CS 221 Project Final Report

Guoqin Ma, Haotian Li

Project name:
Naive Bayes Classifier and Neural Network in Tweets Classification under Disastrous Events
Code and datasets: https://github.com/sebsk/AI_Project

1. Introduction

Social media is playing an increasingly important role in information collection and dissemination and it is changing our communication pattern significantly. Communication via social media platforms features large volume information, wide networks, quick responses, low costs, and intensive interactions. Especially during disastrous events such as flood, wildfire, earthquake, etc., posts posted by people involved in these disasters or bystanders sometimes could send out information much more early than traditional mass media. Thanks to these characteristics, since the rise of social media platforms such as Twitter and Facebook, they have become an effective channel for authorities to collect information when these extreme events occurred. However, the quality of information could not be guaranteed due to the participation from a vast number of users and its autonomous nature. A large fraction of the information collected from these platforms might be rumors, fake news, or totally irrelevant. Besides, during emergencies, agencies are expected to cope with the big data in short time. As Stieglitz et al. (2018) mentioned in their article, "We are still in the early stages of developing appropriate methods and information systems to overcome the existing risks and challenges of using these platforms under disaster and emergency conditions." Therefore, in order to make better use of social media to facilitate decision making in natural disasters, it is imperative to build up an accurate and efficient text classifier to identify the relevance of collected data, such as tweets, in order to transform the large and noisy data into relatively small and valuable data.

2. Task Definition

2.1 Overview

When utilizing those data from social media platform, we may encounter some difficulties. For example, the amount of information will be extremely large, but not all of them are related and informative. Search by keywords is not enough to filter the noise and there should be a second-step filtering where machine learning is involved (Leykin et al., 2018; Zahra et. al, 2017). This requires our classifiers being able to identify valuable information from unusable ones.

Data heterogeneity, as pointed out by Pekar et al. (2016), is an obstacle to text classification in a disaster context in that disasters of each type would have some unique features. Another challenge in this field is that labeled data are limited (Li et al., 2018), the majority of data available are unlabeled, which makes supervised learning difficult.

Therefore, we would investigate into the performance of 2 different classifiers, namely Multinomial naïve Bayes classifier and neural network (both CNN and RNN), in terms of disaster-informative tweets classification, by means of comparing their accuracy and stability. Accuracy is how accurate the classifier's prediction is. Stability is how stable the classifier is when it is used in different disastrous events. Moreover, we would dive into incremental learning or domain adaption approach to study which classifier evolves better using new unlabeled tweets data to predict a newly-occurring event.

2.2 Datasets

We use 4 datasets, namely, 2012 Colorado wildfires, 2013 Australia bushfire, 2013 Colorado floods, 2013 Queensland floods from CrisisLexT26 (Olteanu et al., 2015) in our project. Each dataset contains around 1000 labeled tweets. The tweets are filtered by keyword from tweets included in the 1% sample at the Internet Archive. The labels are according to informativeness (informative or not informative), information types (e.g. caution and advice, infrastructure damage), and information sources (e.g. governments, NGOs). The models are trained and tested on 2012 Colorado wildfires, and further evaluated with the rest 3 datasets.

2.3 Data pre-processing

- Removal

In our dataset, some of the Tweets were released by the government. The objective of this project is to differentiate that informative and effective information from social media. In this case, since the Tweets released by the government are already credible and reliable, they are not relevant to our study. That's why we have to remove them. Apart from this, some of the data have the label of "Not applicable", which is meaningless to classify. So this kind of data should also be removed.

- Modification to the labels

The original 3 labels of our data are "Related and informative", "Not related" and "Related - but not informative". However, what we really want to obtain is "Related and informative" ones. For those "Not related" and "Related - but not informative" information, we just want to discard them without caring about what specific label they have. Therefore, we could actually combine the label "Not related" and "Related - but not informative".

- Text pre-processing

The Tweet texts in the datasets we used usually contain some noisy or irrelevant information which may confuse our classifiers, for example, the URLs and smileys. Hence, text pre-processing is necessary.

Tweet-preprocessor library is used to erase emojis, website URLs and smileys. Furthermore, TweetTokenizer and stopwords in nltk library are used for tokenizing and removing stopwords.

WordNetLemmatizer in nltk library is also adopted to transform all kinds of words into their roots. These text pre-processing can improve the accuracy of our classifiers.

- Hashing Vectorizer (for Naïve Bayes)

For learning, we could use Count Vectorizer or Tfidf Vectorizer. But the limitation is that the classifier generated only has features for those vocabularies which have appeared. If we want to test it with another dataset, some new vocabularies will not have corresponding features. This issue could be solved by using Hashing Vectorizer, which can generate a classifier with 1,048,576 features for almost all vocabularies. Each tweet is converted into a sample with all the words appearing in the tweet as features. We used hashing vectorizer to perform BoW. Each sample has a dimension of 1×1048576. The data is saved in the form of a sparse matrix.

- Word Embedding (Neural Network)

Stanford pre-trained GloVe for Twitter is used to conduct word embedding. Each word is encoded as a 25-dimensional vector (There are 50-dim, 100-dim, 200-dim available as well). The distance (cosine similarity) between 2 word-vectors reflexes the semantic closeness between 2 words. Each tweet is limited to a length of 140 words. Therefore, the input dimension is 140×25 for each tweet.

2.4 Metrics

The metrics we adopt to evaluate different models are:

precision	recall	F1	accuracy
$\frac{true\ positive}{true\ positive + false\ positive}$	$\frac{true\ positive}{true\ positive + false\ negative}$	$\frac{2 \times precision \times recall}{precision + recall}$	$\frac{true\ positive + true\ negative}{total}$

2.5 Loss function

Naïve Bayes: $-\log p(X, Y)$, where X is word feature, Y is our label.

Neural network: $binary\ cross - entropy\ loss = -(ylog(p) + (1-y)log(1-p))$

$y: \mathbf{1}[correct\ prediction],\ p: prediciton\ probability$

2.6 Input/Output pairs

Input: tweets (text)	"@BreakingNews: Magnitude 7.9 earthquake strikes off the coast of Costa Rica"
Output: relevance & informativeness to event	Related and informative

3. Approach

3.1 Naïve Bayes

We chose multinomial naïve Bayes classifier, which is pervasively used for text classification tasks, in our project. The study can be divided into 2 stages. In the 1st one, we used a new dataset "2013 Boston Bombings" to train and test the classifier and analyzed the factors which could affect its accuracy. Then in the 2nd stage, we used the most accurate form of Naïve Bayes classifier to do the same work as the baseline in order to compare their performance.

The specific procedure and results are explained in the "Results" section.

3.2 Convolutional neural network (CNN)

Although intuitively, CNN is more appropriate to treat spatial data such as images for computer vision, it is prevalently used in natural language processing as well as recurrent neural network (RNN). There are 2 unique layers in CNN architecture: Convolutional layer and pooling layer, which are responsible to reduce the size of data. CNN could capture important local points in a text, which may make it suitable for tweets classification during disasters.

3.3 Long-Short Term Memory (LSTM) neural network

LSTM is a state-of-art type of RNN. There are 4 components in LSTM architecture, including a memory cell (storing values), an input gate (controlling to which extent new value into cell), an output gate (controlling to which extent value in cell is used to compute output) and a forget gate (controlling to which extent new value remains in cell).

3.4 Incremental learning

Usually, there are only a limited number of labeled tweets available for supervised learning. To make use of unlabeled tweets, we propose an incremental learning method: use the model to predict a small portion of the unlabeled data, select these data with high confidence to label them and re-fit our model with the newly-labeled data. Then the loop continues until the end.

Unweighted

```
Divide the dataset of a new event into **k** segments.
    for each segment:
        for each tweet:
            predict its label, get probability **p**
            if the label is over $p_0$:
                label the tweet, add it to the train set
        re-fit the model with these new labeled tweets
```

Weighted

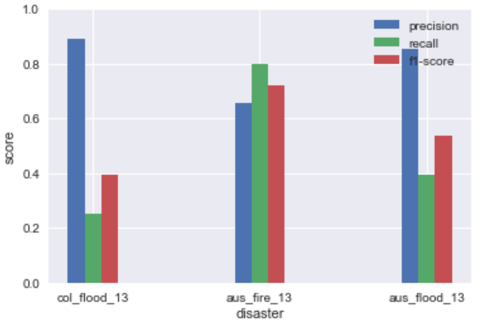
```
Divide the dataset of a new event into **k** segments.
    for each segment:
        for each tweet:
            predict its label, get probability **p**
            label it with weight $\frac{p-0.5}{0.5}$, add to train set
        re-fit the model with these new labeled tweets
```

4. Results

4.1 Baseline

Logistic Regression with hashing vectorizer text preprocessing method is set as the baseline text classifier. The logistic regression model is learned with 2012 Colorado fire dataset and it is applied to 2013 Colorado flood, 2013 Australia fire, 2013 Queensland (Australia) flood respectively to evaluate the accuracy and flexibility of the baseline model.

Event	Precision	Recall	F1-score	Accuracy
2013 Colorado flood	0.8923	0.2518	0.3928	0.4133
2013 Australia fire	0.6580	0.7990	0.7217	0.6562
2013 Australia flood	0.8534	0.3940	0.5391	0.5971



The cross-validation score of the logistic regression model on 2012 Colorado fire dataset is 0.8209. The train score is 0.8848 and the test score is 0.8392 (test size=0.3), which indicates good fitting.

The precision scores of the 2 flood events are high, while the recall score of them are unsatisfactory. This suggests that there are considerable valuable tweets dropped by accident. The model can only distinguish the informative tweets which share the same keywords with the Colorado fire dataset, while it cannot recognize other informative tweets with keywords it does not learn before.

4.2 Naïve Bayes

a. Study the factors which may affect the accuracy

"2013 Boston Bombings" dataset (another dataset from CrisisLexT26) is used for training and testing. Results are also compared between Count Vectorizer and Tfidf Vectorizer.

i. Raw data vs Pre-processed text

		Precision	Recall	F1-score	Accuracy
Raw data	Count Vectorizer	0.8421	0.8348	0.8362	0.8348
	Tfidf Vectorizer	0.8385	0.8391	0.8375	0.8391
Pre-processed	Count Vectorizer	0.8489	0.8478	0.8482	0.8478
	Tfidf Vectorizer	0.8472	0.8478	0.8474	0.8478

Using pre-processed texts is better.

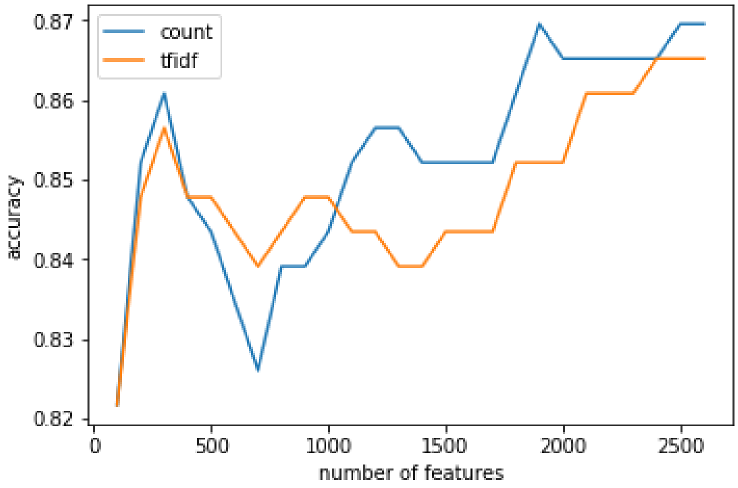
ii. Keep Stopwords vs Remove Stopwords

		Precision	Recall	F1-score	Accuracy
Keep Stopwords	Count Vectorizer	0.8700	0.8696	0.8680	0.8696
	Tfidf Vectorizer	0.8681	0.8652	0.8626	0.8652
Remove Stopwords	Count Vectorizer	0.8489	0.8478	0.8482	0.8478
	Tfidf Vectorizer	0.8472	0.8478	0.8474	0.8478

Keeping stopwords can make the classifier more accurate.

iii. Limit on the maximum number of features

Each appeared vocabulary will become a feature. But we could reduce the number by deleting those features with low probabilities and check how the accuracy will change. The results are shown in the following figure:



From this figure we can find that, generally using more features can provide more accurate results.

Conclusion:

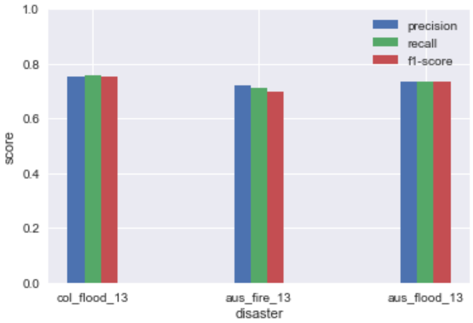
To improve the accuracy, we could use pre-processed texts, keep the stopwords and set no limitation on the maximum number of features.

b. Compared the performance with the baseline

In this part, we used our most accurate Naïve Bayes classifier, and let it complete the same task as the baseline program, namely learning with 2012 Colorado fire dataset and being tested by 2013 Colorado flood, 2013 Australia fire, 2013 Queensland (Australia) flood. Since the baseline used Hashing Vectorizer, we also use it for Naïve Bayes classifier. The results are listed below:

The cross-validation score of the Naïve Bayes model on 2012 Colorado fire dataset is 0.7956. The train score is 0.9725 and the test score is 0.7893.

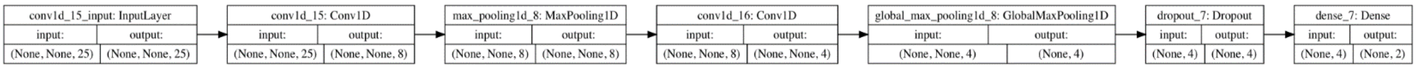
	Precision	Recall	F1-score	Accuracy
2013 Colorado flood	0.7523	0.7579	0.7548	0.7579
2013 Australia fire	0.7195	0.7108	0.6996	0.7108
2013 Australia flood	0.7333	0.7338	0.7335	0.7338



Comparing this with the results of baseline, we can find that the precision decreases slightly, but recall and general accuracy both have remarkable improvement. In other words, compared with baseline, Naïve Bayes model classify slightly more useless Tweets as valuable ones by mistake, but it will not miss many valuable Tweets (baseline performed poorly in this aspect). Moreover, unlike baseline, for Naïve Bayes classifier, the precision, recall and f1-score are quite close to each other, which means Naïve Bayes is more stable and comprehensive than baseline.

4.3 CNN

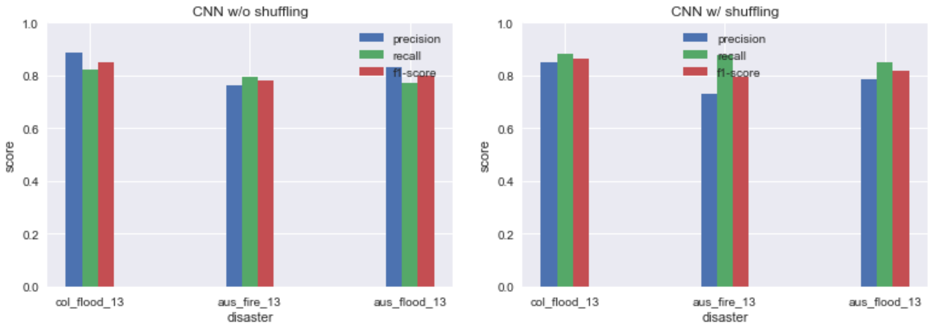
Architecture:



Pros	Cons
Faster both in train and prediction than LSTM. (13s on average to train a CNN model) The output is stable in different events, no heavily affected by disaster type.	Easily overfitting, regularizer and dropout are needed. Convolutional layer hyperparameters need to be fine-tuned to avoid the problem.

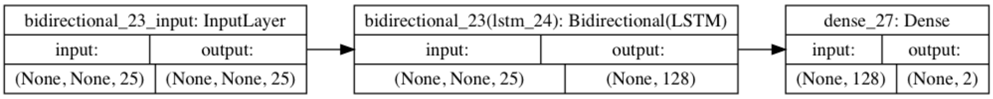
The conventional Keras' "fit" method shuffles the training set but does not resample validation set at each epoch. We wrote a shuffle training function to test the importance of shuffling of validation set. It shows that given the same network architecture, shuffling could increase recall score with little sacrifice in precision, with a slightly better overall performance. Shuffling also decreases the vulnerability of the neural network to overfitting.

Event	Precision		Recall		F1-score		Accuracy	
	w/o s	w/ s	w/o s	w/ s	w/o s	w/ s	w/o s	w/ s
2013 Colorado flood	0.8859	0.8506	0.8205	0.8813	0.8520	0.8657	0.7851	0.7939
2013 Australia fire	0.7648	0.7293	0.7940	0.8771	0.7791	0.7964	0.7488	0.7498
2013 Australia flood	0.8312	0.7839	0.7699	0.8511	0.7994	0.8161	0.7689	0.7707



4.4 Bi-LSTM

Architecture (we also tried out Bi-LSTM followed by a dense layer architecture and a double Bi-LSTM architecture):

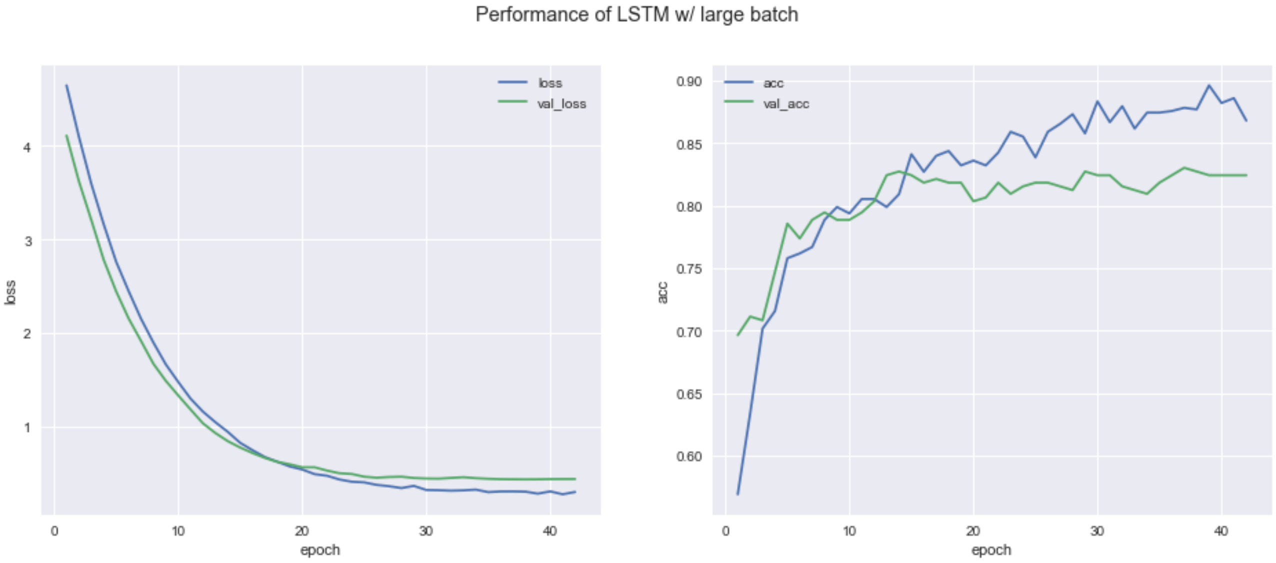


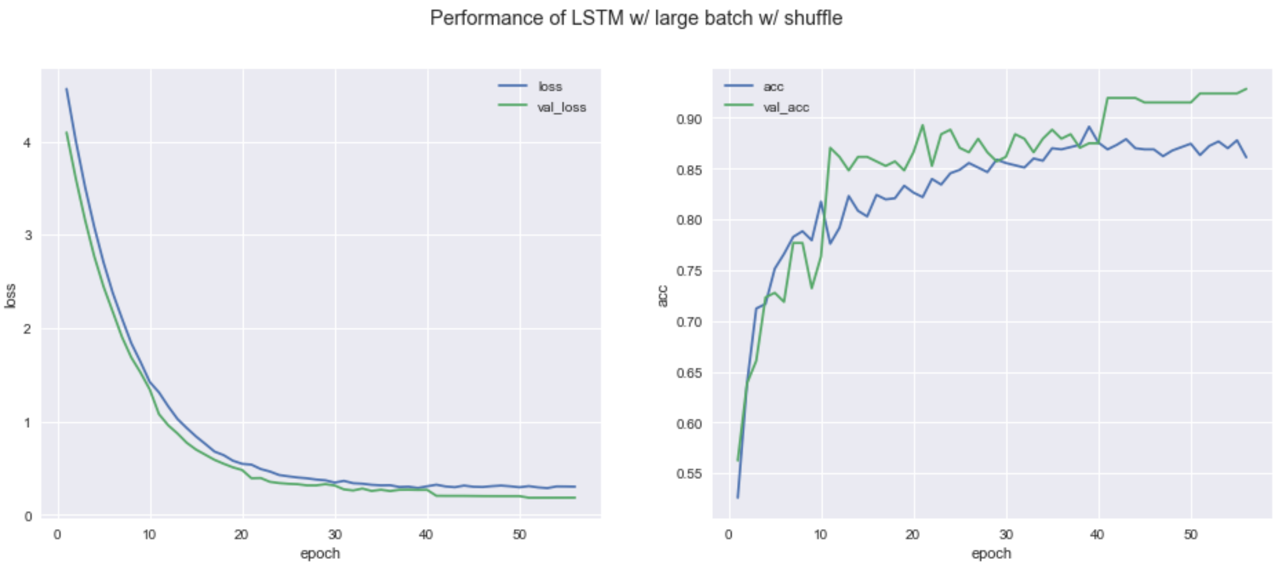
Pros	Cons
1. perform well in different events 2. less sensitive to overfitting compared with CNN 3. highest accuracy among our models.	Training is time-consuming.

Output dim	val_loss	val_acc	train_acc	wall time
4	0.4042	0.8423	0.7951	7min 10s
8	0.3789	0.8571	0.8092	6min 59s
16	0.3895	0.8393	0.8143	5min 52s
32	0.3846	0.8274	0.8335	5min 17s
64	**0.3914**	**0.8720**	**0.8528**	**4min 50s**
128	0.4529	0.8304	0.8937	6min 18s

Output dimension of LSTM layer has a great impact on training time, but not on the validation accuracy. Hence, we decide to choose output dimension=64 (output dimension=128 is overfitting). Furthermore, we decide to remove regularizer because this only cost trivial decrease in scores but can shorten the training time by more than half (wall time is 2min 17s for output dim=64 without regularizer).

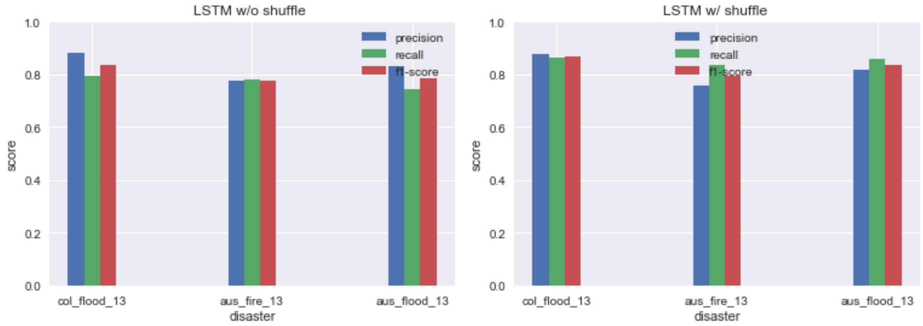
According to Goyal et al. (2017), we could accelerate training of RNN without losing accuracy by increasing the size of mini-batch. The initial learning rate is, as suggested by Goyal et al. (2017), is raised linearly proportionally to batch size. In our experiment, we increased the mini-batch size from 64 to 640, training time shrinks from 5min41s to 2min 20s, with the same accuracy retained.





Again, we compared learning with shuffling and learning without shuffling. The results indicate shuffling is important to improve the performance of the classifier in terms of higher accuracy and being overfitting-averse.

Event	Precision		Recall		F1-score		Accuracy	
	w/o s	w/ s	w/o s	w/ s	w/o s	w/ s	w/o s	w/ s
2013 Colorado flood	0.8841	0.8778	0.7945	0.8625	0.8369	0.8701	0.7666	0.8059
2013 Australia fire	0.7747	0.7602	0.7824	0.8372	0.7785	0.7968	0.7516	0.7618
2013 Australia flood	0.8316	0.8178	0.7428	0.8571	0.7847	0.8370	0.7562	0.8004



4.5 Incremental learning

Naïve Bayes

We applied deterministic incremental learning using Naïve Bayes model. The results of 3 test datasets are listed below:

Event	Accuracy	
	before	after
2013 Colorado flood	0.7579	0.8048
2013 Australia fire	0.7108	0.7637
2013 Australia flood	0.7338	0.7905

From these results, we discover that incremental learning can improve the accuracy remarkably. However, during experimentation we also found that the recall of “Related and informative” class is very high, but the recall of “Not related or not informative” class is quite low. This means to say, under incremental learning our classifier tends to classify more and more tweets as the “Related and informative” ones, which does not seem good. The reason may be that, sometimes it is quite difficult to differentiate “Related and informative” tweets from “Related but not informative” tweets. They could be ambiguous even for a human (this will be discussed in detail in "error analysis" section). This may affect the behaviors of incremental learning.

Neural Network

CNN

Event	Accuracy		
	before	unweighted	weighted
2013 Colorado flood	0.7601	0.7688	0.7699
2013 Australia fire	0.7201	0.7238	0.7201
2013 Australia flood	0.7338	0.7517	0.7437

LSTM

Event	Accuracy		
	before	unweighted	weighted
2013 Colorado flood	0.8059	0.8168	0.8190
2013 Australia fire	0.7618	0.7702	0.7692
2013 Australia flood	0.8004	0.8022	0.8076

The incremental learning results are unstable for the neural network, Occasionally the accuracy of our models would drop, especially for the Australia bushfire event (accuracy drops significantly for this sole event, for example, from 74% to 69% or even lower). We then checked the datasets and found out some conflicts of tweets labels which we will discuss later in the Error Analysis section of this report.

Generally speaking, data shuffle (re-generate validation set every epoch or every several epochs) and regularization are important to guarantee a successful incremental learning.

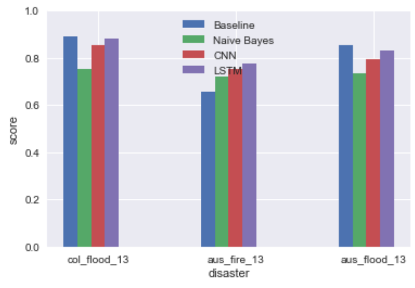
5. Analysis

5.1 Comparison between different methods

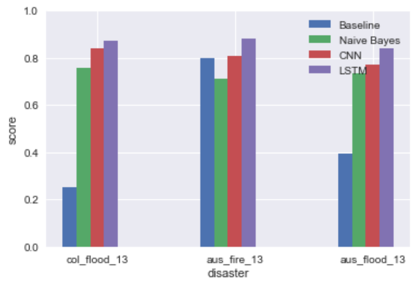
We plotted the precisions, recalls and F1-scores for different methods respectively, as shown in the 3 figures below.

Considering the performance of baseline, we noticed that it has amazing precisions, while its recall and F1-score are not as good. Comparing the results of our proposed models with the baseline, we could find that sometimes the precisions of our proposed methods are slightly lower than baseline. However, in terms of recall and F1-score, our proposed methods are much better than the baseline. Therefore, our proposed methods are generally more accurate and more stable than our baseline in Tweets classifications.

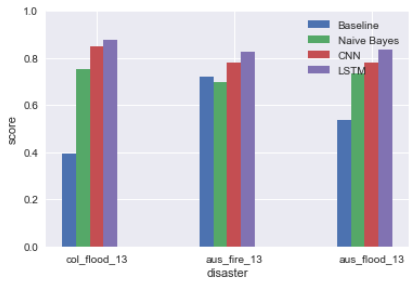
Precision



Recall



F1-score



Among those 3 methods we proposed (Naïve Bayes, CNN and LSTM), it is quite obvious that LSTM has the best performance in terms of accuracy. CNN is worse than LSTM but better than Naïve Bayes. However, if we consider training time, Naïve Bayes is the fastest, CNN is in the next place, and LSTM is the slowest one.

Besides, compared with baseline which is more sensitive to disaster type, our 3 models trained with Colorado wildfire event seem to perform slightly better in the Colorado flood event, which indicates the influence of geolocation of datasets.

5.2 Error Analysis

Baseline:

The results show that the model learned from Colorado fire has a relatively better performance to filter informative tweets in Australia fire and the behavior of baseline model is unsatisfactory to predict the labels of tweets for the 2 flood events. This intuitively makes sense: 2 disasters of the same type share more keywords (such as 'fire', 'burn', 'scorch', 'flame', etc. in fire events, which generally do not show up during flood events). After scrutinizing the classified tweets for the 2 flood events, we found that in the true positive tweets, many include words universal to all the types of disasters, such as 'death', 'destroy', 'damage', 'kill', etc. However, in the false negative tweets, many do not include the common terms.

Our models:

Our models are stable and balanced when predicting labels of the other 3 datasets. We checked those mismatched results and tried to figure out why our models made a mistake.

For false negative tweets, most of them are about donations and volunteering. In these sentences, the tones and vocabularies used are generally quite different from other tweets (e.g., *'RT @DenverChannel: $46,468 raised for #COFloodRelief so far! Call 877-667-6727 to donate and Please RT! http://t.co/Ba2o6QkYLN'*). That might be the reason why our models cannot identify them.

We also found some ambiguous tweets, which is an unavoidable phenomenon when crowdsource workers manually labeled these tweets. In some circumstances, it is indeed difficult to draw a clear line between 'Related and informative' and 'Related but not informative'. For example,

Smoky sky. #Sydney #Bushfire http://... (negative, from Australia bushfire 2013 dataset)
Amazing time lapse footage of Colorado Springs fire. http... (positive, from Colorado wildfire 2012 dataset)
The 2 tweets above all described some scenes of fire. However, the Australia one is labeled as negative and the Colorado one is positive on the contrary.

Kum &amp; Go Donates $15,000 to American Red Cross; Helps communities affected by Colorado flooding. Read more:... http://... (negative, from Colorado flood 2013 dataset)
This tweet is labeled as negative but it did provide info about donation.

The Devastating Floods In Colorado And How You Can Help http://t.co/UsVMBluP6X (negative, from Colorado flood 2013 dataset)
RT @benloiacono: Cigarette butts cause #bushfires. Please bin your butt! #hillsdistrict #SydneyHills #NSWRFS (positive, from Australia bushfire 2013 dataset)
The 2 tweets above all described some advice related to disasters, however, the Australia one is labeled as positive and the Colorado one is negative on the contrary.

6. Conclusion

In this project, we explored three different methods (Naïve Bayes, CNN and LSTM) to accomplish the Tweets classifications. Compared with the baseline (logistic regression), they all achieved some improvements. The baseline has really high precisions, but its recall and F1-score are not as good. Our 3 methods have higher recall and F1-score, which means they are more stable and generally more accurate.

Among our 3 methods, LSTM has the best performance in terms of accuracy, but it takes more time to train than the other two. Naïve Bayes has the lowest accuracy, but it is the fastest among all 3 models.

Future Work:

Unfortunately, we did not have enough time to carry out many experiments to perform more rigorous statistical analysis to show which method fits incremental learning better, which parameters of our models and incremental learning methods have great influences on the performance of incremental learning, which incremental learning method is better than the other, and more basically, to explore more sophisticated methods to perform incremental learning, or semi-supervised learning to deal with rare labeled text data. Some methods, such as expectation maximization for Naïve Bayes classifier, label spreading semi-supervised learning, etc., could be a potential research topic for whoever is interested in this field.

7. References

A. Olteanu, S. Vieweg, C. Castillo. 2015. What to Expect When the Unexpected Happens: Social Media Communications Across Crises. In Proceedings of the ACM 2015 Conference on Computer Supported Cooperative Work and Social Computing (CSCW '15). ACM, Vancouver, BC, Canada.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

Leykin, Dmitry, Mooli Lahad, and Limor Aharonson-Daniel. “Gauging Urban Resilience from Social Media.” International Journal of Disaster Risk Reduction, April 2018. https://doi.org/10.1016/j.ijdrr.2018.04.021.

Li, Hongmin, Doina Caragea, Cornelia Caragea, and Nic Herndon. “Disaster Response Aided by Tweet Classification with a Domain Adaptation Approach.” Journal of Contingencies and Crisis Management 26, no. 1 (March 2018): 16–27. https://doi.org/10.1111/1468-5973.12194.

Priya Goyal, Piotr Dollar, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, Kaiming He. 2017. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour.

Stielitz, Stefan, Deborah Bunker, Milad Mirbabaie, and Christian Ehnis. 2018. “Sense-Making in Social Media during Extreme Events.” Journal of Contingencies and Crisis Management 26(1):4–15. Retrieved April 17, 2018 (http://doi.wiley.com/10.1111/1468-5973.12193)

Zahra, Kiran, and Ross Purves. “Analysing Tweets Describing during Natural Disasters in Europe and Asia,” 2017, 6.