Rumor Detection in Popular Online Social Media Posts

Kang Yueying A0084077Y
Liao Wenhao A0056048E
Mo Rufan A0206542M
Shuang Shuang A0084088W

Abstract

Information credibility is an increasingly important discussion in social media. Due to the ease of content generation in social media, the mere volume of data hinders readers to access accurate and credible contents. The recent Covid-19 case illustrated the ease of rumors to be spread around in social media across the globe. It results in either unnecessary fear or stress in the society or waste of resources due to spreading of inaccurate protection or measures. This paper develops a method for automating rumors on Twitter by learning to predict accuracy assessments through the usage of PHEME, a dataset of potential rumors in Twitter and journalistic assessments of their accuracies. We then analyze the model results to identify word patterns and features that contribute to the rumor detection. After that, we did a validation on a Covid-19 tweet sample to assess the model's capability.

1. Introduction

Emergence of social media platforms has changed people's behaviour of news consumption. More people start to seek and consume news from social media as compared to traditional news organizations due to its more timely nature and ease of sharing and commenting. Study has shown that there is a significant increase of adults from the US to seek news from social media from 49% in 2012 to 62% in 2016(Gottfried and Shearer 2016).

However, the quality of the news spread through social media is lower due to the ease of publication without proper verification. Rumours could be generated easily and spread at a much faster rate and volume as compared to before. There are multiple definitions of rumours. In context of unverified information spread in the social network, we adhere to the previous definition of rumour: information that is being circulated while its veracity is yet to be confirmed, and produces sufficient skepticism and/or anxiety (Allport and Postman, 1946; DiFonzo and Bordia, 2007).

The extent of reach of rumours was highlighted during the 2016 U.S. presidential election campaign. Studies have shown that the top 20% frequently discussed false election stories generated 8,711,000 shares, reactions, and comments on Facebook, which is much higher than the top 20% frequently discussed true election stories posted by major news websites (Silverman 2016).

Therefore, rumour now is deemed as one of the greatest threats to freedom of speech, social and political stability in recent days. Especially with the recent outbreak of novel coronavirus, it is observed that rumour has been generated on a daily basis and is spread through social media platforms. Such false information has caused unnecessary panic which posts threat to social stability.

1.1 Problem Statement

In light of the severe impact of the spread of rumour in social networks, our team wishes to study and build models for rumour detection in online social media, primarily based on textual contents and social network features, including post likes, repost network and authorship.

With the model established, we would like to apply the model to detect rumour generated from the recent novel coronavirus outbreak.

2. Dataset

2.1 Dataset Source

We have explored different sources of datasets to train and test our model. As our context is rumour detection in online social media, we focused on one of the most popular social medias which is tweeter. For model training, we use a dataset from the PHEME rumor scheme whereby the data are accurately labelled as rumors and non rumors. After data is ready, we apply the model to Covid 19 tweets to assess the model's capability

2.2.1 Model Training Data

Our team utilized the PHEME rumor scheme dataset which was developed by the University of Warwick in conjunction with Swissinfo, part of the Swiss Broadcasting Company(Buntain & Golbeck, 2017). Swissinfo journalists, working with researchers from Warwick, constructed the PHEME data set by following a set of major events on Twitter and identifying threads of conversation that were likely to contain or generate rumors. During each rumor selected in the PHEME dataset, journalists selected popular tweets extracted from Twitter's search API and labeled these tweets as rumor or non-rumor. For each tweet in this labeled set, the authors then extracted followup tweets that replied to the source tweet and recursively collected descendant tweets that responded to these replies.

This construction resulted in a set of 5798 labeled conversation threads totaling 103,212 tweets posted with 5 newsworthy events including the Ferguson unrest, the shooting at Charlie Hebdo, the hostage situation in Sydney and the crash of a Germanwings plane etc.

In this study, we mainly focus on whether the source tweet of the tweet thread is a rumor and non rumor, therefore, our analysis focused more on the 5798 source tweets. Among those source tweets, there are 34.0% rumours and 66.0% non-rumours.

Due to different sizes of discussion tweets for each event, we will split training and validation set in each event then combine them so the model developed could be more representative and generalized.

2.1.2 Covid - 19 dataset

We utilized tweets with hashtags relevant to Covid-19 cases as testing dataset. The relevant hashtags include: #coronavirus, #covid19, #coronavirus outbreak, #coronavirusPandemic, #covid19. The range of the tweets are from March 16 to 22, 2020.

The challenge we face is that there is a lack of accurately labelled Covid - 19 related tweets. Due to the recency of the event, there is no readily available labelled data. To combat this, we employed two approaches:

- manual label sampled tweets.We randomly select 60 tweets and our team members did manual label after doing the fact checking against trustworthy media
- sampled 20 tweets from fact-checking organization's tweet account to be as non rumor tweets. This is to improve the quality of labelled data. Below are some examples of the sampled tweets:

1.No evidence to suggest that the mixture of lemon and baking soda can kill SARS-CoV-2 virus

2.There is not enough evidence that a malaria drug and an antibiotic can lead to a quick recovery from Covid-19. Trials are ongoing into the use of hydroxychloroquine, but conclusive results have yet to be published.

3.Fact Check: Michigan Governor Did NOT Violate State's Social Distancing Order; TV Station Used File Footage https://ift.tt/2z8PamN #CoronaVirusFacts #DatosCoronaVirus

4.Coronavirus Update!Total 238 positive coronavirus cases report so far in #SriLanka according to the Epidemiology unit stats. 161 individuals under observation, 7 deaths, 68 recovered and discharged. 163 active patients.

2.2 Data Pre - processing

We perform below pre-processing steps to remove irrelevant information and reduce data noises:

- **Remove url**: many tweets contain url links to external websites. url address normally does not contain useful information, so they are removed.
- **Remove special characters**: Special characters including special symbols (e.g. #, @) and emoji are removed since they do not contribute to the explanation of the text.
- **Remove stop words**: Stop words defined in NLTK package are removed to reduce data noise and feature dimensions
- **Word lemmatization**: Use NLTK package to reduce inflectional forms and sometimes derivationally related forms of each word in tweet to a common base form.

2.3 Feature Engineering

2.3.1 Text representation

Tweet content is the most important feature for us and models to analyse and learn whether the tweet is a rumor or true information. In order to represent tweet content, every row of the dataset will be a single document of the

corpus. We chose TF–IDF Vectors as well as Word Embeddings as the feature creation method. The TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general, while word embedding preserve the order of the words and their contextual considerations.

2.3.2 Additional Features

Original tweets data are standard json objects returned from tweet API portal. In each json object there is additional information besides tweet content, such as tweet creation time and user profile. This data could also be important for us to identify whether the tweet is a rumour. So our preliminary task is to retrieve additional tweet information. After examining the tweet json object, we identify 10 more tweet features belonging to a. tweets syntactic features b. tweet interaction features c. user features. We normalise these additional features together with text features we have in section 2.3.1 to be used in our models.

a) Tweet syntactic features

Punctuations: Certain punctuations can be predictive of the speech act in a tweet. More specifically, the punctuation ? can signal a question or request, while ! can signal an expression of emphasis. These two punctuations also tend to capture readers' attention more due to their embedded meanings. Therefore, we have counted the number of appearances of these two punctuations (before doing data pre-processing) to indicate the lack of or appearance of these symbols.

url_count: Tweet content might contain urls that lead to other sources of information to backup the statement. Therefore we have included the counts of url (before doing data pre-processing) to indicate the presence of url inclusion behaviour.

Text_length, Average word length: These two features capture the length and word choice of the tweets. We have calculated the text_length and average word length after doing the text tokenization.

b) Tweet interaction features

favorite_count: tweeter could express their support of the tweet by clicking the favorite button. We include count of counts to assess support received from the tweeters

retweet_count: tweeter could also express their support by retweeting the tweets. The act itself also helps to propagate the tweet. Therefore the number of retweets indicates the level of propagation and popularity of the posts.

c) User features

verified: A verified Twitter user is a user that Twitter has confirmed to be the real. Verification is done by Twitter to establish authenticity of identities of key individuals and brands. The verified status of a user is visible to everyone.

followers_count: The followers of a user are other people who receive the user's tweets and updates. We include the count of followers to indicate the account's popularity

friend_count: The friends of a user are other people who the user followers. We include the count of friends to indicate the level of interaction the tweet user perform with other accounts

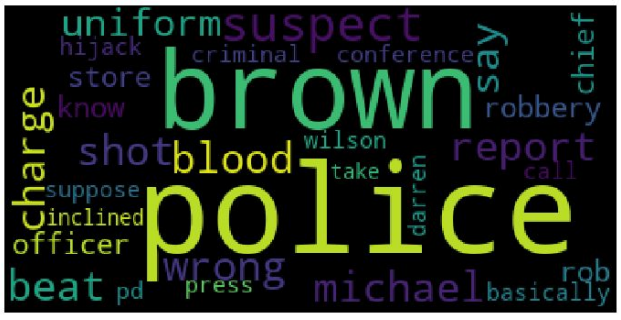
user_age: user age is calculated by taking the difference between the timestamp the post is posed and the timestamp the account is created. It could indicate whether it is a long-established tweet account or an account that is just recently built.

After feature generation, we split the training and testing portion as 80% and 20% for model training.

2.4 Exploratory Data Analysis

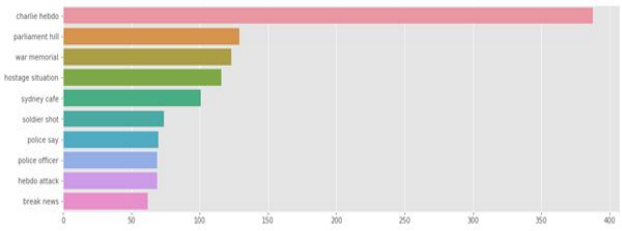
In total there are 3826 non-rumours and 1972 rumour tweets related to 5 different events. Although the number of rumours is less than the number of non-rumours, data is comparatively balanced for most of the events.

Firstly we do a word frequency analysis of the data. Below is the word cloud diagram for the entire 5798 source tweets. As shown, the high frequent words are much event context based.



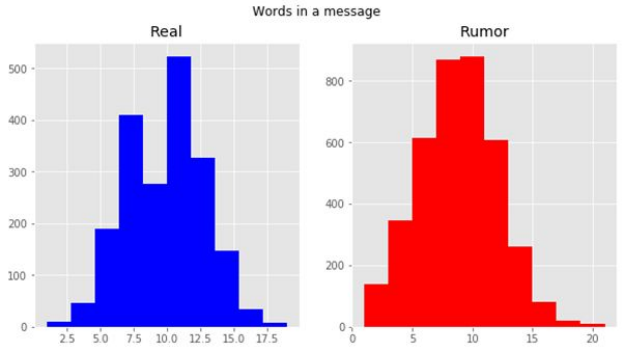
Rumor Detection in popular online social media posts

Next, we did a bigram analysis over the tweets. The most common bigrams are: charlie hebdo, parliament hill, war memorial, hostage situation, sydney cafe, soldier shot, police say, police officer, hedbo attack and break news:

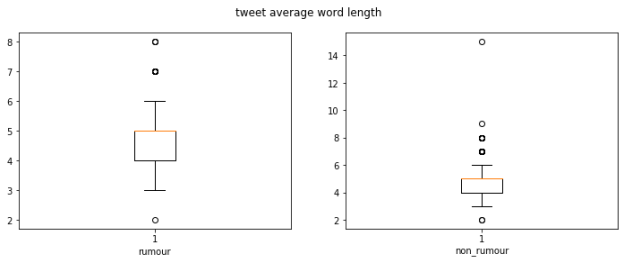


At the individual feature level, our initial findings show that there are differences in tweet syntactic feature distribution between rumours and non-rumours. Especially in average tweet length, user age and punctuation count.

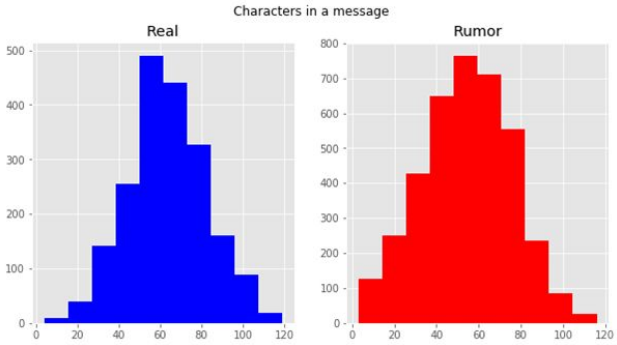
Text Length: text length is the number of words in a message. Rumor tweets has a slightly higher average text length than non rumors



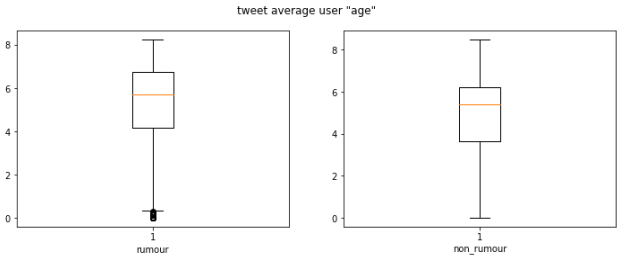
Average word length: the average word length in rumours is slightly longer than that in non-rumours. This could be because rumours tend to use more complicated words to draw people's attention and make content seem more convincing.



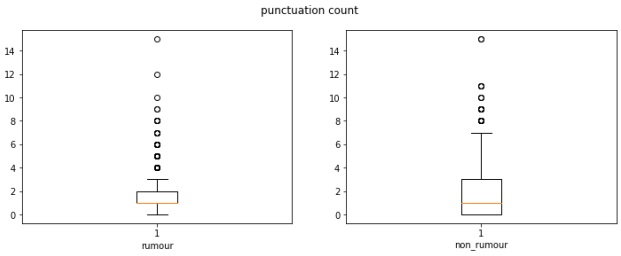
Character Count: Average character length for rumors is slightly higher than non rumors as a result of usage of longer words.



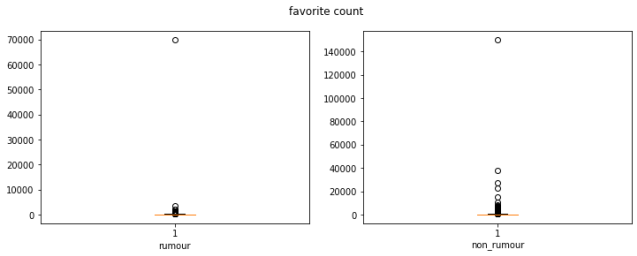
Average user age: although overall user age for rumours and non-rumours are similar, there are more outliers in rumours. The reason could be some people are creating new accounts to spread rumours on purpose.



Punctuation Count: Rumours tend to have less punctuation count, which is indicating the content is less structured and the creator is less prepared.



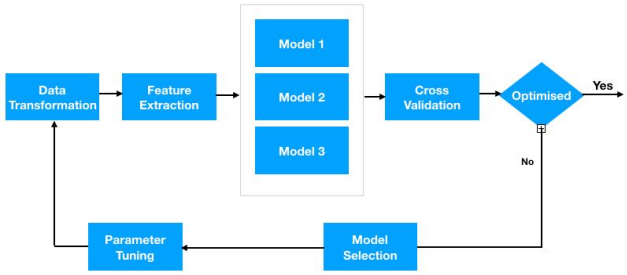
Favourite Count: majority of tweets do not receive many likes. But there are more outliers observed in the non_rumour dataset.



3. Methodology

We build up a processing pipeline by applying multiple transformation methods and machine learning models in our project. Parameter Tuning and Cross Validation should be enabled to optimise final outcomes.

The overall process flow is illustrated as below:



3.1 Models

3.1.1 Machine Learning Models

As a start, we choose Naive Bayes and Logistic Regression as our base model since they are simple and quick in generating results as the baseline. We tried to build these models using text features only first then together with the additional features that we identified. However the accuracy, which we selected as the metrics for model evaluation drops hugely when we train Naive Bayes and Logistic Regression using text and additional features.

Then we choose another two more efficient and flexible models which tend to produce good results: XGBoost and LightGBM. After the first iteration of model building, we found that XGBoost performs slightly better than LightGBM in terms of test accuracy score. Therefore we decide to further tune the hyperparameters of the xgboost model to find the best model result.

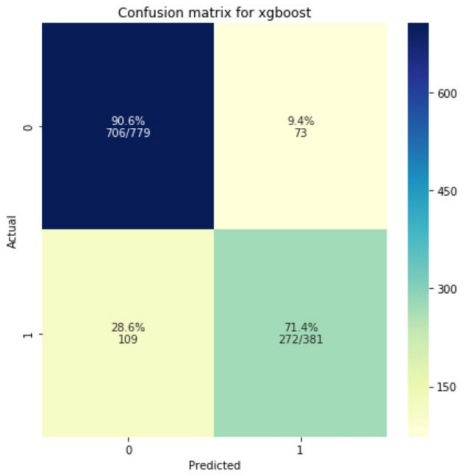
Firstly we decide on the hyperparameters that may have more influence in the model behavior:

1. **max_depth**: The maximum depth of a tree. It is used to control over-fitting as higher depth will allow the model to learn relations very specific to a particular sample. Normally it has values from 3 to 10.
2. **min_child_weight**: Defines the minimum sum of weights of all observations required in a child. It is used to control over-fitting. Higher values prevent a model from learning relations which might be highly specific to the particular sample selected for a tree.
3. **subsample**: Denotes the fraction of observations to be randomly sampled for each tree. Lower values make the algorithm more conservative and prevents overfitting but too small values might lead to under-fitting. Its typical values ranges from 0.5 to 1
4. **colsample_bytree**: Denotes the fraction of columns to be randomly sampled for each tree. Its typical values ranges from 0.5 to 1

Secondly, we use the default learning rate 0.3 and determine the optimal number of trees for this learning rate using XGBoost in built function cv to perform cross-validation at each boosting iteration and thus returns the optimal number required (210).

Then we have performed a Grid Search using 3-Fold Cross Validation in order to exhaustively search in the hyperparameter space for the best performing combination.

Lastly we train our model using the best performing parameters and evaluate the model performance using test split data, observing an increase of 2% in the test accuracy score. The full confusion matrix is shown below:



3.1.2 Neural Network with Embedding

Next, we tried to use Keras to build a neural network model for this problem. As data cleaning is done in previous steps, we jump into corpus and model building.

Here we have two choices of corpus. One is to build our own while the other is to use the pre trained GloVe corpus.

We build our bag of words by using CountVectorizer from package sklearn to convert the dataset into a matrix of tokens. Since there are many words in the bag but not all are important, we use term frequency–inverse document frequency (tf-idf) methodology to change the weight of the words and convert them to a matrix of TF-IDF features.

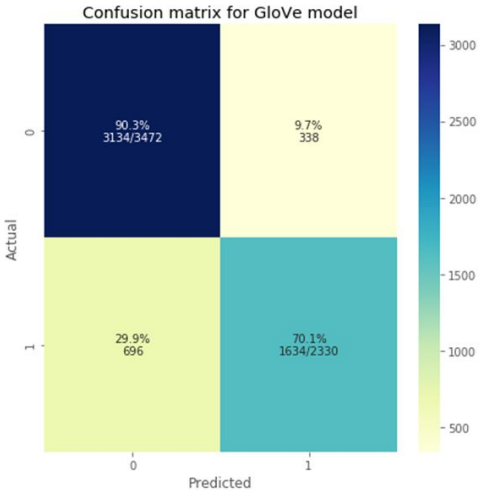
After testing, the result is not ideal, so we use GloVe (global vector for word representation) pretrained corpus model to represent our words. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. We downloaded the pretrained word vectors glove.6B.zip from https://nlp.stanford.edu/projects/glove/.

We then calculated the number of unique words is 5356 in our dataset, which will be used as the input size of our keras model.

Next, we build our keras model with a sequential model. Hyperparameters are:

1. **input layer:** Since we are converting corpus vectors into Dense vectors for training, we chose the embedding layer as our first layer.
2. **embedding initializer:** An array in shape of number of unique words filled with zeros.
3. **optimizer:** Adam
4. **loss function:** Binary_crossentropy
5. **metrics:** Accuracy

The test results are below:



We then conducted k-fold cross validation with 10 folds. The average accuracy is 81.1%.

We also tried to build another NN model test with corpus and all the additional features. However, the result was not good (67% accuracy) so we didn't continue with that model.

4. Result and Evaluation

4.1 Accuracy measure

Since this is a classification question, we would need to select one of the accuracy measures to assess model performance.

In the context of popular events, especially for the event of Covid-19, both false positive and false negative have a significant impact. False negative means a rumor tweet is wrongly identified as a non rumor tweet. If such a tweet goes unchecked, false information will be distributed which causes social disturbance. For example, in the scenario of Covid-19, false information of confirmed number of cases would lead to the public not paying enough attention to this issue. False positive means a non rumor tweet is wrongly identified as a rumor. It will bring equally severe influence. For example, if a tweet of the importance of wearing a mask is being wrongly identified as a rumor and being removed from the tweet platform, the tweet will lose its power of influencing the society to practice necessary protective actions.

Therefore, we choose accuracy score as our final measure.

4.2 Model Results

We have used a series of machine learning models and tried with and without the additional features mentioned in section 2.3.2.

We split the data into training and testing sets and apply the models trained to the testing datasets. Below is a summary of the testing accuracy score for various models.

Model Type	Without Additional Features	With Additional Features
Logistics Regression	0.82	0.67
Naïve Bayes	0.83	0.67
XGBoost	0.83	0.84
LightGBM	0.82	0.83
NN with Embedding	0.81	0.67

As can be seen through the table results, there is an interesting pattern that additional features mentioned in section 2.3.2 help to improve the accuracy score for XGBoost and LightGBM model but lower the accuracy score for SVM and linear regression. One possible reason might be due to the overwhelming number of independent variables.

Among all the models, XGBoost performs the best. Also, the additional features (text syntactic, tweet interaction and user) actually help to slightly improve the accuracy score.

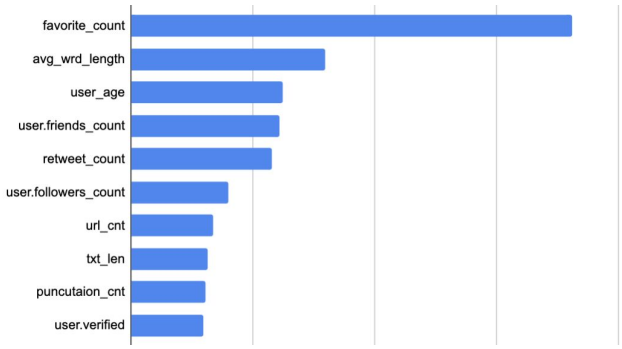
Therefore, among machine learning models, XGBoost with additional features are chosen to compare with the deep learning models

4.3 Results Analysis

In total there are 4821 features which only 10 are additional features. Therefore, I will firstly analyze the results of additional features followed by the frequency of words.

4.3.1 Feature importance of additional features

After removing the work tokens, a feature importance graph is constructed as shown below:



As shown, the top 5 features among all the additional features are : favorite count, average word length, user age, user friends count and retweet count.

Favorite count, Retweet count: These indicate the popularity of the posts play an important role in detection of rumors. As mentioned in section 2.4, non rumors tweets tend to have more extreme favorite counts as compared to non rumors. This might indicate in general, the public has sufficient common sense to identify a rumor. This might probably be due to citizen's growing adaptability to identify fake news/reports patterns.

Average word length: As mentioned in section 2.4, in rumor datasets the average word length is longer than the no rumor datasts. Therefore, the analysis further confirms the hypothesis that this difference contributes significantly to the model. This shows that tweets with more complicated words might be a good indicator for a rumor tweet.

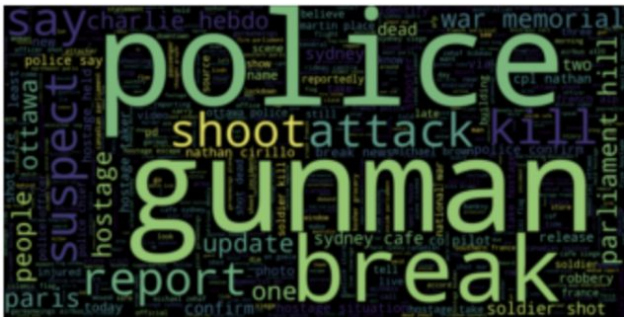
User Age: As mentioned in section 2.4, user age for rumor datasets are smaller than non rumors. The feature importance further proves that a relatively new account is more prone to spread rumors since rumor distributors tend to create new accounts for disseminate rumors.

User Friend Count: this finding is quite to the contrary of common sense. Number of friends is the number of tweet accounts a user follows. Conventional thinking might be that follower count might be of higher importance than friend count because the number of followers indicate how many twitter accounts support this

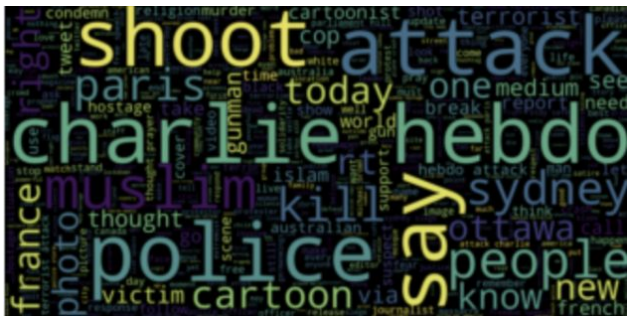
account, and it only can be changed by others initiating a follow action. In contrast, friend count could be easily faked by mass following other accounts remove follows. These results might show that tweet accounts that spread rumors might pay more attention to follower count and neglect the importance of friend count when trying to mimic a normal account.

4.3.2 Word Frequency analysis

The first figure is the word cloud for rumor tweets:



The second figure is the word cloud for non rumor tweets:



As can be seen, although words such as “police” “shoot” appear in both word clouds, the high frequency words in rumors are only concentrated among a few words; whereas for non rumors tweets, there are more high frequency words. Also, the high frequency words in non rumors then to have more factual words such as “Charlie”, “Sydney”, “Muslim”, “Paris”, “France”; whereby for rumor tweets, high frequency words then to be more eye catching words such as “gunman”, “break”,”suspect”, “hostage”. This could indicate that in rumor tweets, more sentimental words which could create attention and negative annotations appear to be more frequently used.

Another discovery is that high frequency keywords are contextual and more relevant to a specific event. This suggests that the model needs to be trained for a specific event context. It needs enough training data for it to perform accurately.

5. Model Application

As mentioned in section 4, when sufficient training data is provided, the XGBoost model with additional features could achieve a high accuracy rate of 84%.

In light of the recent Covid-19 cases, we wish to apply the model to Covid-19 related tweets and see the rumor detection capabilities.

As described in section 2.2.2, we have sourced Covid-19 related tweets and manually labelled them. Due to the time and resource constraints, we labelled 80 tweets, out of which 9 are rumors and 71 are non rumors.

After applying the model to the datasts:

- out of 71 non rumors, we predict 66 correctly
- out of 9 rumors, we predict 1 correctly

Although the accuracy is 84%, the precision is only 17% and recall is only 11%. We have identified the reason for low performance

Lack of labeled training data on Covid-19 tweets

As mentioned in section 4.3.2, the word frequency shows that high frequency words are context based. The training data that we used to train the rumor detection model does not include any pandemic or Covid-19 scenarios. Therefore, the performance of the model on a new topic with new sets of corpus might not be good.

Lack of professional knowledge to label data

Another reason is that the label is done by our own team members instead of a professional team. Although we try our best to do the fact check, it may still differ from a professional opinion.

Imbalanced test data

In the test data, only 11% of the data are rumors. This imbalanced data leads to high accuracy score despite low precision and recall

Nevertheless, with the high accuracy for within event type prediction, the model still shows potential to be implemented in the future when more labelled data are

available. Model could be readjusted then to be applied to detect rumor tweets for Covid-19.

6. Limitation and Future Work

Our original intention was to apply this rumor detector to identify the rumors in the Covid-19 related tweets and news. However, as mentioned in section 5, due to the nature of text, the corpus of different scenarios can vary a lot. The training dataset we used only covers the cases in its own scenario.

As Covid-19 is still new to humans, the related tweets and news can cover a wide range of topics including medical science, sociality, climate, geography, economics, politics, etc. The size of data needed to build a comprehensive corpus is tremendous, so is the effort needed to properly label these data as rumors or facts. The resources and knowledge of our team is limited. We may not be able to cover such a wide range of topics and label the messages properly.

The future work for this project can be continuing to gather datasets with proper labels. With a more comprehensive dataset, we can enlarge our corpus, enhance the model performance and apply it into covid 19 related topics.

References

Gottfried, Jeffrey, and Elisa Shearer. 2016. "News Use across Social Media Platforms 2016." Pew Research Center, May 26. https://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/

Craig Silverman. 2016. This analysis shows how viral fake election news stories outperformed real news on Facebook. BuzzFeed News 16 (2016)

DiFonzo, N. and Bordia, P. (2007). Rumor, gossip and urban legends. Diogenes, 54(1):19– 35

Procter, R., Crump, J., Karstedt, S., Voss, A., and Cantijoch, M. (2013). Reading the riots: What were the police doing on twitter? Policing and society, 23(4):413–436.

Bazerli, G., Bean, T., Crandall, A., Coutin, M., Kasindi, L., Procter, R. N., Rodger, S., Saber, D., Slachmuijlder, L., and Trewinnard, T. (2015). Humanitarianism 2.0. Global Policy Journal.

Zubiaga, Arkaitz; Wong Sak Hoi, Geraldine; Liakata, Maria; Procter, Rob (2016): PHEME dataset of rumours and non-rumours. figshare. Dataset. https://doi.org/10.6084/m9.figshare.4010619.v1

Buntain, C., & Golbeck, J. (2017). Automatically Identifying Fake News in Popular Twitter Threads. 2017 IEEE International Conference on Smart Cloud (SmartCloud). doi: 10.1109/smartcloud.2017.40

Github Project: https://github.com/ppplalala-wh/BT5153_Group_K.L.M.S.git