Coursework

Pornphapat Innwon 31378218

January 2019

1 Introduction

Social media is the platform that people usually spread their opinions or information because it is the easiest and fastest approach to let a large number of people see what they want to express their ideas. In perspective of the news report, the journalists use this platform to identify new stories as they often can not be in the story situation. The major advantage of social media for the news report is that people like to spread the news. In this platform, people can spread a story by just clicking a few buttons. However, people also tend to spread fake news. The reason for this is they do not know the reliable of the news. Whether the writer deceives people accidentally or with intention, this gives more works to the journalists to verify the stories. With the improvement of technology, people can use machine learning algorithms as a way to classify data automatically. Thus, this paper will compare five selected machine learning algorithm, discuss the advantages and disadvantages and justify which algorithm is the best.

2 Data Characterization

Regarding the data, the work uses twitter posts in the MediaEval 2015 datasets. According to the MediaEval 2015 paper (1), the purpose of this data is to build a model that can classify news source in twitter automatically. In this use case, this work will design machine learning models that can classify a fake or real post. The work here will aim for the high precision models. The data format is in tab-separated value. The list of columns inside the datasets is shown below.

- `tweetId:` Post id in twitter
- `tweetText:` Post text message
- `userId:` User id in twitter
- `imageId(s):` Categorized image id
- `timestamp:` Time and date that post is created
- `label:` The classification of the post

For the data volume, the number of tweets in the training set and the testing test is 14277 and 3755 posts respectively. About the detail, the data contains some major event posts such as Hurricane Sandy and Boston Marathon Bombing.

Regarding the data quality, the posts have mixture of languages such as English, Spanish and Chinese that can cause diversity in the feature selection. The data also need some cleaning as some posts have some bad grammar, typos and some noises such as expressive special symbols and twitter shorten URL.

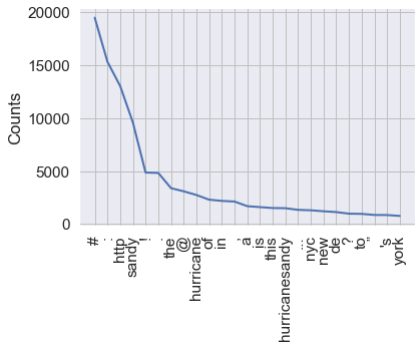
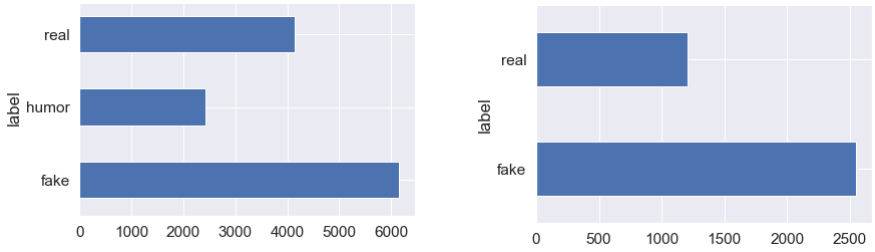


Figure 1: Count of words in the training set

The figure above shows that most of common words has no value to classify the posts and should be removed.

For the data bias, there are some retweets which are duplicates of the original post. These duplicates should be removed. In addition, most of the posts are fake as illustrated in these graphs. Note that, the humour posts are considered as fake posts in the evaluation.



(a) Training set

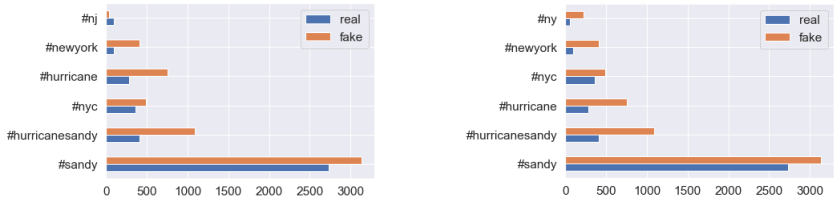
(b) Testing set

Figure 2: Data volume

The figure explains that fake tweets are more common than real tweets. In other words, the models tend to classify posts as fake news due to the unbalance of the data. This can reduce the performance of identifying real news.

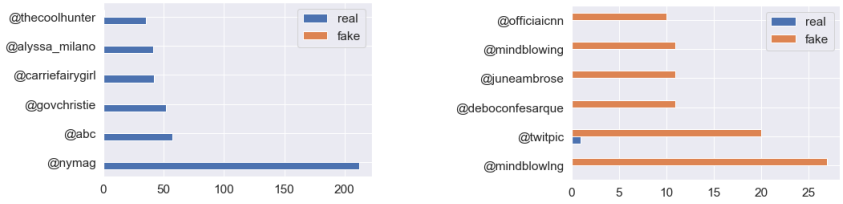
3 Data analysis

In twitter, there are mention and hashtag features which are convenient to the users. This datasets also have these two features. These can be use to as a clue to find the good features for the model. Note that, the humor posts are now identified as fake posts.



(a) sorted by real posts (b) sorted by fake posts

Figure 3: Number of hashtags



(a) sorted by real posts (b) sorted by fake posts

Figure 4: Number of mentions

While there are very few distinctions in the hashtag figure, the two bottom graph clearly shows the difference mentions are used in each type of posts. The real posts tend to mention reliable news twitter @nymag. On the other hand, the fake posts like to use a mention to over express the post such as @mindblowing. These terms can be used to separate two kinds of posts. This suggests that the contents inside the post are important features in classifying the tweets.

4 Algorithm Design

Before building anything, the pre-processing must be done first. The steps are described below.

- `Retweet Removal` Retweets can cause bias to the machine learning model.
- `Normalization` This includes lowercase,removing punctuations and removing URLs.
- `Lemmatization` This method is to reduce words into dictionary form. It uses the POS tagging or 'part of speech tagging' to accurately determine the dictionary form.
- `Stop word` Stop words are words that are common in the sentence such as 'a', 'in' and 'with'. This should be removed.
- `Noise Removal` Removing some symbols such as
  n from the tweet.

For all design, the model uses a tf-idf approach to find important features in the tweetText column. The tf-idf or 'term frequency-inverse document frequency' is a scoring technique in a bag of word feature that gives the importance of words that can inform the difference in each document. For instance, the most common words have low tf-idf value because it occurs almost everywhere in the documents. By doing a tf-idf method, this create about 12000 features. The reason for choosing this approach is the great performance in term of word frequency according to zhang paper (2).

For each algorithm designs, the details are listed below. Note that, there is no dimension reduction implemented because, from the trial and error, it makes accuracy worse. In addition, the gridsearchCV is used to find the optimal solution.

- `Multinomial Naive Bayes` The algorithm based on the Bayes' Theorem. It uses probability to find relation of the data. From the Rish's paper (3), the algorithm often works effectively in classification.
  - `feature selection` select top 3000 tf
- `Random Forest algorithm` Decision tree ensemble method for both classification and regression. The algorithm works great in high dimension problem.
  - `feature selection` select top 8500 tf
- `K-neighbour algorithm` The algorithm can be used to find similarity of the word vectors according to Trstenjak's paper (4)
  - `feature selection` select top 12000 tf

- `Bagging Classifier` One of the decision tree ensemble method. It can help the case when the model is overfitted.
  - `feature selection` select top 12000 tf
- `Logistic Regression` The reason is probability strategy which similar to Naive Bayes algorithm.
  - `feature selection` select top 12000 tf

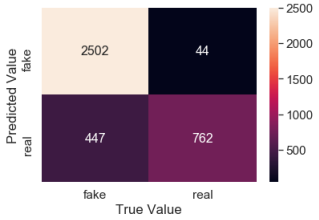
5 Evaluation

(a) Multinomial Naive Bayes

(b) Random Forest algorithm

(a) K-neighbour algorithm

(b) Bagging Classifier



Logistic Regression

Figure 5: Confusion matrix

Model	Accuracy	Recall	Precision	F1-score
Multinomial Naive Bayes	**0.89**	**0.85**	0.89	**0.86**
Random Forest algorithm	0.81	0.78	0.79	0.79
K-nearest Neighbors algorithm	0.87	0.81	0.89	0.83
Bagging Classifier	0.85	0.80	0.85	0.81
Logistic Regression	0.87	0.81	**0.90**	0.83

the figure and the table shows the statistic of the each algorithms. From the result, it seems that Multinomial Naive Bayes have the highest performance in every aspect except the precision. From the experiment, List of strength and weakness of each algorithm is explained below.

- `Multinomial Naive Bayes`
  - `Strength`
    - have probabilistic prediction
    - data scalable
    - easy to implement
  - `Weakness`
    - based on bad relation assumption
    - need large data
    - can lose accuracy
- `Random Forest algorithm`
  - `Strength`
    - reduce over-fitted data
    - can be used in both classification and regression
    - can balance imbalance data
  - `Weakness`
    - hard to control the result
    - interpretation problem
    - use large memory when computing large dataset

- `K-neighbour algorithm`
  - `Strength`
    - good in large dataset
    - can be used in both classification and regression
    - very simple
  - `Weakness`
    - have to scale the data
    - bad in high dimension
    - need to test the parameter k
- `Bagging Classifier`
  - `Strength`
    - reduces over-fitting
    - have many parameters to control the model
    - easy to implement
  - `Weakness`
    - consumes largest amount of time when using many features
    - uses high memory
    - loss of interpretability
- `Logistic Regression`
  - `Strength`
    - good when the class boundary is linear
    - efficient
    - can be used in both classification and regression
  - `Weakness`
    - bad in non-linear situation
    - can over-fitted easily
    - only predicts discrete value

The ranking list of five machine learning algorithms are shown below:

1. Logistic Regression
2. Multinomial Naive Bayes
3. Random Forest algorithm
4. K-neighbour algorithm
5. Bagging Classifier

For this task, the aim is to get high precision. According to the result and these analysis papers (5) (6), it shows that logistic regression is the best machine learning in this task. The second machine learning is multinomial Naive Bayes because of the assumption that some features are independent which can cause loss of accuracy. The third rank is K-neighbour algorithm. The problem of this model is it is necessary to find the right k value based on the dataset size. The reason why the fourth rank is a random forest algorithm is that it is hard to optimize value to the dataset even cross-validation has been used. Lastly, the least machine learning for this task is bagging classifier mainly due to the large consumption of memory and waste of time to get the model.

6 Conclusion

In conclusion, this work shows how to tackle the problem using data analysis and pieces of literature to guide. The result shows that each algorithm has its strengths and weaknesses which can identify the algorithm that is proper for the task. For insight, from the experiment and the data analysis, it shows that datasets still have a bias that greatly reduces the performance of the machine learning. For instance, some words still have the wrong spelling. There are some garbage words such as the string of random numbers. Most importantly, several abbreviations commonly used in the chat such as "ikr" and "b4". These things need to be fix.

References

[1] C. Boididou, K. Andreadou, S. Papadopoulos, D.-T. Dang-Nguyen, G. Boato, M. Riegler, Y. Kompatsiaris, *et al.*, "Verifying multimedia use at mediaeval 2015.," in *MediaEval*, 2015.

[2] W. Zhang, T. Yoshida, and X. Tang, "A comparative study of tf* idf, lsi and multi-words for text classification," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2758–2765, 2011.

[3] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, pp. 41–46, 2001.

[4] B. Trstenjak, S. Mikac, and D. Donko, "Knn with tf-idf based framework for text categorization," *Procedia Engineering*, vol. 69, pp. 1356–1364, 2014.

[5] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The impact of features extraction on the sentiment analysis," *Procedia Computer Science*, vol. 152, pp. 341–348, 2019.

[6] M. Aldwairi and A. Alwahedi, "Detecting fake news in social media networks," *Procedia Computer Science*, vol. 141, pp. 215–222, 2018.