# CS 221 Project Progress Report

Guoqin Ma, HAOTIAN LI

TOTAL POINTS

**10 / 10**

QUESTION 1

## 1 Progress Report 10 / 10

✓ + **10 pts** **Click here to replace this description.**

+ **9 pts** Click here to replace this description.

+ **8 pts** Click here to replace this description.

+ **7 pts** Click here to replace this description.

+ **0 pts** Click here to replace this description.

Great job! The technical implementation is very strong. However, the task and dataset should be explained more in the final submission and poster. How large are your datasets? What motivated your decision to train on one dataset, but test on 3? And who labeled your dataset for relevance and informativeness? It would be good to see statistical analysis on the balance on your dataset as well. Keep up the experimentation and error analysis!

# CS 221 Project Progress 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. Motivation

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

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 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.

| 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

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.

Data pre-processing

- Modification

We removed those Tweets which were released by the government. In our projects, tweets labeled "related and informative" are positive tags, which are the desired tweets useful for disaster decision making.

- Text pre-processing

Remove emojis, website URLs and smileys, stopwords. Perform lemmatization to transform all kinds of words into their roots.

- Hashing Tricks

Each tweet is converted into a sample with all the words appearing in the tweet as features.

- Word Embedding

Stanford pre-trained GloVe for twitter is used to conduct word embedding. 25 is chosen as the dimension of word vectors.

Metrics

The metrics we adopt to evaluate different models are:

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

Loss function

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

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

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

Naïve Bayes

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.

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.

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).

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.

Deterministic

```
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
```

Probabilistic

```
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
```

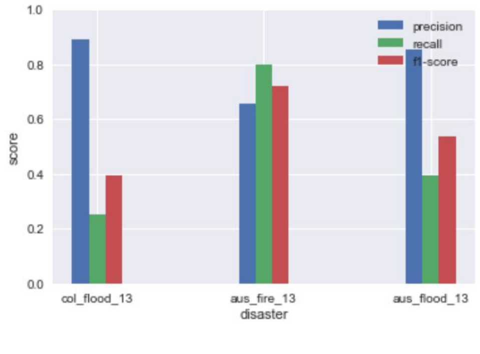
## 4. Results

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.

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.

| 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 results show that the model learned from Colorado fire has relatively better performance to filter informative tweets in Australia fire than the other 2 flood events, which 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).

Furthermore, 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.

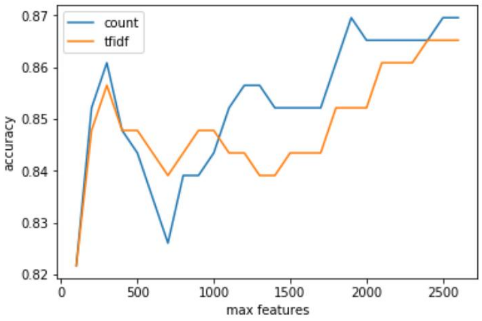
Naïve Bayes

a. *Study the factors which may affect the accuracy.*

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

We first compared "raw data" vs "pre-processed text". Secondly, we compared "keep stopwords" vs "remove stopwords". Due to the limitation of space, the specific results such as precision, recall and accuracy are not presented here. The conclusion is using pre-processed texts and keeping stopwords can make the classifier more accurate.

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



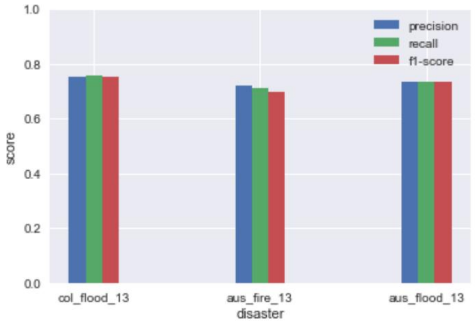
From this figure, we can find that generally using more features can provide more accurate results.

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 |



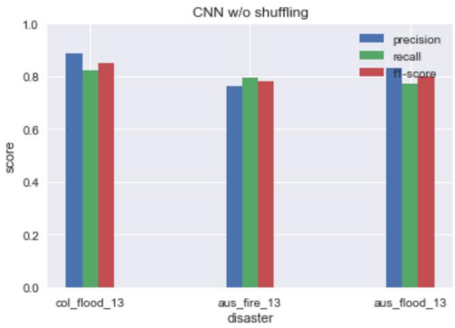
Comparing this with the results of baseline, we can find that the precision decreases slightly, but recall and general accuracy both have remarkable improvement. In other words, compared with baseline, Naïve Bayes model classify slightly more useless Tweets as valuable ones by mistake, but it will not 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.

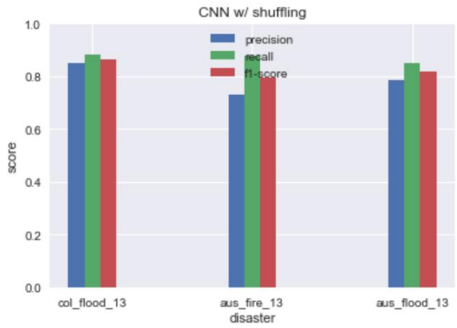
CNN

| Pros | Cons |
|---|---|
| Faster both in train and prediction than LSTM.<br>The output is stable in different events, no heavily affected by disaster type. | Easily overfitting, regularizer is needed. |

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





Bi-LSTM

Pros: Perform well in different events, good incremental learning results, less sensitive to overfitting.

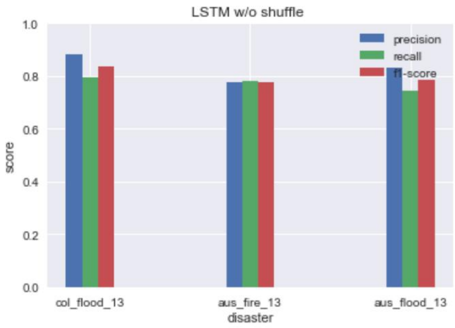
Cons: Time is a great concern. Run-time heavily depends on dropout and regularizer.

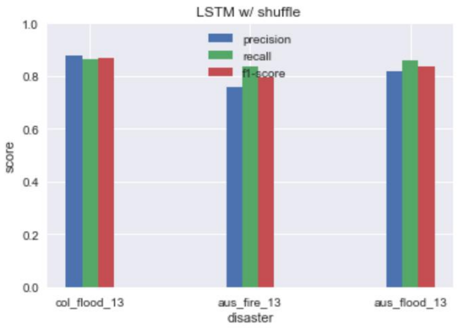
| LSTM | val_loss | val_acc | train_acc | wall time |
|---|---|---|---|---|
| Output dim | | | | |
| 4 | 0.4042 | 0.8423 | 0.7951 | 7min 10s |
| 8 | 0.3789 | 0.8571 | 0.8092 | 6min 59s |
| 16 | 0.3895 | 0.8393 | 0.8143 | 5min 52s |
| 32 | 0.3846 | 0.8274 | 0.8335 | 5min 17s |
| **64** | **0.3914** | **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).

Again, compare learning with shuffling and learning without shuffling. The results indicate shuffling is important to improve the performance of the classifier.

| 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 |





## 5. Plan

a. Repeat test to prevent contingency.

b. Keep parameter tuning for our models.

c. One of the main issues of this project is that, in practice, when a new event occurs, the labeled data are extremely limited and very difficult to obtain. In this situation, we may want to use the labeled data of other previous events to train our classifier and used it to classify new Tweets about this current event. This will involve domain adaption. So, we plan to do more research on this topic and provide some feasible algorithms.

## 6. 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.

Stieglitz, 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.

## 1 Progress Report 10 / 10

✓ **+ 10 pts** **Click here to replace this description.**

+ **9 pts** Click here to replace this description.

+ **8 pts** Click here to replace this description.

+ **7 pts** Click here to replace this description.

+ **0 pts** Click here to replace this description.

Great job! The technical implementation is very strong. However, the task and dataset should be explained more in the final submission and poster. How large are your datasets? What motivated your decision to train on one dataset, but test on 3? And who labeled your dataset for relevance and informativeness? It would be good to see statistical analysis on the balance on your dataset as well. Keep up the experimentation and error analysis!