Natural Language Processing with Disaster Tweets

Nguyen Diep Linh L.NGUYEN.DIEP@STUDENT.TUE.NL

1. Introduction

With the development of technology, social media has become popular as people can create various content for different purposes, share it with other users, network and enhance the information flow. According to a paper, there are four main social media functions: (1) information dissemination, (2) disaster planning and training, (3) collaborative problem solving and decision making, and (4) information gathering (Chan et al., 2014). Such information is invaluable to manage crisis or disaster in preparedness, response and recovery (Krajewski et al., 2017). However, there is so much information about disaster generated from social media that is usually confusing to determine whether the information is exactly about disaster.

In times of emergency, Twitter has become a vital communication channel. Because smartphones are so common, users can use them to announce a real-time emergency. As a result, more agencies are interested in automating Twitter monitoring, such as disaster relief and news organizations. However, it's not always clear whether someone's remarks on Twitter are actually a disaster warning. For example, a user wrote "On plus side LOOK AT THE SKY LAST NIGHT IT WAS ABLAZE" to describe a sunset view. The author explicitly mentioned the word "ABLAZE", but she referred to its metaphorical meaning. This is clear to a human right away, especially with the visual aid from the picture in the post, but it is less clear to a machine.

In this paper, the challenge will be to build machine learning models to predict which tweets are about real disasters and which ones are not. This form of automatic tweet classification can be valuable both during and after any natural disasters. During the disaster, it can aid crisis management, inform first responders, and others in taking appropriate action. After the disaster, analysts can use social media data to better understand how individuals behaved during the event. This kind of knowledge is crucial for increasing risk communication and protective decision-making before and during catastrophes, and thereby reducing potential harm (Demuth et al., 2012).

2. Related work

Multiple studies have been carried out to analyze disasters with social media (Wang and Ye, 2018) (Said et al., 2019), with applications in sentiment analysis (Beigi et al., 2016) and situational awareness (Verma et al., 2011). Besides, there have been many efforts in classifying Tweets to extract relevant information in social media based on natural language processing (NLP) techniques. The paper by Klein et al. (2013) suggested filtering the real-time media stream by assessing the message seriousity, extracting facts using NLP, and grouping posts using a novel event identification technique to detect emergency events in Twitter streams. The classification system of flu disaster by Lamb et al. (2013) first categorized related tweets and then applied finer-grained classifying techniques.

Similar classification system have been studied to classify tweets in more general domains, such us related to news, events, and opinions (Sriram et al., 2010), or labeled as reporting, sentiment, information, action, preparation and movement (Stowe et al., 2016). It can also be applied for sentiment analysis to identify and classify sentiment-expressing tweets (Kouloumpis et al., 2011).

3. Data

The data is about disaster tweets, which are original from Datasets Resource Center by Appen (https://appen.com/datasets-resource-center/). This project tackles a challenge on Kaggle, where the data are made publicly available. There are two dataset files, the training data and the testing data. The train data is the train.csv file with 7613 rows and 5 columns, and the test data is the test.csv file with 3263 rows and 4 columns. Each sample in the train and test set has the following information: the text of a tweet, a keyword from that tweet (although this may be blank), the location the tweet was sent from (may also be blank). The goal is to predict whether a given tweet is about a real disaster or not. If the given tweet is about a real disaster, predict a 1, otherwise, predict a 0.
The data columns are:

- id: a unique identifier for each tweet
- text: the text of the tweet
- location: the location the tweet was sent from (may be blank)
- keyword: a particular keyword from the tweet (may be blank)
- target: in train.csv only, this denotes whether a tweet is about a real disaster (1) or not (0)

3.1 Exploratory Data Analysis

There are 0.8% of keyword is missing in both training and test set, and 33% of location is missing in both training and test set. In the training set, there are 3271 disaster tweets (43%) and 4342 non-disaster tweets (57%). Thus, the distribution of the target classes is slightly unbalanced, which might result in having more prediction towards class 0. Besides, the distribution of the number of characters and words in tweets are plotted in Figure 1 and Figure 2. The average word length in each tweet category are shown in Figure 3.
From these figures, it can be seen that both classes have a similar distribution of number of characters, number of words and average word length. In general, there is not a special feature of any of them.

Next, the corpus (collection of words) is created to check the most common stopwords,

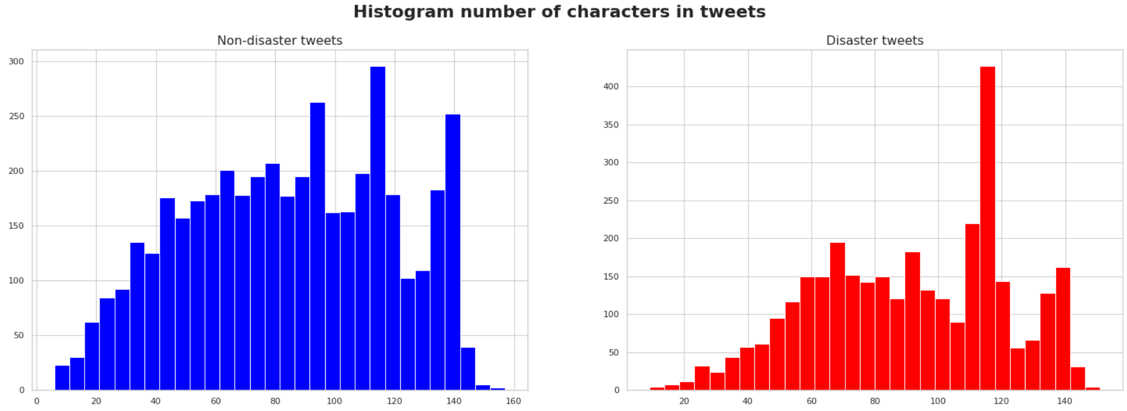

Figure 1: Histogram number of characters in tweets

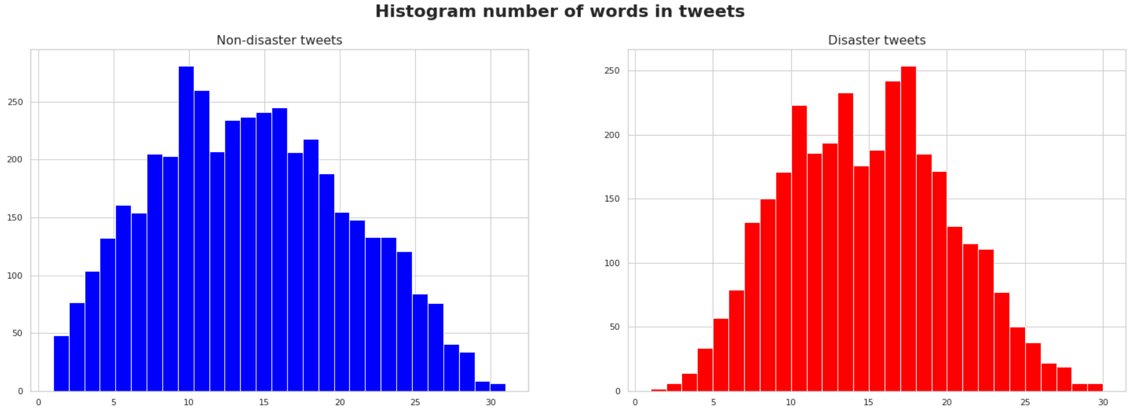

Figure 2: Histogram number of words in tweets

analyzing punctuations and visualize the most common words in each tweet classification. The figures are shown in Figure 4, 5 and 6.
The most common stopwords in non-disaster tweets are 'the', 'a', 'to', while the most common stopwords in disaster tweets are 'the', 'in', 'of'. In both tweet classes, the most common punctuation is hyphen, which is probably used to link the words of some specific names. From the Figure 6, the most common words are illustrated but they do not tell too much about the context and indicate about a real disaster. This suggests that the next step would be to clean the tweet texts so that they are more relevant for the modelling.

3.2 Data Cleaning

As indicated in the previous section, the data need to be preprocessed to make the prediction more accurate. The tweets contained in the dataset are almost raw, which have a lot of irrelevant information such as tags, symbols, punctuations, emojis, etc. These does not add

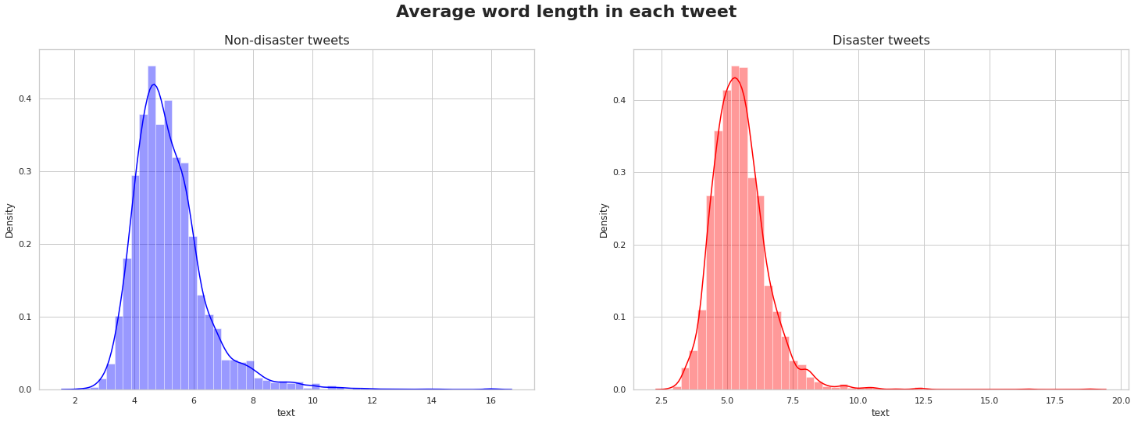

Figure 3: Average word length in each tweet

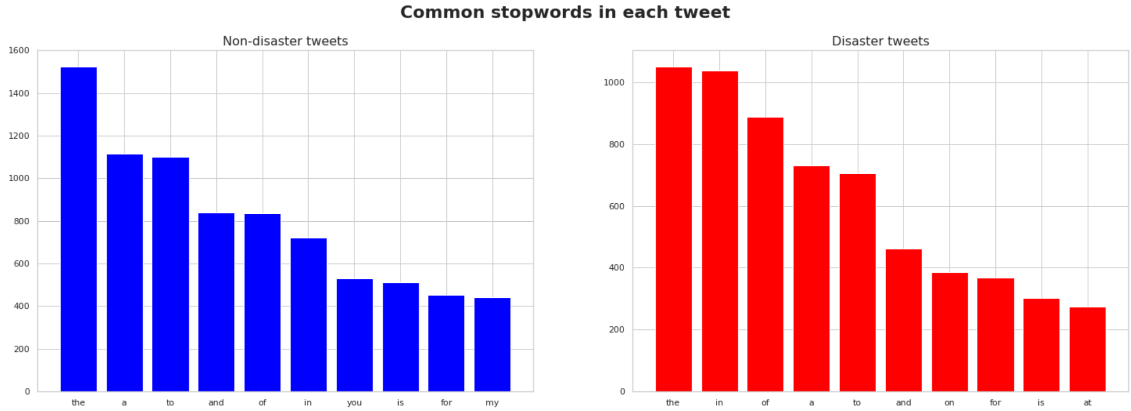

Figure 4: Common stopwords in each tweet

significant information to the prediction but will make the sentences more subjective. Thus, the following steps are taken to clean the data and return more relevant texts in training of the model.

- Removing URLs: Some tweets may have links ‘URLs’ which correspond to videos or other webpages containing key information about the subject they are trying to communicate. However, the link itself may not be informative and thus it should be removed to clean the sentences. The function to remove URLs is called remove_URL.

- Removing HTML tags: It needs to be considered that some tweets were obtained using web scrapping, by which there are some elements of the source identified by special tags. These tags may be irrelevant for the classification of the tweets. The function to remove these HTML tags is called remove_html.

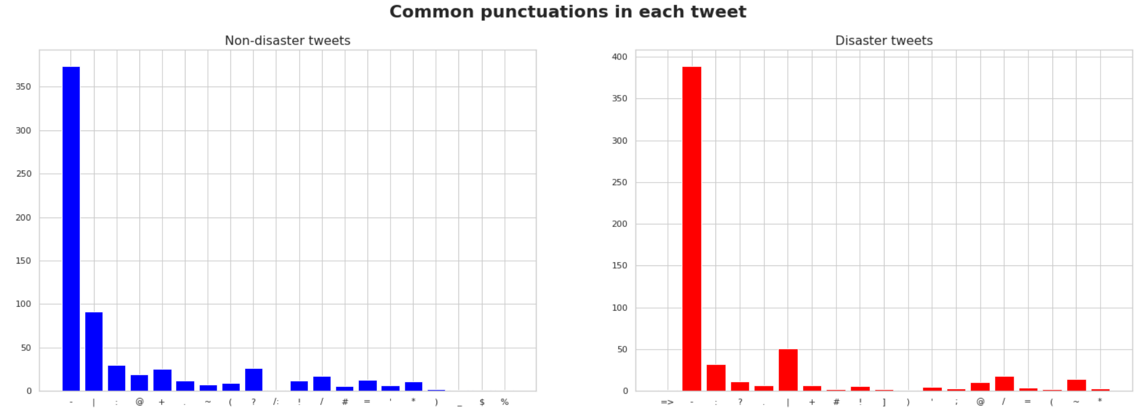

Figure 5: Common punctuations in each tweet

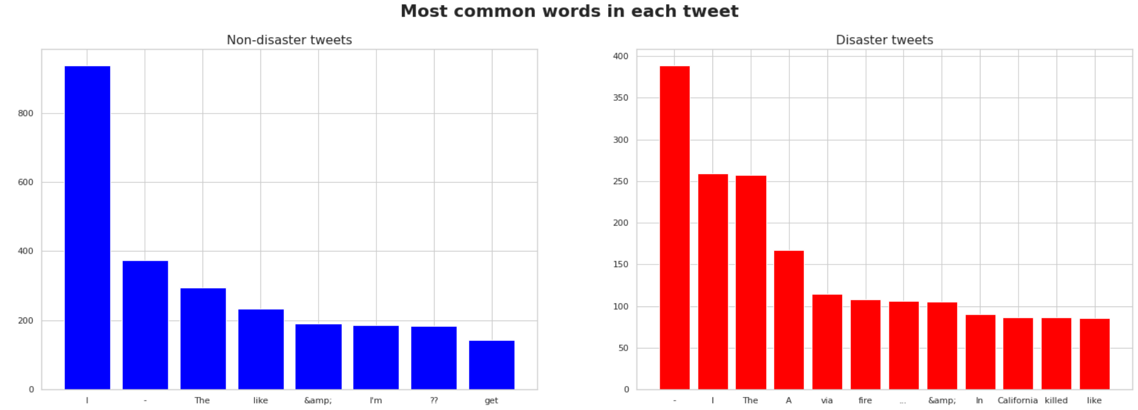

Figure 6: Most common words in each tweet

- Removing emojis: Emojis are useful to conveying the sentiment of the message's authors. By translating their meaning into words, the context of the tweets might be expanded. However, these could frustrate the prediction when finding the same emojis expression for disaster and non-disaster tweets. Thus, the emojis will be removed using function remove_emoji.

- Removing punctuations: Symbols and punctuations might be written in the tweets but they are uninformative for the classification. Thus, they will be removed using a function called remove_punct:

- Removing multiple spaces: Some tweets have different types of extra whitespaces, which do not add anything to the corpus and thus will be removed.

- Replacing contractions and acronyms: Acronyms are often used to make the writing faster and more convenient for the users. However, it may have spelling errors but can be split in smaller meaningful words. To search for the meaning of those words in a systematic and fast way, a function is created to replace the contractions and acronyms by the actual words they stand for.

- Removing numbers: To classify the tweets, the focus is on the embedding texts. Numbers could refer to something relevant for predicting the disaster, such as the coordinates, code of an accident, number of people killed, but it can also be irrelevant, such as the release year of a movie, number of followers on social media, etc. Thus, this project will not consider the use of these numbers in the model.

After cleaning the tweets, the most common words are plotted for each tweets category. The visualization is shown in Figure 7.

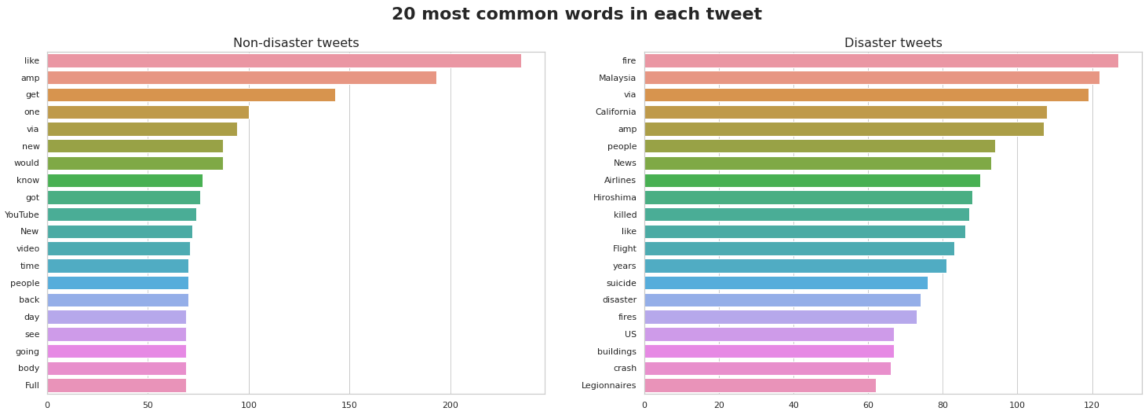

Figure 7: 20 most common words in each tweet

From the figure, it can be seen that non-disaster tweets mostly contain words that are irrelevant to a disaster, such as 'Youtube', 'New', 'video', 'time', 'back', 'going', etc. On the other hand, disaster tweets contain much more critical words, such as 'fire', specific names 'Malaysia', 'California', 'Hiroshima', 'killed', 'suicide', 'disaster', 'crash', etc. This indicates that the data cleaning process is important to filter the relevant information in the tweets.

4. Models

To measure the performance of the model, there is a train-test split from the training data, to get the training and validation set. The random state is set to be 100. Then, the words in the data are tokenized using CountVectorizer in sklearn. By doing this, the raw messages (sequence of characters) are transformed into vectors of token counts (sequences of numbers) to count the term frequency of each word.

Next, the term frequency–inverse document frequency (TF-IDF) is calculated to indicate the importance of a word to a document in a corpus. This number relates to how frequent a word appears in the document and is offset by the document frequency in the corpus that contain the word. This is calculated using the function TfidfTransformer() from sklearn.feature_extraction.text.

After that, the model is fitted on the training set and then assess on the validation set. The target of the validation data is predicted using the trained model. In the end, the prediction is assessed using accuracy score, the precision, recall, f1-score and the confusion matrix.

4.1 Baseline model - Multinomial Naive Bayes

The baseline model is a Multinomial Naive Bayes model, which uses the words contained in each tweet to indicate whether the tweet is about a real disaster or not. This classifier is suitable for classification with discrete features, such as word counts for text classification.

The model is fitted using MultinomialNB() function from sklearn.naive_bayes. After fitting the model, the accuracy score of the train set is around 0.79 and the accuracy score of the test set is around 0.56. The detailed result for the test set is shown in the table below:

	precision	recall	f1-score	support
0	0.91	0.58	0.71	1725
1	0.10	0.43	0.16	179
accuracy			0.56	1904
macro avg	0.50	0.50	0.43	1904
weighted avg	0.83	0.56	0.65	1904

The confusion matrix of the test set for this model is visualized in Figure 8. The model tends to predict the tweets to be non-disaster, and only predict a disaster in around 10% of the case.

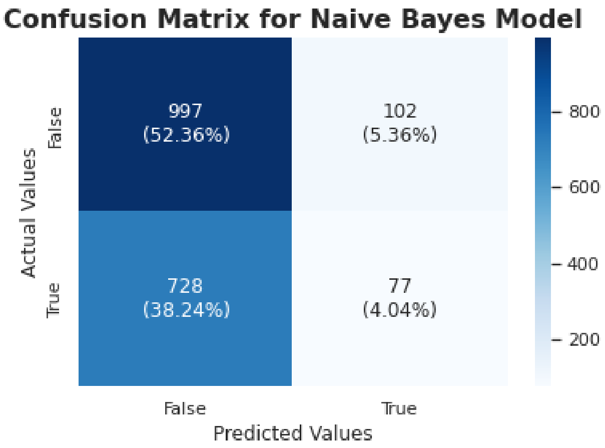

Figure 8: Confusion Matrix for Naive Bayes Model

4.2 XGBoost model

The next model to consider is the XGBoost (Extreme Gradient Boosting) model. This is a scalable and distributed gradient-boosted decision tree ensembles that is powerful for many machine learning tasks, including classification. It provides parallel tree boosting for training and predicting the class label of the tweets.

The model is fitted using XGBClassifier() with parameters use_label_encoder=False, eval_metric='auc', and random_state=100 from package xgboost. After fitting the model, the accuracy score of the train set is around 0.62 and the accuracy score of the test set is around 0.56. The detailed result for the test set is shown in the table below:

	precision	recall	f1-score	support
0	0.93	0.57	0.71	1788
1	0.05	0.34	0.09	116
accuracy			0.56	1904
macro avg	0.49	0.46	0.40	1904
weighted avg	0.88	0.56	0.67	1904

The confusion matrix of the test set for this model is visualized in Figure 9. Compared to the baseline model, this model predicts more disaster tweets and the false prediction rate is much lower than the baseline model.

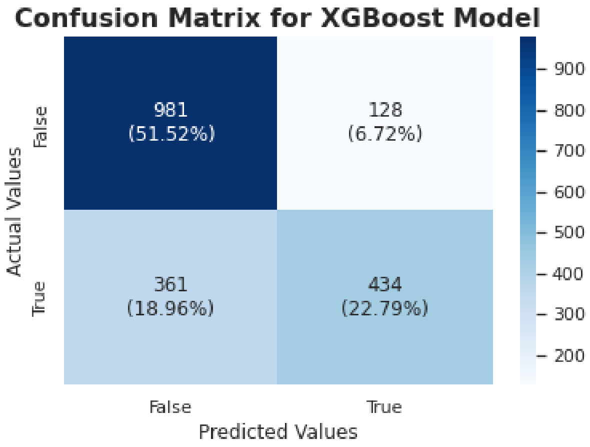

Figure 9: Confusion Matrix for XGBoost Model

4.3 GloVe LSTM model

GloVe (Global Vectors for Word Representation) is an unsupervised learning algorithm to obtain a vector representation for words. This concentrates on words co-occurrences over the whole corpus, which embeds how likely that two words appear together in the text. In this project, a Glove pre-trained model is used for text classification with 100 embedding dimensions.

Using the Glove-weights, an LSTM model is fitted, with the property of the layers as below:

```
Model: 'sequential'
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 31, 100)           1703100

 spatial_dropout1d (SpatialD  (None, 31, 100)          0
 ropout1D)

 lstm (LSTM)                 (None, 64)                42240

 dense (Dense)               (None, 1)                 65

=================================================================
Total params: 1,745,405
Trainable params: 1,745,405
Non-trainable params: 0
```

As can be seen, the model contains around 1.7 million parameters to compute, which would take a long time.

After fitting the model, the the accuracy score of the test set is around 0.82. Figure 10 visualizes the training and validation accuracy per epoch and the training and validation loss per epoch of the neural network model.

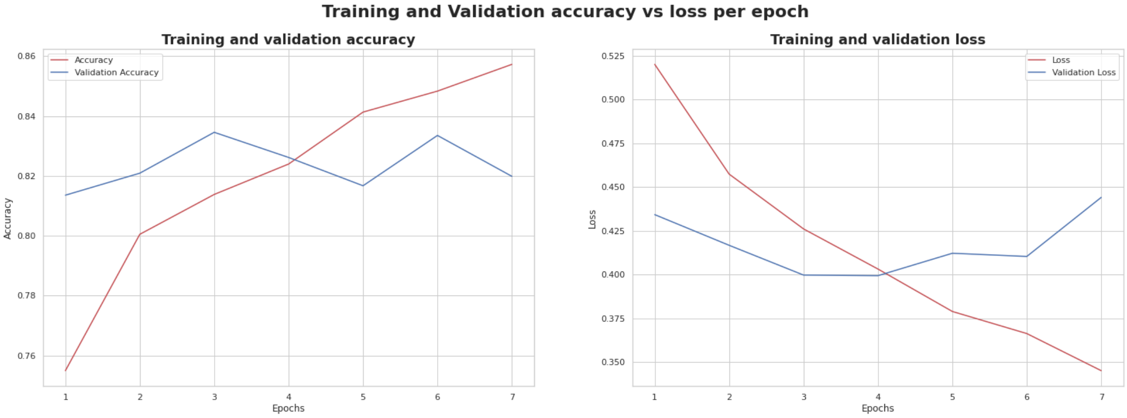

Figure 10: Training and Validation accuracy vs loss per epoch

The detailed result for the test set is shown in the table below:

	precision	recall	f1-score	support
0	0.85	0.84	0.85	1120
1	0.77	0.79	0.78	784
accuracy			0.82	1904
macro avg	0.81	0.82	0.81	1904
weighted avg	0.82	0.82	0.82	1904

The confusion matrix of the test set for this model is visualized in Figure 11. Compared to the previous two models, this model performs the best since it has the lowest false prediction rate, for around 18% of the case. The tweets are mostly classified correctly, with around 49.26% of the case to correct non-disaster tweets and 32.72% to correct disaster tweets.

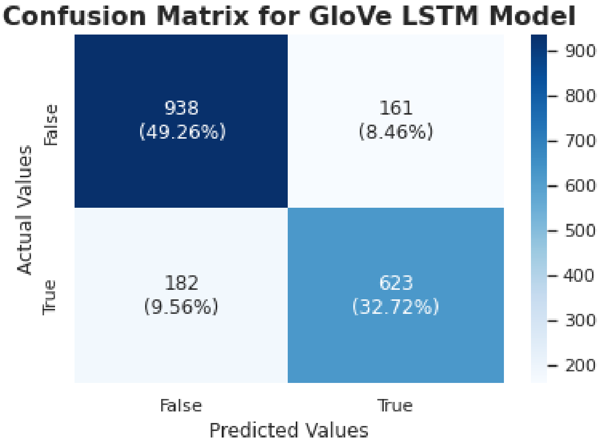

Figure 11: Confusion Matrix for GloVe LSTM Model

5. Conclusion

In this project, the challenge is to predict which tweets are about real disasters and which ones are not. After analyzing the data and cleaning the unimportant parts of the text, three models are fitted, namely Multinomial Naive Bayes, XGBoost and GloVe LSTM models. The first two models have the same accuracy score of 0.56 on the validation set, but perform differently in classifying the labels. The last neural network model has outperformed the first two, with the accuracy score of 0.82 on the validation set and can classify the true tweet label with far less errors.

References

G. Beigi, X. Hu, R. Maciejewski, and H. Liu. An overview of sentiment analysis in social media and its applications in disaster relief. *Sentiment analysis and ontology engineering*, pages 313–340, 2016.

J. C. Chan et al. The role of social media in crisis preparedness, response and recovery. 2014.

J. L. Demuth, R. E. Morss, B. H. Morrow, and J. K. Lazo. Creation and communication of hurricane risk information. *Bulletin of the American Meteorological Society*, 93(8): 1133–1145, 2012.

B. Klein, F. Castanedo, I. Elejalde, D. Lopez-de Ipina, and A. Prada Nespral. Emergency event detection in twitter streams based on natural language processing. In *Ubiquitous Computing and Ambient Intelligence. Context-Awareness and Context-Driven Interaction*, pages 239–246. Springer, 2013.

E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis: The good the bad and the omg! In *Proceedings of the international AAAI conference on web and social media*, volume 5, pages 538–541, 2011.

W. F. Krajewski, D. Ceynar, I. Demir, R. Goska, A. Kruger, C. Langel, R. Mantilla, J. Niemeier, F. Quintero, B.-C. Seo, S. J. Small, L. J. Weber, and N. C. Young. Real-time flood forecasting and information system for the state of iowa. *Bulletin of the American Meteorological Society*, 98(3):539 – 554, 2017.

A. Lamb, M. Paul, and M. Dredze. Separating fact from fear: Tracking flu infections on twitter. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 789–795, 2013.

N. Said, K. Ahmad, M. Riegler, K. Pogorelov, L. Hassan, N. Ahmad, and N. Conci. Natural disasters detection in social media and satellite imagery: a survey. *Multimedia Tools and Applications*, 78(22):31267–31302, 2019.

B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842, 2010.

K. Stowe, M. Paul, M. Palmer, L. Palen, and K. M. Anderson. Identifying and categorizing disaster-related tweets. In *Proceedings of The fourth international workshop on natural language processing for social media*, pages 1–6, 2016.

S. Verma, S. Vieweg, W. Corvey, L. Palen, J. Martin, M. Palmer, A. Schram, and K. Anderson. Natural language processing to the rescue? extracting" situational awareness" tweets during mass emergency. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, pages 385–392, 2011.

Z. Wang and X. Ye. Social media analytics for natural disaster management. *International Journal of Geographical Information Science*, 32(1):49–72, 2018.

Appendix

All the code for this project can be found at https://github.com/Linh181/NLP.