COMP90042 Natural Language Processing Project 2022

Rumour Detection and Analysis on Twitter

Anonymous, Anonymous, Anonymous

Task 1

1.1 Data Structure

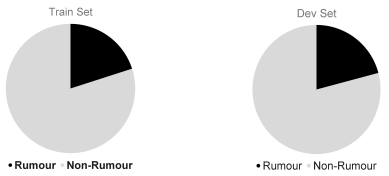
With the 30,419 tweet id given for the train set and 10,554 tweet id for the development set, we have successfully retrieved 20,323 tweets in the train set and 7,234 tweets in the development set. Around 20% of the original tweets are unretrievable via the Twitter API, which is typically due to the deletion of tweets or the fact that the user has been suspended by the Twitter platform.

We have removed the instances(events) with the absent source tweets for both the training set (1,895 events) and the development set (632 events) as the classification of tweets significantly hinges on the context of source tweets (see Section 1.2). Based on the above steps, the size of sets we used in Task 1 is as shown in Table 1.

# of instances	Original	After Removal
Train Set	1,895	1,557
Dev Set	632	532
Test Set	558	558

Table 1: The Number of Instances Before and After Trimming

And the distributions of the binary classes in the train set and dev set are seen in Figure 1.



20.04% rumour 20.86% rumour

Figure 1: The Distribution of Two Classes in Both Sets

As seen in the above diagram, the distributions of two classes are similar which meets the critical distribution assumption for machine learning.

1.2 Feature Selection and Engineering

We have tested the performance of different feature selection and means of engineering.

1.2.1 Feature Selection:

I. Only use the text of the source tweet
II. Use the texts of all tweets in each event, with all reply tweets sorted by created timestamp.
III. Use the text of the source tweet and the last three (at most) reply tweets chronologically sorted.
IV. Use the text of the source tweet and the last ten (at most) reply tweets chronologically sorted.

1.2.2 Feature Engineering:

a. Original tweets texts.
b. Stopwords Removal.
c. Denoting URLs by “HTTL” tokens.
d. Denoting @mentions by “@USER” tokens.

The performances (F1 score for development set) of combinations between various feature selections and engineering as shown in the table below:

Selection	Engineering	Accuracy	F1-score
I	a	0.94361	0.91036
	cd	0.90977	0.83952
	bcd	0.92857	0.88117
II	a	0.93421	0.90667
	cd	**0.97368**	**0.96414**
	bcd	0.95301	0.84232
III	a	0.91165	0.88191
	cd	0.96053	0.94157
	bcd	0.91541	0.88636
IV	a	0.95865	0.93937
	cd	0.95489	0.93426
	bcd	0.93985	0.91235

Table 2: Performance of Models with Various Feature Selection and Engineering Combination

Note: the performance is based on a model with BERT pre-trained layer and RNN fine-tuning. With all other parameters set equal (batch size = 4 and epoch number = 5).

The above result is not surprising. Feature II (all tweets included and sorted) outperforms the other feature selections (with padding set to be 512 in the BERT-tokenizing process). Because a higher volume of

retweets will include more information regarding the attitudes of re-tweeters towards source tweets, making the features more informative.

The engineering process regarding @ mention and URL replacement improves the performance of our model because the URL and @ mention are typically less informative, which would also increase the noise in the model.

The removal of stopwords (from the NLTK dictionary) does not improve the performance of the model, which is supported by the findings by Qiao et al., (2019). Negation words included in the NLTK stopword dictionary ("not", "nor" "wasn", etc) play a significant role in indicating the relations between reply tweets and source tweets. By removing stopwords, the features become less informative for the model.

Therefore, we decided to use the combination of II (all sorted tweets in the event) + cd (replacement towards URL and @ mentions) to feed our models.

1.3 Model

1.3.1 Baseline: Logistic Regression

The Baseline model we chose is simply a Logistic Regression model. The performance of LR in this task is surprisingly good, with an accuracy of 90.226% and a macro-F1 score of 84.232%.

In the following models, we introduced Bidirectional Embedding Representation from Transformer (BERT) (Devlin et al., 2018., pp.5) from the pre-trained model. The features are preprocessed as described in section 1.2.

1.3.2 Model 1: BERT with LSTM

For an attempt at the RNN model, we have implemented the Long-short term memory model. RNN model connects the previous information with the present state but may lose long-term information. LSTM is a type of RNN, but with the ability to solve the vanishing gradient problem with the memory cells. Since tweets can be long, it has been expected that the memory cells might be helpful for capturing long-range dependencies. This model feeds the last layer of the pre-trained BERT to an LSTM and uses a fully connected layer with the sigmoid function for final classification. However, after tuning with several input dimensions, output dimensions, hidden layers, bach sizes and epoch numbers, the best result obtained was an accuracy of 80.26%, which is even lower than our baseline.

1.3.3 Model 2: BERT with Single-Layer FNN Classifier

After pre-trained with the BERT, we directly used the last layer for further rumour-classification tasks. The parameters are fine-tuned to optimise the rumour labelling.

A considerable amount of effort was allocated to this model. We have adjusted the batch size to capture the best performances within the certain epoch ranges (5/10/20).

Batch Size	Epoch #	Accuracy	F1-score
1	5	0.94925	0.91719
	10	0.96805	0.95113
	20	0.96805	0.95113
2	5	0.96992	0.95385
	10	0.95865	0.94010
	20	0.97774	0.96585
4	5	0.97368	0.96414
	10	**0.98120**	**0.97135**
	20	0.97932	0.96859
8	5	0.96241	0.94191
	10	0.97180	0.95688
	20	0.97368	0.95989
16	5	0.95301	0.93333
	10	0.96617	0.94911
	20	0.96617	0.94911

Table 3: Performance of BERT with FNN Classifier

As seen from the statistics, the overall performance of this model stands out among all other models we have developed. The employment of BERT pre-trained layer.

We notice the drastic improvement of performance within the first five epochs, with the next five epochs leading to a subtle improvement. In our experiment, the improvement from epoch 11 to epoch 20 is negligible.

Although we failed to test the performance of this model with a larger batch size due to the GPU limitation, we found it surprising that the F1 score of the models increased from 0.95113 (batch size = 1) to 0.97135 (batch size = 4) and then falls back to 0.94911 (batch size = 16).

Therefore we decided to use a batch size of 4 and the parameters with the best performance within the first ten epochs.

1.3.4 Model 3/4: Last 4 layers of BERT with multi-layers FNN/CNN fine-tuning

The BERT uses the information from the last 2-4 layers to apply a max-pooling and eventually get the last layer. Some information might be lost during this projection process; therefore, feeding the outer four layers to a classifier might improve the F1 score (Alammar, 2018).

The FNN and CNN were used as classifiers for comparison. However, after the performance of the model converged, the performance on the dev set did not improve as expected. The configurations of the neural network we used are from Safay's work (Safaya et al., 2020, #). A batch size of 1 and 512 paddings is used.

Classifier	Ep5	Ep10	Ep12	Ep14
FNN(F1)	88.3775	90.8320	94.6617	95.6513
CNN(F1)	90.0442	92.6667	94.4562	94.4495

Table 4: Comparison between FNN model and CNN model

1.4 Evaluation and Conclusion

The best performances of the models we implemented:

Classifier	LR	BERT+ LSTM	BERT+CNN	BERT+ Multi-layer-FNN	BERT+ Single-Layer FNN
F1	0.84232	0.7916	0.89662	0.91113	**0.97135**
Accuracy	0.90226	0.8026	0.90673	0.92805	**0.98120**

Table 5: Comparison among each model

Based on table 5, the performances of the classifiers obtained from the development dataset, it can be observed that the BERT + FNN with 4 batch size and ten epochs numbers outperform overall. Yet, it only scored 0.90909 on the Kaggle public leaderboard. Meanwhile, the BERT + FNN with 1 batch size and 15 epochs did not perform well when verifying using the development data set, but it achieved a score of 0.91836 in Kaggle, which leads us to our best ranking on the public leaderboard. However, the result in the private leaderboard shows that the best-performed model with the development dataset turned out to be the one that performed the best on the leaderboard with a score of 0.90196. This might be caused by the applied dropout rate, which avoids some neurons playing an excessive proportion. Thus, it is more stable overall.

Several factors might cause such inconsistent performance of models in development and test dataset. Firstly, the sizes of training and development datasets are limited, which may result in the overfitting of models. Secondly, the distribution of the test dataset might differ from the development dataset. Thus, it underperforms when scoring with the test dataset.

Considering the gap in performances using different verifying data sets, we decided to use the classifier with the best performance over the development dataset for Task 2. This is because the performance of a model verified with a known labelled dataset would be more controllable.

1.5 Limitation

Several limitations may exist in this task. First of all, our best-performed models all involve the use of BERT. However, BERT is pre-trained with the BooksCorpus and English Wikipedia (Devlin et al., 2018., pp.5), which may not fit the expression of tweets as they can be more conversational and casual. Secondly, although we are using pre-trained models, it still requires fine-tuning with different parameters to achieve the best result. Since the number and performance of available resources such as GPU limit our attempt in different parameters. For example, a BERT model with 1 batch size and ten epochs takes about an hour to fine-tune with a train set size of around 1,000.

Task 2

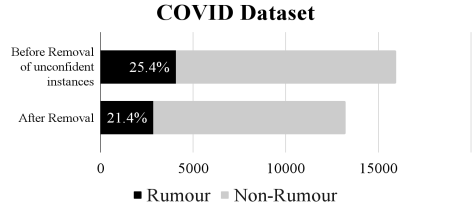
2.1 Data Structure

Given the 254,681 tweets id provided for this task, we managed to retrieve 202,058 of them via Twitter API, the same reason as Task 1 for the missing tweets.

We also removed the instances with absent source tweets. The total number of instances we used for this task is 15,959 (out of 17,458 originally).

The trimmed dataset was subsequently fitted into the BERT + single layer fine-tuning model developed in Task 1, and we got 2,833 events labelled "rumour" and 10,411 events labelled "non-rumour".

Moreover, we filtered out the instances with confidences lower than 0.95 (1,715 events were removed from our dataset). The final dataset we are using for the analysis task is as follows.



2.2 Analysis

2.2.1 Topic Analysis

Latent Dirichlet Allocation (Blei et al., 2003) is used to extract the topic from the source tweet. The preprocessing techniques are lemmatization, stemming, non-alphanumeric character removal, stop words removal, and short words removal(Since lemmatization produces some human non-readable short words, it's hard to interpret them).

2.2.1.a Rumour Topic Analysis

Overall, there are five topics that are meaningful out of 9 extractions.

1. Precaution

Some sorts of words like[Could, prevent, breaking, doctors, spread, positive, covid] are shown in this topic. It's possible that those rumours are about providing fake precautions.

2. Immigration-related Topics

This topic involves some countries' names, for example, US and England. Also, some terms like “attend” and “millions” are in this topic. This rumour topic might be about people fleeing from stricken regions to overseas.

3. Medical Treatment Related Topics

Some hot words from these two topics are [capacity, death, Texas, hospital, death]. This rumour might be the death rate is high, and the hospital somewhere is out of its capacity.

4. President of the United States Topics

[blame, state, explain, think]; Most of the words are the reaction to his speech.

2.2.1.b Rumour Topic Trends Analysis

We chunked the rumour tweets on a 1-month scale producing six individual groups. Extract 9 topics from each of them. In the beginning(months 1-2), the rumour topics involve [China, Virus, Wuhan, arrest, protest]. Then (Months 3-4), most of the rumour topics concern how to stop this pandemic from both federal government and individual perspectives. The topics are[government, leadership, trump, protect, Fauci, health]. In the meantime, the spreading is exaggerated. We can see some topics have keywords like [millions, Indian, England]. In the last stage(month 5 - month 6), the top 3 rumour topics are [1. death, count, report, hospital, positive],[2.reopen], [protesting, black].

2.2.1.c Rumour and non-rumour differences overall

There are five topics out of 9 containing words like[doctor, scientist, expert, science, American today] which may mean those tweets are quoting from some sources. Three topics like [reopen the school, impact child, patient] are quite formal and might be from some news account.

Trend

Unlike the variance in the rumour topic, the non-rumour tweets are quite stable. Some keywords like[China, US, covid-19, lockdown] can be seen constantly over this 6 months period.

2.2.3 Hashtag Analysis

We retrieved the top 100 frequently used hashtags for both rumour source tweets and non-rumour source tweets and combined the hashtags with the same meanings.

Hashtag Examples of Rumour Source Tweet

Topics	Hashtags example	# of hashtags
COVID_origins	#Coronavirus, #CoronaviruspPandemic #CovidConversations #WuhanVirus #WuhanCoronaVirus	141
President of US	#Trump #Trump2020 #TrumpMeltDown #TrumpResignNow #President #TrumpIsAnIdiot #ObamaWasBetterAtEverything	25
Region/Country	#China #Iran #Russia #Beijing #Wuhan	13

Hashtag Examples of Non-Rumour Source Tweets

Topics	Hashtags example	# of hashtags
COVID_origins	#Covid19 #Coronavirus #Covid #CoronaVirusPandemic #Pandemic #CoronaVirusRussia #Corona #Covid19Insa #CoronvirusLockDown #Covid19SA #Covid19UK #CovidSafe	1,648
Scientific facts	#StayHome #TogetherAtHome #WashYourHands #CovidSafe #SafeHands #WearAMask	112
Region/Country	#China #India #Florida #Nigeria #Insa #Italy #UK #Taiwan #NewZealand #Texas	92

Note: the above tables are based on the top 100 frequently-used hashtags of each class instance.

With the analysis of the frequently used hashtags of each class, we have found that both classes tagged variant formats of “COVID-19” at most. An interesting finding is that the Rumour class prefers to use #Coronavirus (52.48% of the COVID origins hashtags) over others (#Covid19, 43.26%), while Non-rumour class tweets prefer to use #Covid19(67.72%) over others (#Coronavirus 27.61%)

We also realised that a significant proportion of rumour-tweets tagged with Trump-related topics(3.67%), while this is a tiny fraction (0.48%) in non-rumour tweets class.
In contrast, the tweets in non-rumour classes tend to hashtag more with the scientific facts(3.90%) compared to the tweets in rumour classes (1.27%).

Meanwhile, the non-rumour tweets use more various location-related hashtags compared to rumour tweets, with the hashtags of rumour tweets concentrated in #china, #iran, #russia… and non-rumour tweets distributed in other countries/regions like #floria, #italy, #taiwan, etc.

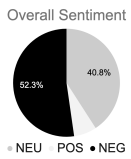
2.2.4 Sentiment Analysis

For sentiment Analysis of each class, a Toolkit from Pysentimiento (Pérez et al., 2021) was employed which returns a dictionary of probabilities that a tweet text is neutral, positive, and positive. We labelled the tweets with the sentiment with the highest probability.

The overall distribution of sentiments towards COVID-19 reflected by the combination of source tweets and reply tweets is demonstrated in the chart below:

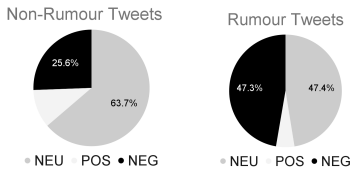
Overall Tweets Sentiment Distribution

NEU	48,985
POS	8,283
NEG	62,745



It is not surprising that most of the COVID-related tweets are labelled “negative” which is a good indicator of the public's negative attitude towards COVID-19.

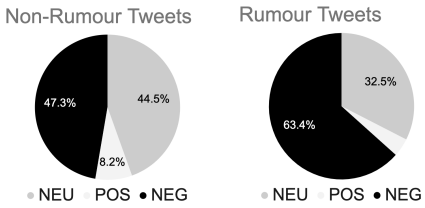
Source tweets Sentiment Distribution



Source Tweets	NEU	POS	NEG
Non rumour	6,637	1,111	2,663
Rumour	1,344	149	1,340

Compared to the tweets that are labelled as non-rumour, it is shown that the rumour tweets are more likely to deliver a negative sentiment versus positive sentiment. In contrast, the proportion of neutral rumour tweets is a lot lower than that of neutral non-rumour tweets.

Reply Tweets Sentiment Distribution



Reply Tweets	NEU	POS	NEG
Non rumour	32,616	5,980	34,664
Rumour	11,910	1,494	23,252

Compared to that of source tweets, both proportions of a negative class of non-rumour tweets and rumour tweets increased by around 20%. Within the positive class proportion remains, less neutral sentiments are delivered with reply tweets for both rumour tweets and non-rumour tweets

2.2.4 User Image Analysis

To observe the characteristics of rumour-creating users, we have extracted several attributes from the source tweets, including the number of followers and friends, whether the user is verified, and the interval between account creation and tweets sent time. We calculated the followers over friends ratio, which may show the authenticity of the user. The lower this ratio indicates the user would be more likely to be fake. Because fake users can have more friends than followers. (Castellini et al., 2017, pp.198) The following four pie charts show the number of users within several ranges of this ratio. The ratio lower than 500 has been ignored since the lower ratio contains less information.

Chart 1 and Chart 2 show this ratio among non-verified users. It can be seen from Chart 1 that there are 76% of users who sent rumour tweets lies in the lowest 500-2000 range, while this downs to only 69.8% for non-rumour tweets senders.

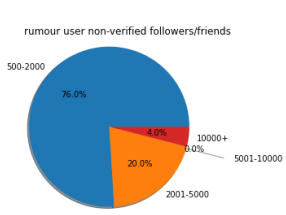


Chart 1

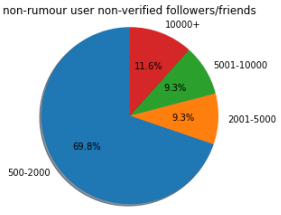


Chart 2

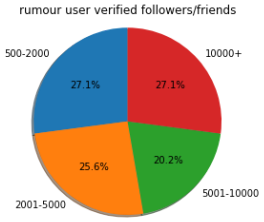


Chart 3

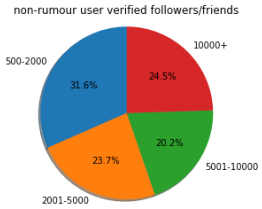


Chart 4

However, Chart 3 and Chart 4, which represent the distribution of this ratio among verified users, do not show such apparent gaps among the ranges. This might be caused by the verification that Twitter has filtered a large number of fake users.

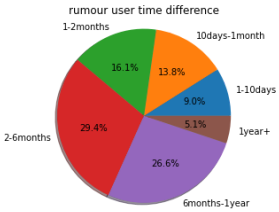


Chart 5

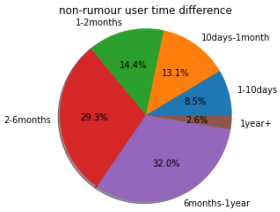


Chart 6

Chart 5 and Chart 6 show the time interval between the account creation time and the tweet send time. Both rumour and non-rumour tweet senders have a similar distribution. But it can be observed that there are 5.1% of the rumour creators with an account that has existed for over a year, while only 2.6% for non-rumour senders. We have also observed that 100% of verified rumour senders are within 2 to 6 months of the time interval between creating the account and sending.

Based on our comparisons of the extracted features, most of the rumour creators are with fewer friends, followers, and favourites, and are mainly not verified. However, a similar distribution of user features also exists among non-rumour senders. In fact, although existing some noticeable facts, there are no apparent differences between rumour creators and normal users among the given data set. In a word, user-based features may not be a suitable choice in the task of rumour detection. (Enayet & El-Beltagy, 2017, pp.473)

References List

Alammar, J. (2018, December 3). *The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)*. Jay Alammar. Retrieved May 12, 2022, from http://jalammar.github.io/illustrated-bert/

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research 3*.

Castellini, J., Poggioni, V., & Sorbi, G. (2017, August). Fake Twitter followers detection by Denoising Autoencoder. *Proceedings of the International Conference on Web Intelligence*, 195-202.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 16.

Enayet, O., & El-Beltagy, S. R. (2017, August). NileTMRG at SemEval-2017 Task 8: Determining Rumour and Veracity Support for Rumours on Twitter. *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, 470-474.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). arXiv. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, *1907*(11692v1).

Pérez, J., Giudici, M., C., J., & Luque, F. (2021). arXiv preprint arXiv. *pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks.*, *2106*(09462).

Qiao, Y., Xiong, C., & Liu, Z. (2019). Understanding the Behaviors of BERT in Ranking. *arXiv preprint arXiv*, *1904*(07531). https://arxiv.org/pdf/1904.07531.pdf

Safaya, A., Yuret, D., & Abdullatif, M. (2020). KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. *Proceedings of the 14th International Workshop on Semantic Evaluation*, *14*(Barcelona, Spain (Online)), 2054–2059.