Rumour Detect & Analysis
Natural Language Processing Project 1

Authors:
Ari Boyd, 992301
Chenghao Li, 1067999

1 Introduction

As the Covid-19 pandemic unfolded, citizens and organizations of many countries took to social networking platforms to spread their knowledge surrounding the phenomenon.
Some of this knowledge was based on substantiated facts such as data from government sources like the World Health Organization. However there was also a large number of tweets making statements that were based on hearsay, or personal opinion presented as fact.
These statements, commonly referred to as "rumours" are an important class of social media objects to understand, as they could range from harmless assertions, to potentially deadly recommendations.
This report details processing and classifier training on a set of 'tweets' from the popular online platform 'Twitter'. The tweets are grouped into conversation threads, where the first 'source' tweet has been labelled as either a rumour or a non-rumour. The data is split into a training set, a development set, and a test set that has no associated labels.

2 Rumour Detection Modelling

2.1 Preprocessing

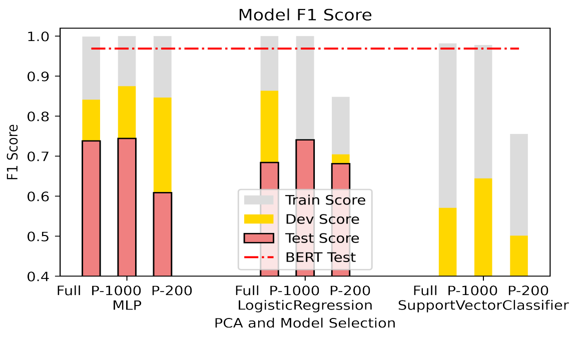
In order to perform classification, we first transform the labels for the source tweets from 'rumour' and 'nonrumour' to 1 and 0 respectively. We then group each source tweet's reply id set into a list sorted by time of tweet creation. Using that list, we get the text for the source tweet as one attribute, and then we get the concatenated text of all of the replies to the source tweet as another single attribute. We group the replies in this way as our BERT method can take up to two sentences, the source and the grouped replies. The labels for each source tweet are joined onto their row (except for the test set). This is the general pre-processing, but both models require additional, separate processing as they need very different input.

2.2 Model Selections

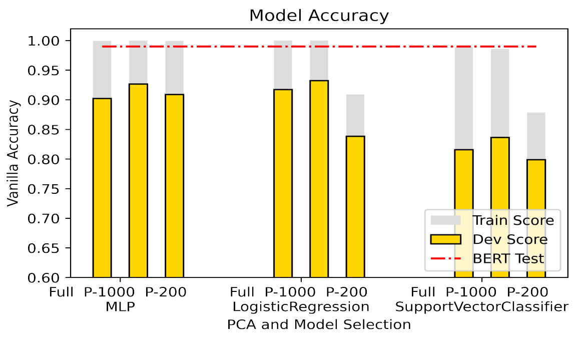
2.2.1 Model: Classic Classifiers on Bag of Words

The first approach tested was to apply a Bag of Words tokeniser to only the source tweet's text, then train a variety of standard classifiers on it. We tried this approach because it is fast to implement, is likely to give reasonable results, and can be thus used as a baseline to a more advanced approach.
In order to apply this method, some additional preprocessing is necessary. We remove any words with numbers, non-English words, URLs, and stop words. We then use the WordNet lemmatiser to convert them into lemmas in order to reduce the size of the input. After the BoW tokeniser is run, we apply a standardization scaler, fitted on the training set, to ensure our data has 0 mean and unit variance so we can apply PCA. PCA is applied here because the non-reduced data has over 5000 attributes, which may easily lead to overfitting in some models. We train models on the scaled with no reduction data, data reduced to 1000 components through PCA, and data reduced to 200 components. We train three models on this data. Model 1 is a Multilayer Perceptron (Popescu et al., 2009), which is just a feedforward neural network with a single, 100 neuron hidden layer. Model 2 is a Logistic Regression model, which models the probability of the classes as the log-odds of a linear model (DeMaris, 1995). Model 3 is a Support Vector Classifier (Boser et al., 1992) with linear kernel, which attempts to find a hyperplane to divide the two classes. As is visible from Figure 1a, MLP and Logistic Regression perform well on fitting the training set, and have generally good performance on the development set, but struggle on the test set. The SVC struggles greatly on the development set, and is thus not

worth assessing on the test set. Looking at the 6 test F1 scores, 1000 component PCA seems the best feature set, and the MLP trained on it is the strongest with an F1-score of about 0.74. Although this makes it the best representative of the standard classifiers, its test score is still rather lacking. One possible explanation for both MLP and LR's poor test performance is that the test set contains many words unseen in the training set, made clear by observing Figure 2. We can see here that the development dictionary contains only 28% of the words observed in the training set, and the test set contains an even smaller 20%. This makes it difficult for these models to generalise, as they are using a bag of words approach. We attempt to alleviate this problem somewhat by using a pretrained model, detailed in the next section.



(a) F1 Score



(b) Accuracy

Figure 1: (a) Train, dev, and test F1 scores for MLP, Logistic Reg and SVC on full, 1000 PCA, and 200 PCA data. (b) Train and dev accuracies for the three models on the three feature sets

2.2.2 Model: BERT

The approach that yielded the best results for us was applying the pre-trained Bidirectional Encoding Representations from Transformers (Devlin et al., 2018) for Sequence Classification to the full conversation thread text using the Hugging Face implementation(vinai, 2021). Out of many existing transformers, this bertweet model captures recent tweets with COVID-19 covered. It also uses "cased" model since the alpha form of a word could contain sentiment information. We tried this approach because it provides State of the Art performance on many natural language tasks, including classification tasks(Devlin et al., 2018). BERT is a stack of transformers that has been trained to do missing word or next sentence prediction on a massive corpus, and can then be retrained on a target application to yield very useful word embeddings. In terms of BERT specific preprocessing, we ran a prebuilt BERT tokeniser on the source text and the replies text. We also normalized the Tweets by converting user mentions and url links into model recognised tokens so our model can make use of those special tokens, with tests been conducted, by making use of special tokens can we improve the model further. We then we use a data collator to split the data into batches. After fitting the model with training data, its performance is evaluated on the development data using a variety of batch sizes, number of epochs, and dropout rates. The different hyper-parameters tested, and their resulting F1 Scores on the development set can be seen in 1.

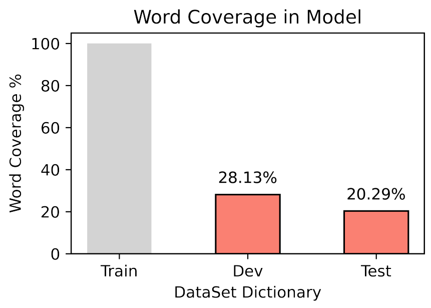


Figure 2: Percentage of Seen Words

Analysing Table 1, we see that the number of epochs appears to increase the performance up until 10 epochs, after which it decreases. Similarly with batch size, the optimal spot seems to be batches of 20, with 8 being too small, and 36 too large. Finally, adding some dropout appears to enhance the results. In sum, the best performing model was the one with 10 epochs, a batch size of 20, and a 0.1 dropout rate in the neural network classifier's hidden layers. Even the weakest of the BERT models, with a test score of 0.813, still outperformed the best standard classifier's 0.74. The best BERT model scored 0.969 on the public set, a remarkable improvement, and achieved a private score of 0.92405.

Epoch	Batch	Dropout	f-beta	Score
3	8	0.0	0.356	0.813
3	8	0.2	0.368	0.865
10	20	0.1	0.380	0.969
20	20	0.1	0.376	0.920
20	36	0.1	0.368	0.915

Table 1: Performance of BERT model (measured with macro f-beta score with beta=2) on the constructed validation sets, and public test set. These are a section of our tuning results.
Epochs: number of iterations of training classifier.
Batch: mini-Batch Size used for training BERT, higher batch leads to higher performance and effectiveness but needs more memory.
Dropout: dropout ratio for neural network classification hidden layers. Higher dropout would cause model learn too little.

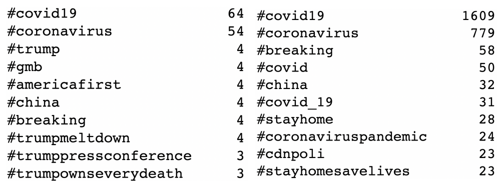
2.3 Evaluation

Why do we choose to use f-beta with beta = 2 rather than f-1 or plain old accuracy as our main evaluation metric? The reason is that we have imbalanced classes as mentioned before, therefore the model could tend towards predicting only the majority class while still achieving a high accuracy score. With the smaller training sample of rumour tweets it is detrimental to classify rumour tweets as non-rumours. We need to have high recall[1]. Since f-1 score is f-beta with beta = 1. Therefore, to adjust more weight towards recall, we set beta > 1, beta = 2. In addition to that, we also use Macro averaging rather than weighted averaging, which also treats the minority label "rumour" as important as "non-rumour".

3 Rumour Analysis

In this section of the report, we discuss our analysis on Covid Tweet data using our model, as well as several off-the-shelf classifiers. This data set contains 250,000 tweet ids, which we preprocess in the same way as our training data, and classify using our model. The data was collected using standard twitter API and a python crawler on the twitter ids provided by teaching team. We found that nearly 11.82% of these COVID-19 related tweets have been classified as rumours. The data is split into rumour and nonrumour tweet threads, based on our classification, so we can examine the differences. We then perform several analyses on the rumour and nonrumour tweets to try to identify interesting differences.

3.1 Hashtag Analysis



#covid19	64
#coronavirus	54
#trump	4
#gmb	4
#americafirst	4
#china	4
#breaking	4
#trumpmeltdown	4
#trumppressconference	3
#trumpownseverydeath	3

(a) Rumours

#covid19	1609
#coronavirus	779
#breaking	58
#covid	50
#china	32
#covid_19	31
#stayhome	28
#coronaviruspandemic	24
#cdnpoli	23
#stayhomesavelives	23

(b) Nonrumours

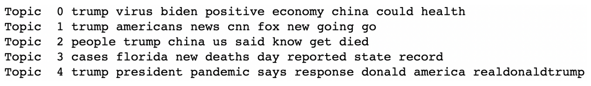
Figure 3: Top ten hashtags and their counts for rumour and nonrumour tweets

With this analysis, we seek to answer the question, “What kind of hashtags are used by rumour and nonrumour tweets, and what is the overlap/difference between them”. As hashtags are used to spread tweets that share a common theme, examining them gives us insight into what sort of things are being expressed by rumour and nonrumour tweeters. The hashtags are extracted using regex, lowercased, and counted. As is visible from Figure 3, the hashtags tweeted in rumours and nonrumours are quite similar, but do have some differences in content. The rumour hashtags contain more references to former American president Donald Trump, whereas the nonrumours have #stayhome and #stayhomesavelives. This suggests rumour tweet’s topics are more often related to Donald Trump, and nonrumour tweet’s topics are more often related to preventing covid spread. A similar result is observed later in the topic analysis section. One other point of note is that while the nonrumour class has only 7 times as many tweets as the rumour class, the nonrumour tweets contain over 25 times as many #covid19 instances compared to the rumour tweets. A similar situation occurs with #coronavirus. This suggests that nonrumour tweets contain more references to the actual virus than rumours, although the reason as to why this is the case is unclear.

3.2 Topic Analysis

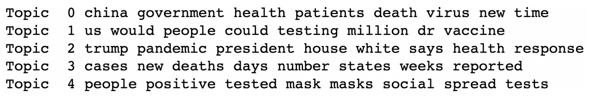
Additionally, we analyse the source texts to see if there are differences between the rumours and nonrumours in terms of topics, using a Latent Dirichlet Allocation (Blei et al., 2003). Latent Dirichlet Allocation is a method in which the topic distribution of words and documents are both

[1]Recall: refers to the percentage of total relevant results correctly classified by your algorithm.
Precision: refers to the percentage of the model’s predictions which are relevant



```
Topic  0 trump virus biden positive economy china could health
Topic  1 trump americans news cnn fox new going go
Topic  2 people trump china us said know get died
Topic  3 cases florida new deaths day reported state record
Topic  4 trump president pandemic says response donald america realdonaldtrump
```

(a) Rumours



```
Topic  0 china government health patients death virus new time
Topic  1 us would people could testing million dr vaccine
Topic  2 trump pandemic president house white says health response
Topic  3 cases new deaths days number states weeks reported
Topic  4 people positive tested mask masks social spread tests
```

(b) Nonrumours

Figure 4: Top eight words for each of the five topics generated for rumour and nonrumour tweets

Label	Sentiment	Source	Replies
Non-Rum	Neg	8167	7567
	Neu	5701	6970
	Pos	1526	857
Total		15394	15394
Rumour	Neg	1108	1277
	Neu	840	737
	Pos	116	50
Total		2064	2064

Table 2: Sentiment statistics on raw twitter text data. Tweets classified as Negative are counted as -1, Neutral as 0 and Positive as 1.

updated iteratively, through random sampling, until the distributions converge. This approach requires some additional preprocessing. Any tweet with no source text is dropped, and URLS and all numbers are removed, as URLs are hard to interpret, and including numbers can lead to some very strange topics tokens. Then stopwords are removed using the NLTK english stopword corpus, extended with "coronavirus" and "covid", as these words are in many tweets and tell us very little to distinguish the topics. Using sklearn's CountVectorizer, the texts are transformed into count vectors of their tokens, with the top 10% and bottom 0.01% frequency tokens removed to avoid overly general tokens and noise. A 5 topic LDA is then fitted onto each dataset, and the resulting top 8 words for each topic output. These outputs can be observed in Figures 4a and 4b. From these topics, a pattern similar to the hashtag analysis emerges: rumours are more often concerned with former American president Donald Trump, and nonrumours contain topics like the vaccine and masks. There is also some overlap, as both Topic 3s seem to be about the number of new cases.

	Metric	Source	Reply
Non-Rumour	**mean**	-0.271	-0.436
	std	0.630	0.598
Rumour	**mean**	-0.480	-0.594
	std	0.602	0.538

Table 3: Metric to quantify overall sentiments with negative number represents stronger negative responses and vice versa.

3.3 Sentiment Analysis

Another point of interest is the general sentiment expressed by rumour and non-rumour tweets. This is a standard question to ask when viewing two groups of tweets, and we use an off-the-shelf classifier to get the sentiment, which is a pre-trained transformer model on tweets data from January 2018 to December 2021 with the COVID-19 pandemic period data covered (cardiffnlp, 2022). From our results it is clear that the general sentiment of COVID-19 rumours is negative, but rumours are on average more negative than nonrumours. The raw data for the sentiment analysis is visible in Table 2.

We assign -1 to mean a negative sentiment, and +1 to be positive sentiment. Applying the central limit theorem we can assume the data follows a normal distribution, and after doing some calculations, the results are shown in Table 3. From the Table, the source of the rumour tweets tend to deliver negative sentiment related information (-0.480) and they receive more negative public replies (-0.594). Unfortunately, our analysis can't determine the reason for more negative replies on rumour tweets. For example, it could be that the rumour tweets incite hatred in the general public, or it could be the general public criticising the rumours.

The general negative sentiment of both classes is understandable. Most of the topics discussed in tweets concern things such as the number of deaths and various lock-downs, so it is reasonable that the general sentiments are towards negative aspects.

4 Conclusion

In this report we address rumour classification and analysis. Out BERT model greatly outperforms the standard classifiers, achieving good performance. Our analyses found rumours are more negative and concern Donald Trump more than nonrumours.

5 Team Contribution

992301: Task 1: I tried a couple classifiers that performed poorly with my implementation, including an lstm on the source and reply text concatenated. They were thus abandoned in favour of the state of the art, BERT. I also suggested some of the hyperparameters for bert we tried.
Task 2: I did the hashtag and topic analysis.
Report sections I wrote: Introduction, preprocessing, standard classifiers, task 2 introduction, task 2 preprocessing, hashtag and topic analysis, and conclusion.

1067999: Task 1: Firstly, crawl the data, then did classic bag of words model (Logistic Regression, MLP, SVC) with PCA. Then tried several Bert Models with different pre-trained dataset. Use params suggested by my teammate to fine-tune the model.
Task 2: Did label prediction and sentiment analysis after crawling the dataset.
Report section: BERT, Evaluation, sentiment analysis.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA. Association for Computing Machinery.

cardiffnlp. 2022. Twitter-roberta-base for sentiment analysis - updated (2021).

Alfred DeMaris. 1995. A tutorial in logistic regression. *Journal of Marriage and Family*, 57(4):956–968.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Marius-Constantin Popescu, Valentina Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. 2009. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8.

vinai. 2021. Bertweet: A pre-trained language model for english tweets.