Natural Language Processing With Disaster Tweets

Yoonjung Choi, Abhiteja Mandava, Ishaan Bhalla, Sakruthi Avirineni
CMPE 255 Data Mining
San Jose State University

***Abstract* – This paper is to predict a disaster based on a data set collecting text from twitter. By analyzing tweets and building a model, we can utilize this study for companies to track, monitor and predict disasters from the real time data and this study would help making prediction model. This problem is categorized by a supervised learning, binary classification and natural language processing. For data preprocessing, we cleaned text from unnecessary text such as URL, Emojis or HTML tags and normalized text by using useful algorithms; tokenizer, stopwords and lemmatization. We created four numerical feature data sets by using Count Vectorizer, Inverse Document Frequency, Word2Vec and Word2Vec with PCA applied, and trained each numerical feature sets on different models; Decision Tree, Support Vector Machine, Logistic Regression, and Ensemble Model. We found each combination how each model has the higher performance on which feature set. Then, for fine-tuning, we modified parameters of models to increase accuracy.**

***Keywords* – NLP, PCA, Ensemble**

I. Introduction

Social Network Service has been playing a crucial role in communication and Natural Language Processing has been widely used to analyze it and extract potential patterns. Twitter is one of the popular SNS platforms and many tweets has been delivered in emergency situation. Since there are demands for companies to utilize this tweets, we investigated and developed natural language processes and prediction models to have the best performance.

A. Data set

The data set has been collected from the company figure-eight and originally shared on their 'Data For Everyone' website [1]. We found the data set from Kaggle Competition [2]. It contains 7613 samples with the following features:

Feature	Dtype	Descrition
id	int64	
keyword	object	39 non-values
location.	object	2533 non-valus
text	object	tweetter content
target	int64	0:non-disaster, 1:disaster

Fig. 1. The table shows feature, type, and description.

The each figure shows the count of top 15 keywords of each target.

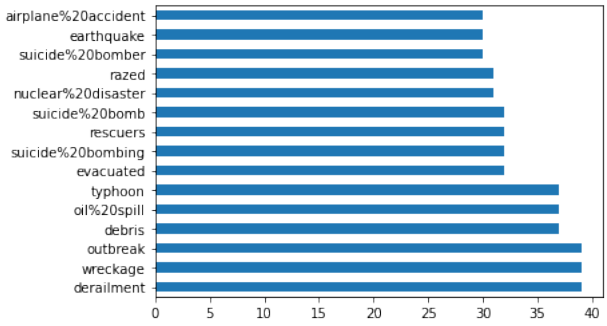


Fig. 2. Top 15 of disaster tweets' keywords

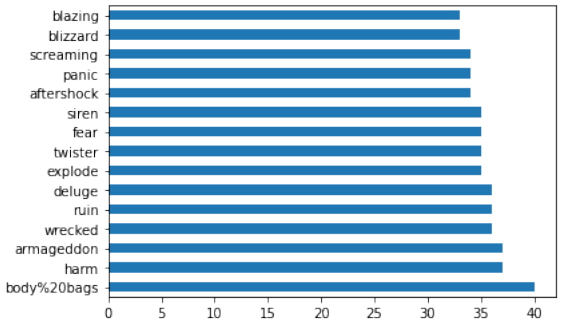


Fig. 3. Top 15 of non-disaster tweets' keywords

II. Data Exploration

We use only two features 'text' and 'target'.The figure shows the percentage of feature 'target"s distribution.

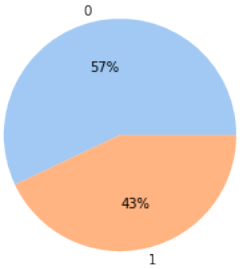


Fig. 4. there are 4342 samples for target(0) and 3271 samples for target(1)

In the Natural Language Processing, there are several common processing steps and many practical algorithms. We need to clean data, normalize data, create numerical feature vector.

A. Data Cleaning

We should remove unnecessary words to make sure it has meaningful values. For cleaning text, we have changed all words to lowercase, removed URL, HTML tags, Emojis, punctuation and ASCII codes.

text
Crying out for more! Set me ablaze
On plus side LOOK AT THE SKY LAST NIGHT IT WAS ABLAZE http://t.co/qqsmshaJ3N
@PhDSquares #mufc they've built so much hype around new acquisitions but I doubt they will set the EPL ablaze this season.
INEC Office in Abia Set Ablaze - http://t.co/3lmaomknnA
Barbados #Bridgetown JAMAICA ‰ÛÒ Two cars set ablaze: SANTA CRUZ ‰ÛÓ Head of the St Elizabeth Police Superintende... http://t.co/wDUEaj8Q4J

Fig. 5. The table shows partial of text

CleanText
crying out for more set me ablaze
on plus side look at the sky last night it was ablaze
phdsquares mufc theyve built so much hype around new acquisitions but i doubt they will set the epl ablaze this season
inec office in abia set ablaze
barbados bridgetown jamaica two cars set ablaze santa cruz head of the st elizabeth police superintende

Fig. 6. The table shows partial of cleaned text

B. Data Preprocessing

Now, we have a cleaned text set and we should apply some methods to normalize words. NLTK [3] provides easy-to use interfaces for natural language processing

1) Tokenizer: Tokenizers divide strings into lists of substrings. We can use 'tokenize' library to find the words and punctuation in a sentences.

TokenizedText
['crying', 'out', 'for', 'more', 'set', 'me', 'ablaze']
['on', 'plus', 'side', 'look', 'at', 'the', 'sky', 'last', 'night', 'it', 'was', 'ablaze']
['phdsquares', 'mufc', 'theyve', 'built', 'so', 'much', 'hype', 'around', 'new', 'acquisitions', 'but', 'i', 'doubt', 'they', 'will', 'set', 'the', 'epl', 'ablaze', 'this', 'season']
['inec', 'office', 'in', 'abia', 'set', 'ablaze']
['barbados', 'bridgetown', 'jamaica', 'two', 'cars', 'set', 'ablaze', 'santa', 'cruz', 'head', 'of', 'the', 'st', 'elizabeth', 'police', 'superintende']

2) Stopwords: We should remove commonly used words (such as "the", "a", "is", "in").

RemoveStopWords
['crying', 'set', 'ablaze']
['plus', 'side', 'look', 'sky', 'last', 'night', 'ablaze']
['phdsquares', 'mufc', 'theyve', 'built', 'much', 'hype', 'around', 'new', 'acquisitions', 'doubt', 'set', 'epl', 'ablaze', 'season']
['inec', 'office', 'abia', 'set', 'ablaze']
['barbados', 'bridgetown', 'jamaica', 'two', 'cars', 'set', 'ablaze', 'santa', 'cruz', 'head', 'st', 'elizabeth', 'police', 'superintende']

3) Stemming: Stemming is the process of producing morphological variants of a root/base word. For example, words such as "Likes", "liked", "likely" and "liking" will be reduced to "like" after stemming. There are different algorithms for stemming. Porter Stemmer is one of them and is a basic stemmer. Its' advantages are straightforward and fast to run.

PorterStemmer
['cri', 'set', 'ablaz']
['plu', 'side', 'look', 'sky', 'last', 'night', 'ablaz']
['phdsquar', 'mufc', 'theyv', 'built', 'much', 'hype', 'around', 'new', 'acquisit', 'doubt', 'set', 'epl', 'ablaz', 'season']
['inec', 'offic', 'abia', 'set', 'ablaz']
['barbado', 'bridgetown', 'jamaica', 'two', 'car', 'set', 'ablaz', 'santa', 'cruz', 'head', 'st', 'elizabeth', 'polic', 'superintend']

4) Lemmatization: Lemmatization is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. Both stemming and lemmatization are word normalization techniques, but we can find the word in dictionary in case of lemmatization. For example, original words 'populated' changed 'popul' in Stemming, but it is not changed in lemmatization. Lemmatization is more better performed than Stemming [4]. We decided to apply lemmatization.

LemmatizedText
['cry', 'set', 'ablaze']
['plus', 'side', 'look', 'sky', 'last', 'night', 'ablaze']
['phdsquares', 'mufc', 'theyve', 'built', 'much', 'hype', 'around', 'new', 'acquisition', 'doubt', 'set', 'epl', 'ablaze', 'season']
['inec', 'office', 'abia', 'set', 'ablaze']
['barbados', 'bridgetown', 'jamaica', 'two', 'car', 'set', 'ablaze', 'santa', 'cruz', 'head', 'st', 'elizabeth', 'police', 'superintende']

5) Data Visualization: After normalized text, we made data visualization by using word cloud. In disaster tweet's words, we can discover disaster related words; suicide, police, news, kill, attack, death, california, storm, flood. In the other hand, the non disaster tweets shows that time, want, great, feel, read, but also killed, injury or emergency are found.

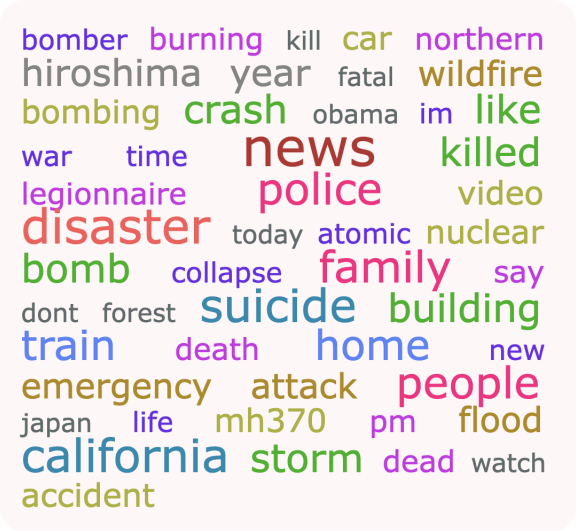


Fig. 7. target(1) disaster tweets' words

Fig. 8. target(0) non disaster tweets' words

The Figure 7 represent histogram of the number of words at each sample.

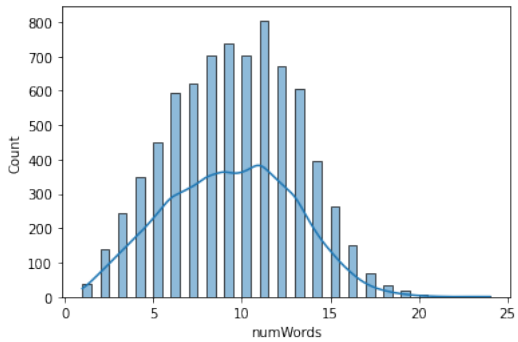


Fig. 9. histogram of lengths of tweets' words

C. Transforming numerical feature vector

Bag of Words model is a simplified representation used in natural language processing. A text is represented as the bag of its words, disregarding grammar and describes the occurrences of words with in a document.

1) CountVectorizer: CountVectorizer can be used for bag of words model. This convert a collection of text documents to a matrix of token counts.

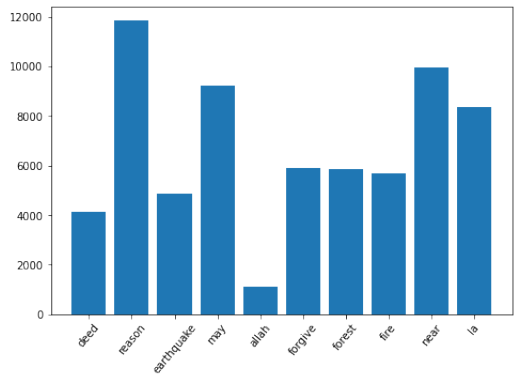


Fig. 10. Bar chart shows the number of ten words from dictionary

2) TF-IDF: The Term Frequency Inverse Document Frequency is a measure of whether a term is common or rare. It gives weight more to a term that occurs in only a few documents.

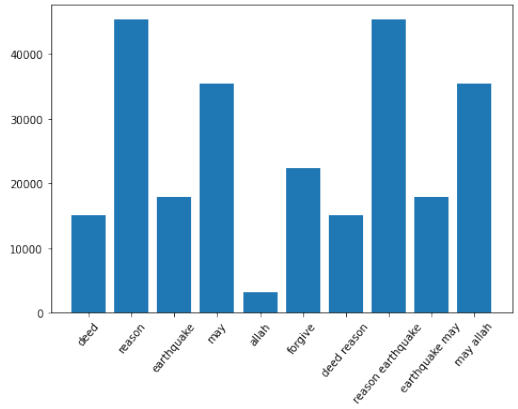


Fig. 11. Bar chart shows the number of ten words from dictionary

3) Word2Vec: Word2Vec uses a neural network model to learn word associations from a large corpus of text [5]. Word2Vec represents words in vector space in a way of similar meaning words are positioned in close locations but dissimilar words are placed far away.

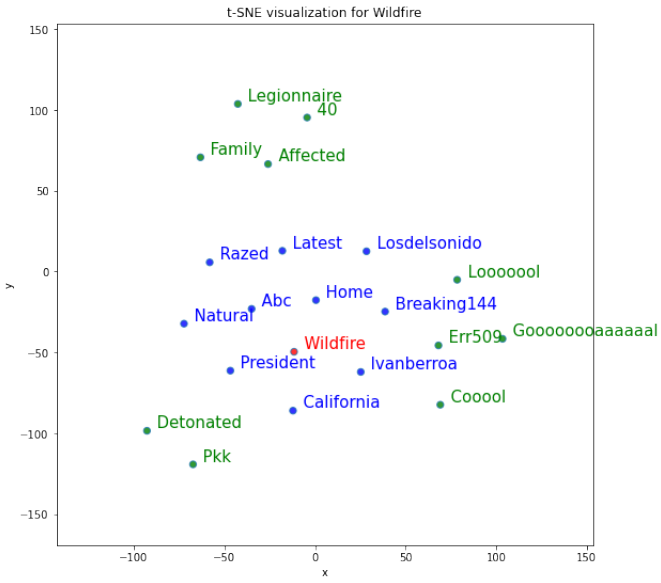
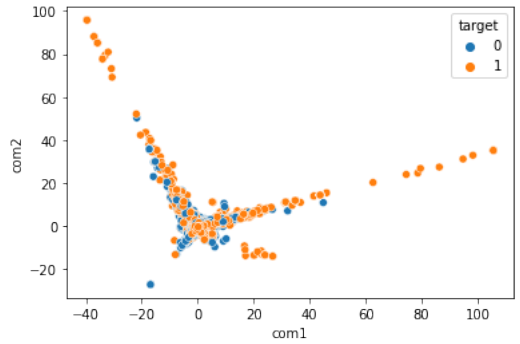


Fig. 12. For word 'wildfire', blue words represent most similar words and green words represent most dissimilar words

4) Word2Vec with PCA applied: As principal component analysis is a strategy to reduce dimension, we applied PCA with 100 components on feature set from Word2Vec. The below figure is shown when applying PCA with 2 components.



III. METHODS

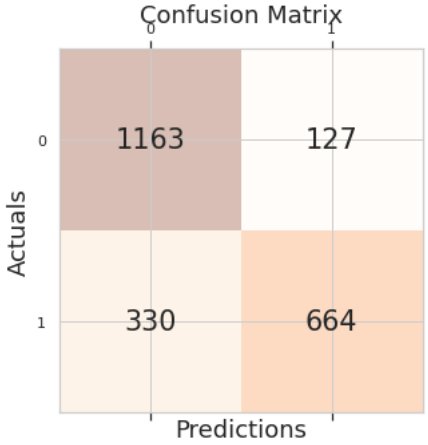
Our workflow is several steps. We trained each numerical feature sets on the basic model to find which feature set yields better accuracy. For fine-tuning, we adjusted parameters on the model with the selected feature. We repeated the same steps on other models. We also trained each features with ensemble method.

1) SVM: Support Vector Machine is a supervised learning model used for classification and regression problems. We trained each numerical feature sets on basic SVM, which means no changes of parameters. In case of SVM, Count Vector feature set has higher accuracy and f1 score than other feature sets.

SVM	Count Vector	Tf-Idf	W2V	W2V + PCA
accuracy	0.799	0.761	0.624	0.709
Recall	0.639	0.493	0.163	1.434
Precision	0.864	0.923	0.857	0.809
F1 Score	0.735	0.643	0.274	0.565

We adjusted parameters to yield best accuracy. In the final SVM model, it has default C value as 1, gamma value as 'auto', kernel value as 'sigmoid'. We obtained the result and confusion Matrix of the model.

SVM	Accuracy	Recall	Precision	F1
score	0.800	0.668	0.839	0.744

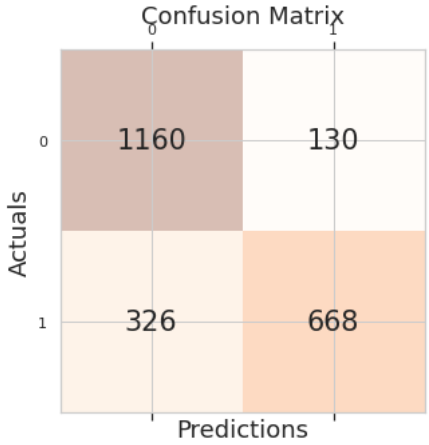


2) Logistic Regression: Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. It is also a supervised learning model used for classification problems. We trained each feature sets on basic Logistic Regression without fine-tuning, and Count Vector has higher accuracy as well.

LR	Count Vector	Tf-Idf	W2V	W2V + PCA
accuracy	0.797	0.776	0.669	0.751
Recall	0.693	0.539	0.314	0.603
Precision	0.813	0.903	0.806	0.776
F1 Score	0.749	0.677	0.452	0.678

From the fine-tuning, we finalized parameters as C=0.15, penalty='l2', tol=0.001, solver='saga', random state=42, max iter=1000. We obtained the result and confusion matrix.

LR	Accuracy	Recall	Precision	F1
score	0.800	0.672	0.837	0.746

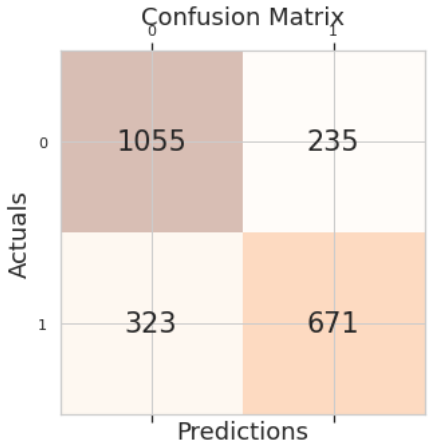


3) Decision Tree: Decision Tree is a non-parametric supervised learning method used for classification and regression problems. We trained each numerical feature sets on basic decision tree, and Tf-Idf feature has a little bit higher accuracy rather than others.

DT	Count Vector	Tf-Idf	W2V	W2V + PCA
accuracy	0.749	0.752	0.655	0.683
Recall	0.671	0.684	0.616	0.621
Precision	0.731	0.730	0.614	0.640
F1 Score	0.700	0.706	0.615	0.630

From the fine-tuning, we finalized parameters as min samples split=8. We obtained the result and confusion matrix.

DT	Accuracy	Recall	Precision	F1
score	0.756	0.675	0.741	0.706



4) Ensemble: Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would. We trained each features on ensemble model consisted of SVM, Logistic Regression and Decision Tree.

5) Results:

IV. CONCLUSION

REFERENCES

[1] OPPEN, URL: https://appen.com/datasets-resource-center.

[2] Kaggle, "Natural Language Processing with Disaster Tweets", URL: https://www.kaggle.com/competitions/ nlp-getting-started/data.

[3] NLTK, "Natural Language Toolkit", URL: https://www.nltk.org/index.html.

[4] Baeldung, "Naturalstemming-vs-lemmatization", URL: https://www.baeldung.com/cs/stemming-vs-lemmatization.

[5] Wikipedia, "word2vec", URL: https://en.wikipedia.org/wiki/Word2vec.

[1]–[5].