster. In this study we have aggregated large amounts of annotated Tweets coming from multiple different topics, and built several classifiers using a variety of NLP tools, including the very new BERT transformer models. We demonstrate that the efficacy of such models at identifying informative tweets is higher compared to other recent approaches, as well as make available a new annotated dataset aggregated from 7 of the larges available sources.

Introduction

Social media became one of the primary sources of news and information on certain topics such as crisis. Platforms such as Twitter provide relevant information on a keyword that is related to the interest of the user. However, not all relevant information is informative: users use the platforms for public conversation as well, which might not be of interest to users seeking information. As social media platforms are massive, it is crucial to detect and identify to informative tweets and prioritize them in search results.

Previous work focused on Twitter and studied classifying informative tweets during disasters using textual and/or context features. They fall short of studying their methods' generalizability on different disaster related datasets and also non-disaster related datasets such as conflict and health related issues where informativeness of tweets are also crucial. In this work, we make the first attemp to address this issue by proposing a generalizable classifier. We evaluate our classifier both by standard machine learning classifier as well as information retrieval evaluation metrics to account for the information overload associated with social media. Our contribution are as follows:

1. We propose a generalizable classifier to classify informative tweets. We evaluate it on different public disaster-related datasets.
2. We make the first attempt to test if such a classifier can generalize to a health related dataset and a conflict related dataset.
3. We evaluate our classifier by search engine metrics. Our study is the first attempt to build a search engine for informative tweets.
4. We build a new informative tweets dataset to help researchers on future studies on the subject.

The organization of this paper is as follows: Section 2) we survey related work, Section 3) we present the datasets and analyze the characteristics n-gram which we believe helpful for our classification task. Section 4) We present our methodology in engineering features and selection classification algorithms. Section 5) we provide our experimental results. Section 6) we discuss our results and their implications.

Background and Related Work

(Neppalli, Caragea, and Caragea 2018; Caragea, Silvescu, and Tapia 2016; Nguyen et al. 2017) showed that deep learning models overperforms traditional classifiers using Bag of Words and handcrafted features. (Ghafarian and Yazdi 2020) proposed an approach learning on distributions. (Burel and Alani 2018) attempted to classify tweets using crisislex dataset according to different information types albeit with poor results.

SMM 82.4%

(Zhang and Vucetic 2016) proposed a semi-supervised approach to find informative tweets during disasters.

(Khatua, Khatua, and Cambria 2019) trained domain specific word vectors for outbreaks instead of general models.

(Madichetty and Sridevi 2020) proposed a classifier using images which might not work when the tweets has no images attached to them.

Other related work include detecting disaster related tweets using supervised learning (Palshikar, Apte, and Pandita 2018) and a lexicon (**?**).

Datasets

We have conducted an extensive search for datasets containing information about tweets related to different global events, that were annotated in terms of informativeness or news-worthiness. We present here the list of datasets found. The events covered in these sets are related to different topics ranging from natural disasters, to political conflicts and health crisis.

Dataset 1, CrisisLexT26: This collection includes tweets collected during 26 large crisis events in 2012 and 2013. Contains 25K tweets, labeled for informativeness, information type, and source. (Olteanu, Vieweg, and Castillo 2015) **Dataset 2, CrisisMMD**: Consists of 16K manually annotated tweets and images collected during seven major natural disasters that happened in 2017 across different parts of the World. The provided datasets include three types of annotations. (Informativeness, Humanitarian categories and Damage severity assessment) (Ofli, Alam, and Imran 2020; Alam, Ofli, and Imran 2018) **Dataset 3, NewsNotNews**: This dataset consists of the identifiers of 2.7K tweets related to different news topics such as Health, terroristic attacks, natural disasters, science and education. Every tweet has been annotated for informativeness. (University of Duisburg-Essen, Germany and Aggarwal 2019) **Dataset 4, WNUT-2020 Covid19**: Dataset containing 8k tweets, collected during the Covid 19 pandemic. The data has been labeled for informativeness by 3 independent annotators, obtaining an iner-annotator score of 0.818. (Nguyen et al. 2020) **Dataset 5**: Collection of tweets related to 2 nat-

ural disasters (Joplin tornado 2011 and hurricane Sandy 2012), consisting of 1k tweets annotated for informativeness. (Imran et al. 2013) **Dataset 6, CrisisNLP**: Dataset containing tweets related to 19 different events around the world (Natural disasters, War & conflicts, biological and different accidents). Totals 22k entries, each labeled into different categories for information type. (Imran, Mitra, and Castillo 2016) **Dataset 7, Karabakh Tweets**: Collection of 260K unlabeled tweets related to conflicts in the Karabakh region. We have manually annotated 200 tweets for informativeness in order to see how our model performs with tweets related purely to politics.

Each dataset contains a different set of features, such as additional information about the event covered, multiple categories per type of news, or some information about the Twitter. But each of them contained at least these two features: the unique identifier of the tweet as well as an annotation based on the informativeness of the tweet. It is these two fields that we use when cleaning and homogenizing the data as well as when scrapping content from Twitter for additional features on the tweets and the users that sent them.

Methodology

Data filtering and transformation

The first step we took when dealing with the data, was to first get the text of the tweets from the datasets that provided only a reference to the tweet id. Afterwards, we filter any non English tweets, and drop all the different features not relevant to our study, so that for every dataset we know the text of the tweets, the informativeness label, and the id of the tweet. There were some datasets that provided the label as multiple categories regarding the type of information, so we merged the categories to simply informative/non-informative.

Feature augmentation

Since we know the ids of every tweet, now we can make use of Twitters API and grab additional information on the authors and the tweets themselves, that can be useful when training our models. We are adding the following features:

•Tweet content Features

1. **retweet_count** The number of times the tweet was retweeted.
2. **favorite_count** The number of times the tweet was liked.
3. **tweet_type** A boolean indicating if a tweet is a new tweet or a retweet of another tweet.

•User-based Features

1. **followers_count** The number of followers the author has.
2. **friends_count** The number of friends the author has.
3. **listed_count** The number of communities that the author is listed in.
4. **favorites_count** The total number of liked tweets by the author.
5. **statuses_count** The number of tweets posted by the author.
6. **has_description** Boolean indicating weather the Twitter account of the author contains a description.
7. **bio_has_url** Boolean indicating weather the short bio of the user also contains a link.
8. **protected** Boolean indicating weather the account of the user is protected (has restricted access).
9. **verified** Boolean indicating weather the account of the author has been verified by Twitter. Typically true only for famous and largely followed accounts.
10. **default_profile** Boolean that indicates if the user has not altered the theme or background of their user profile.
11. **default_profile_image** Boolean that indicates if the user has not altered the profile picture of their account.

Since a lot of the tweets included in the datasets are from a few years before the time this project was conducted, there is a proportion of tweets that have been either deleted by their authors or removed by Twitter, so we are unable to find any of the features described above. For those cases, we take the mean/median (depending on the feature) of the other tweets of the same dataset that had the corresponding informativeness label.

Text processing

Once we have filtered out all non-english tweets, we apply a processing pipeline to every tweet which consists of removing all stopwords of the English language, lemmatizing the remaining words and lower-casing everything. This will decrease the vocabulary size and will be very useful for the next steps.

In order to use text as part of the input, we utilize the Bag of Words Technique (BOW), which consists of vectorizing the text of every tweet. We do this by creating a fixed size vector for each tweet, the length of which is equal to the total number of unique words in the entirety of the datasets. We assign the total number of times a word is present in a tweet, to the appropriate position in the vector. For the vectorization, we have tested two ways of creating tokens for the vocabulary: unigrams and bigrams. The choice between the two is decided separately for each of the models we will present, during a hyper-parameter tuning step. A second optional transformation is applied to the vectors, which is multiplying every term by their TF-IDF (term frequency–inverse document frequency) score. This is intended to reflect how important a word is to a document in a dataset. Weather this transformation is used or not, is also tested during the hyper-parameter tuning of each model.

Models

In this project, we have built and evaluated the performance of several models which can be devided in

two categories: Traditional Models using BOW and Models using word embeddings. (Neppalli, Caragea, and Caragea 2018; Caragea, Silvescu, and Tapia 2016; Nguyen et al. 2017) have showed that the more traditional models underperform in term of classification compared to more sophisticated deep-learning models. We will start by describing the approach we followed for the more traditional models and then discuss the more complex ones.

•Models using Bag of Words

Each one of the following models takes as training input the BOW matrices we computed from the text of the tweets, together with the informativeness label of each tweet. For each one of them we are trying a BOW matrix which was built using both unigrams and bigrams of the input text, as well as with and without a multiplication factor that corresponds to the TF-IDF score of the terms. This choice is performed in the hyperparameter tuning phase, during which we also test a variety of different parameters which are different from model to model. The models which were tested are the following:

1. **Support vector machine (SVM)** Parameters tested: Loss$\in \{Hinge, Log, Perceptron\}$, Maximum iterations $\in \{10, 100\}$, Alpha $\in \{1e-5, 1e-3\}$
2. **Logistic Regression (LR)** Parameters tested: Penality$\in \{L2, L1\}$, Maximum iterations $\in \{100, 1000\}$
3. **Multinomial Naive Bayes (NB)** Parameters tested: Alpha $\in \{1, 1e-1, 1e-3\}$
4. **Decision Tree (DT)** Parameters tested: Max depth of tree: $\in \{200, 400, 1000\}$
5. **Random Forrest (RF)** Parameters tested: Minimum samples to split $\in \{4, 10\}$

•Models using Word Embeddings

1. **Neural network using Glove embeddings**

   GloVe is an unsupervised learning algorithm for obtaining vector representations for words created by (Pennington, Socher, and Manning 2014). Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

   We begin by tokenizing the processed text of each tweets, which assigns a unique id to each term of the vocabulary, and constructing vectors for each tweet, containing the token ids of that sentence. Since the vectors have different sizes, depending on the number of tokens in each tweet, we are padding with 0 so that they all have the same length.

   Afterwards we create an embedding matrix, which will contain the Glove embedding of only the tokens present in our vocabulary. We are using a pre-trained word-vector set which contains 6 Billion tokens, and an embedding dimension of 300.

   Using Keras (Chollet and others 2015), we have built a neural network which consists of an Embedding layer that uses the weights of the embedding matrix, a global Max pooling layer, and two dense layers (using the Relu and Sigmoid activation functions). The model is compiled using 'Adam' as an optimizer, and is trained on batches of size 10. We have tested this model in two ways: by disabling the training of the word embeddings in the embedding layer, and also by enabling it. By doing this, the computational time it takes to train the model is drastically increased, but it can lead to better results.

2. **Convolutional Neural Net Model**

   The next model we tried is a CNN using word embeddings as input. This is a replication of the paper published by (Neppalli, Caragea, and Caragea 2018; Caragea, Silvescu,and Tapia 2016; Nguyen et al. 2017). Convolutional Neural Networks are a category of Deep Neural Networks, based on the concepts of local receptive field and weight replication. CNNs consist of combinations of convolution and sub-sampling layers, which help extract meaningful representations of the input data.

   For this part of the study, to keep the replication as close as possible the original paper, we are also performing the additional feature expansion as described in the original paper. Namely we are adding 18 features that are part of Tweet Content Features, User Features and Polarity-based Features.

   In the original paper, in terms of the word embeddings, a pretrained dataset of word2vec embeddings was used, which was trained on crisis related tweets. However this dataset was not available to download, so in our replication we had to make use of other pretrained word2vec embeddings. So instead we are using the largest collection of pretrained embeddings made available by (Mikolov, Le, and Sutskever 2013)

   In addition to that, we are also using BERT embeddings, similar to what we use in the next model we will present (Sanh et al. 2020). More details on how BERT embeddings work will be presented in the next section. The CNN model has a single convolution layer followed by max-over-time pooling layer and a fully connected layer. We use the word embeddings of the tweets as input to this model, which is reconstructed as a matrix. Multiple filters of different sizes are used to convolve this matrix, the outputs of which are fed to a pooling layer that scans for the maximum value. The output feature maps corresponding to the filters used are finally concatenated to get the feature representation of input. The dimension of each word vector is 300 when using word2vec embedding, and 768 when using BERT embeddings. As in the original paper, we used three convolution filter sizes: 3 * 150, 4 * 150 and 5 * 150, and a dropout ration of 0.5.

3. **BERT Model**

The last model we developed, uses pretained BERT embeddings (Devlin et al. 2018). BERT (Bidirectional Encoder Representations from Transformers) and its many variants have successfully helped produce new state of-the-art performance results for various NLP applications. In this study, we have tested 3 of the best know variants of BERT: vanilla BERT, RoBERTa (Liu et al. 2019) and DistilBERT (Sanh et al. 2019). However, in this paper we will be presenting only the results when using pretrained DistilBERT embeddings, as there was very little difference between the 3, and DistilBERT was much faster to run for the quantity of data we have.

A first model we built was constructed by using the Simple Transformer package (Rajapakse), which provides a very simple to use tool to make predictions of textual data. Although this first model was performing well, ultimately we decided to develop a new model ourselves. The main reason why we did this was because the provided package only worked with textual data, meaning that we had no way to add the additional numerical features we have computed. And in addition to that, it was not clear how the inner workings of the classifier used in this package worked, so we were unable to make any adjustments that would better suit our data.

Instead, we simply used the pretrained DistilBERT data to extract the embeddings of each cleaned and processed tweet text, using the Sentence-Transformer package (Reimers and Gurevych 2019). By doing this we construct a sparse matrix of 768 columns (the dimension of BERT embeddings) and one row per each tweet, which was used as input to a classification model.

This allows us to add the additional features we have computed, and vertically stack them alongside the aforementioned matrix. Since the values of the embedding columns, and the new feature columns are very differently scaled, we standardize the matrix using a standard scaler which removes the mean and scales the data to unit variance.

Finally we are able to perform a binary classification task on the computed matrix by using a hyper-parameter tuned Ridge Regression model. Parameters tested for the model are:
Alpha $\in \{1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3\}$,
Normalize$\in \{True, False\}$.

Experimental Results

In this study we are interested in building a classifier that performs very well on classifying tweets for their informativeness. So we are interested in 3 different scenarios: First we want to see how our models perform when they are trained on a mix of datasets containing tweets related to many different topics. Second, how well the classifiers would perform when we train and test on datasets of different topics, and then compare the different test-cases. And finally, the most difficult task, we want to see if the models we have prepared are able to transfer knowledge about informativeness when trained on twitter data that is disaster-related, and evaluated on new data that is mainly health-related or conflict-related.

1. **Training/testing on all datasets separately**

The first task we will be presenting is the following: We are checking the performance of the different models by training each model once for each of the available datasets, and using the trained model to predict the informativeness on all the datasets one by one. So we are creating a grid, where every dataset is used both as a training set and as a testing set, and we see how the models perform for all the different combinations of datasets. We are using heatmaps to visualize the results, which will be helpful when comparing the different classifiers.

Note that the metric we are using is the ROC-AUC score of the classifiers. We are not interested at checking simply the accuracy, since we are want to have an idea of how the models perform in terms of metrics more suited towards information retrieval.

The last dataset, which contains data related to conflicts in the Karabakh area was excluded from this task, since it is very small compared to the rest, and training only on 200 samples while testing on multiple thousands is not meaningful.

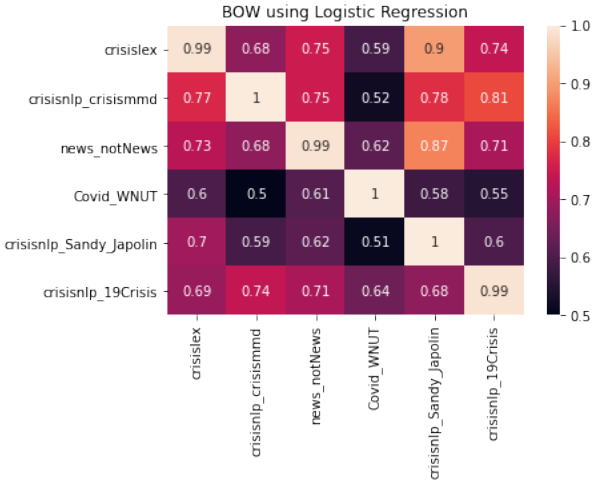

Figure 1: Best model using BOW

We are presenting in Figure 1 the best performing model out of the 5 descried in the section "Models using Bag of Words". Showing how all 5 performed would not be particularly interesting since they all performed very close to each other (with the exception of the Decision Tree and Random Forrest classifiers which performed noticeably worse), and would make this document much more cluttered.

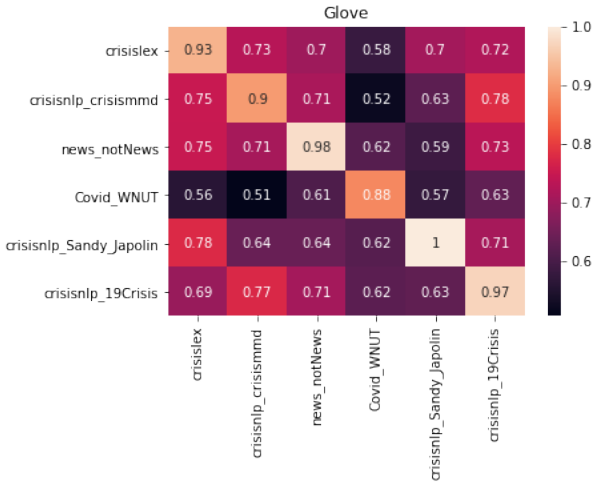

Figure 2: Glove Model

Figure 2 shows how the Glove classifier performed. When performing this task, we decided to disable the option that sets all the parameters of the embedding layer to trainable, since this was leading to very similar results, but with a much higher time complexity to run.

For the CNN classifier that we replicated from the Neppalli at al. paper, we repeated this task twice, once using word2vec embedding as input and a second time using DistilBert embeddings. The results (Figure 3) are quite similar, with some cases the word2vec embeddings performing better and in others the Bert embeddings.

Lastly, we present the performance of the BERT model in Figure 4.

When comparing the different classifiers, we are able to see that gnerally the BOW model is performing worse than the more complex models using word embeddings. It's not a staightforward task to see which one of the models is performing the best, but in the majority of cases the DistilBERT model yields the best results, with the Glove model following second. It is clear to see from the heatmaps, that when training on the 4th dataset (WNUT-2020 Covid19), which is vastly different from the rest of the datasets, the results are much lower. The same thing goes when testing on this dataset, although to a lesser extent.

2. **General classifier using all the datasets merged**

The next task we have performed is evaluating the models, when using all the datasets together as one big dataset. The rows of the merged dataset have been shuffled and we have split the data int train and test sets using 30% of the data for testing. The metrics used to evaluate the performance are the ROC-AUC score, as well ad the F1-score. In order to better compare the results of the models built for this study, we are using as reference the CNN model replicated by the Neppalli et al. study, using the exact same neural net shape and word2vec embeddings, in order to have a model as close as possible to the original

Figure 3: CNN model using 2 different embeddings

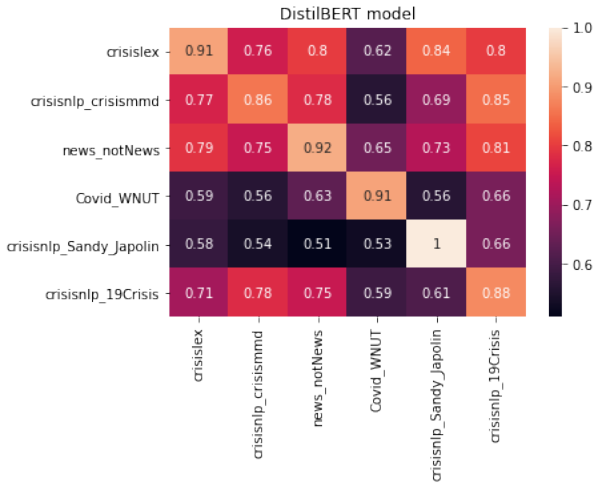

Figure 4: DistilBERT model

paper.

General classifier results		
Model	ROC-AUC score	F1-score
CNN using word2vec	0.695	0.784
Glove classifier	0.860	0.871
DistilBERT classifier	0.867	0.877

As we can see from the results, both models we have constructed perform much better than the CNN model used for comparison. The results for Glove and Bert are close, but the latter model has the best scores for both ROC and F1.

3. **Transferring knowledge between data type**

The last task we will describe is also the hardest one. We have developed a first attempt at generalizing a classifier trained on data predominately related to natural-disaster tweets, to make prediction on the informativeness on tweets which are health and conflict related. So in order to achieve this, we are using as training input to the classifiers all the tweets coming from the natural disaster related datasets (namely datasets 1,2,3,5,6). After training on this data, we then predict and evaluate on dataset 4 (WNUT-2020 Covid19) which contains health related tweets, and dataset 7 (Karabakh Tweets) which contains conflict related tweets.

For this task we will be presenting only the results of the BERT model we have developed, since it has been consistently performing better in all the tasks. We will once again compare the results of our classifier, with that of the CNN classifier using word2vec embeddings from the Neppalli et al. paper. The following table describes the results in trems of ROC score.

Transferring knowledge results		
Model	COVID 19 tweets	Karabakh Tweets
CNN using word2vec	0.559	0.584
DistilBERT classifier	0.588	0.632

As we can see, the results for this tasks are very underwhelming. Both types of classifiers achieve low scores on both test sets, with a slightly better performance on the conflict related dataset. We can also observe a better score for both test cases when using the DistilBERT classifier.

Discussion

In this paper we have explored various state of the art techniques from the NLP repertoire and applied them in practice to the task of detecting informativeness in a very volatile context, such as the tweets sent by users on Twitter. Judging by the results we got from the tasks described above, it seems that newer techniques which rely on word embeddings and deep learning outperform the more classic Bag of Words models. We saw that there was no one model that very clearly is better at classifying informativeness, but that it very much depends from dataset to dataset. With that being said, the models we presented in this paper based on BERT embeddings and Glove embeddings, seem to have a visible edge compared to other state of the art models to which we compared them to. However, we were not able to successfully build a model that can generalize twitter data coming from a topic such as natural dis-tasters (the most widely available topic for annotated Twitter data regarding informativeness), to other topics such as health or political conflicts. Our classifiers, same as the others described in other papers, fall short on this task and achieved very low performance. This is clearly a very hard task to achieve, that would perhaps require other Natural Language Processing tools which were not considered in this study. Lastly, we built and are making available a new informative tweets dataset, by merging and homogenizing 7 separate datasets. This will hopefully help researchers on future studies on the subject. The datasets can be downloaded through this link:
https://github.com/byrek3d/Masters-Semester-Project

References

[Alam, Ofli, and Imran 2018] Alam, F.; Ofli, F.; and Imran, M. 2018. Crisismmd: Multimodal twitter datasets from natural disasters. In *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*.

[Burel and Alani 2018] Burel, G., and Alani, H. 2018. Crisis event extraction service (crees)-automatic detection and classification of crisis-related content on social media.

[Caragea, Silvescu, and Tapia 2016] Caragea, C.; Silvescu, A.; and Tapia, A. H. 2016. Identifying informative messages in disaster events using convolutional neural networks. In *International Conference on Information Systems for Crisis Response and Management*, 137–147.

[Chollet and others 2015] Chollet, F., et al. 2015. Keras. https://keras.io.

[Devlin et al. 2018] Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

[Ghafarian and Yazdi 2020] Ghafarian, S. H., and Yazdi, H. S. 2020. Identifying crisis-related informative tweets using learning on distributions. *Information Processing & Management* 57(2):102145.

[Imran et al. 2013] Imran, M.; Elbassuoni, S.; Castillo, C.; Diaz, F.; and Meier, P. 2013. Practical extraction of disaster-relevant information from social media. In *Proceedings of the 22nd international conference on World Wide Web companion*, 1021–1024. International World Wide Web Conferences Steering Committee.

[Imran, Mitra, and Castillo 2016] Imran, M.; Mitra, P.; and Castillo, C. 2016. Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA).

[Khatua, Khatua, and Cambria 2019] Khatua, A.; Khatua, A.; and Cambria, E. 2019. A tale of two epidemics: Contextual word2vec for classifying twitter streams during outbreaks. *Information Processing & Management* 56(1):247–257.

[Liu et al. 2019] Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach.

[Madichetty and Sridevi 2020] Madichetty, S., and Sridevi, M. 2020. Classifying informative and non-informative tweets from the twitter by adapting image features during disaster. *Multimedia Tools and Applications* 79(39):28901–28923.

[Mikolov, Le, and Sutskever 2013] Mikolov, T.; Le, Q. V.; and Sutskever, I. 2013. Exploiting similarities among languages for machine translation. *CoRR* abs/1309.4168.

[Neppalli, Caragea, and Caragea 2018] Neppalli, V. K.; Caragea, C.; and Caragea, D. 2018. Deep neural networks versus naive bayes classifiers for identifying informative tweets during disasters. In *ISCRAM*.

[Nguyen et al. 2017] Nguyen, D. T.; Al-Mannai, K.; Joty, S. R.; Sajjad, H.; Imran, M.; and Mitra, P. 2017. Robust classification of crisis-related data on social networks using convolutional neural networks. *ICWSM* 31(3):632–635.

[Nguyen et al. 2020] Nguyen, D. Q.; Vu, T.; Rahimi, A.; Dao, M. H.; Nguyen, L. T.; and Doan, L. 2020. WNUT-2020 Task 2: Identification of Informative COVID-19 English Tweets. In *Proceedings of the 6th Workshop on Noisy User-generated Text*, 314–318.

[Ofli, Alam, and Imran 2020] Ofli, F.; Alam, F.; and Imran, M. 2020. Analysis of social media data using multimodal deep learning for disaster response. In *17th International Conference on Information Systems for Crisis Response and Management*. ISCRAM.

[Olteanu, Vieweg, and Castillo 2015] Olteanu, A.; Vieweg, S.; and Castillo, C. 2015. What to Expect When the Unexpected Happens: Social Media Communications Across Crises. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 994–1009. Vancouver BC Canada: ACM.

[Palshikar, Apte, and Pandita 2018] Palshikar, G. K.; Apte, M.; and Pandita, D. 2018. Weakly supervised and online learning of word models for classification to detect disaster reporting tweets. *Information Systems Frontiers* 20(5):949–959.

[Pennington, Socher, and Manning 2014] Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

[Rajapakse] Rajapakse, T. Simple Transformers.

[Reimers and Gurevych 2019] Reimers, N., and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

[Sanh et al. 2019] Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108.

[Sanh et al. 2020] Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

[University of Duisburg-Essen, Germany and Aggarwal 2019] University of Duisburg-Essen, Germany, and Aggarwal, P. 2019. Classification Approaches to Identify Informative Tweets. In *Proceedings of the Student Research Workshop Associated with RANLP 2019*, 7–15. Incoma Ltd.

[Zhang and Vucetic 2016] Zhang, S., and Vucetic, S. 2016. Semi-supervised discovery of informative tweets during the emerging disasters. *arXiv preprint arXiv:1610.03750*.