A Study of COVID-19 Misinformation on Twitter

COMP90042 Project Report - 1047538

Abstract

With the growing communication via online media, the ability to filter between news and fake news has becoming more important. This paper aims to build a rumour classifier using current SOTA, BERT. The classifier then will be used to classify COVID-19 related tweets into rumour and non-rumour. Further analysis is performed, such as topic extraction to understand better the difference between classes. It was found that non-rumour tweets have more diverse conversation and more positive tone when compared to rumour tweets.

1 Introduction

The COVID-19 pandemic has caused us to change the way we live. With restrictions in place, social media is one of our main ways to connect with each other and consume news. According to a survey in 2020, 53% of U.S. adults often receive their news from social media [1]. This shows the importance to have news that spread on social media to be valid.

However, due to its open-access nature and minimum supervision, misinformation often occurs in social media. One way to help reduce this is by using an automated rumour classifier that can detect rumours. This project aims to build such classifier and implement it on a COVID-19 related dataset for a further investigation on the characteristics of rumours. This project is completed using a Twitter dataset.

2 Methodology and Data

The aim of this project is to create a rumour classification which then will be used to predict COVID-19 related tweets that enable us to analyse further what makes a tweet *rumour* or *non-rumour*.

Dataset	Total Count	Rumour	Non-Rumour
Train	4,641	1,583	3,058
Dev	580	187	393
Test	581	NA	NA

Table 1: Rumour classifier data distribution

The structure of the dataset comprises of source and reaction tweets. The metadata information of each tweet can be read further on Twitter API documentation [2].

2.1 Rumour Classification

The source tweet is the one to be classified with the label l indicating that the tweet is rumour (l = 1) or non-rumour (l = 0). Therefore, the task of the rumour detection can be defined as binary classification problem. The dataset statistics, including train, development, and test set, can be found in Table 1.

Each line in the train dataset contains one source tweet and multiple reaction tweets in form of reply, ranging from 0 (no reply at all to source tweet) to 345. This information is sufficient to be a feature that can be focused on. First intuition is that a vectorized words from combined text of source and reply tweets can be built and fed to logistic regression model.

All of the work done by using Google Colab [3] with help of using GPU capability for neural network model using PyTorch [4].

2.2 Logistic Regression

Logistic regression is a probabilistic discriminative model that uses a logistic function and assumes a Bernoulli distribution to transform a linear problem into a classification [5]. The text first is vectorised into a bag of words representation using ScikitLearn's [6] CountVectorizer with further pre-processing using TfIdfTransformer to provide a matrix of counts. The transformed data fed to the logistic regression model using the default parameters from ScikitLearn. The trained model is tested against the development set to produce the accuracy and used as the final model for the rumour classifier.

2.3 Multi-layer Perceptron (MLP)

The predictive capability of MLP often outperforms logistic regression due to the non-linear nature that able to model complex boundaries. Using the default model of ScikitLearn's MLPClassifier, the data, processed the same way as Logistic Regression, is fed to the

MLP model and tested against the development set to calculate the accuracy.

2.4 Bidirectional Encoder Representations from Transformer (BERT)

BERT is a language representation model designed to pre-trained deep bidirectional representations from unlabeled text that can be fine-tuned in downstream by just adding an additional layer [7]. This model will be the focus of the rumour classification task.

The pre-processing phase can be divided into three iterations: baseline, with cleaning, and using max match algorithm for the hashtag. In general, the train dataset information defined in a class called PreprocessDataset taking Dataset class from torch.utils.data. The dataset (in pandas dataframe), labels, and AutoTokenizer from 🤗 (huggingface) Transformers [8] are initialised. The AutoTokenizer loads the 'bert-base-uncased' which then will be used to tokenise sentences into BERT tokens. Acknowledging the feature of sequence-pair classification, the tweets are split into source and reply tweets. AutoTokenizer is then used to get the input sequence, attention mask, and the segment id tokens that indicates source and replies portion. All the replies are concatenated using one space. Padding is defined to '*max_length*', set at 512 to match the input specified by BERT model. Tweet tokens more than 512 by length will be truncated by force. The output of AutoTokenizer then will be converted to PyTorch tensors alongside with the labels encoded with 0 if it is *non-rumour* and 1 otherwise.

No data cleansing done in the first iteration. In the second iteration, Twitter handle (words starting with '@') and URLs are removed. The hashtags (words starting with '#') are extracted by removing only the '#'. In the third iteration, max match algorithm is used to the hashtags with adding the '#' back to the sentence.

The max match algorithm uses a combination of NLTK [9] corpus, such as words, brown, and reuters. The lemmatization method uses NLTK's WordNetLemmatizer. Some hashtags contain words that are not yet in the combined corpus, so two words, i.e., 'hebdo' and 'wiki', are added manually to the corpus. In terms of deciding between max match and reversed max match for the hashtag, a unigram model with laplace smoothing is used to calculate the probability in log space with respect to the combined corpus.

Hyperparameters	#
Batch size	16, 32
Learning rate (Adam)	5e-5, 3e-5, 2e-5
Number of epochs	2, 3, 4

Table 2: BERT recommended hyperparameters.

All of the preprocessed dataset will go into DataLoader class from torch.util.data to be fed into the classifier.

The classifier comprises of a BERT layer and a single feed forward network. The BERT layer use BERT model that loads the base pretrained model (uncased). All of the information loaded in DataLoader are fed to the BERT model, resulting in a contextualised embedding of [CLS] that can be passed to a feed forward network with input dimension of 768 and a single scalar output determining the class of the tweet. The training phase use *cuda* feature to harness GPU computational power of Google Colab.

Binary cross-entropy with logits (sigmoid) is used for the loss function. Adam is chosen as the optimiser due to how well it performs on very large data such as text analysis [10]. Other hyperparameters chosen as recommended by BERT authors (Devlin, et. al.) (Table 1), i.e., learning rate and number of epochs are set at 2e-5 and 4, respectively. Due to Google Colab GPU limitation to process the data, batches of 8 is used instead.

After all batches have been trained, the trained model is tested against the development set. The best development accuracy from all epochs is used for the final classifier.

2.5 COVID-19 Twitter Data

The dataset for COVID-19 Twitter contains 17,453 rows with 237,223 total replies. Figure 1 shows the volume of COVID-19 related tweets over time.

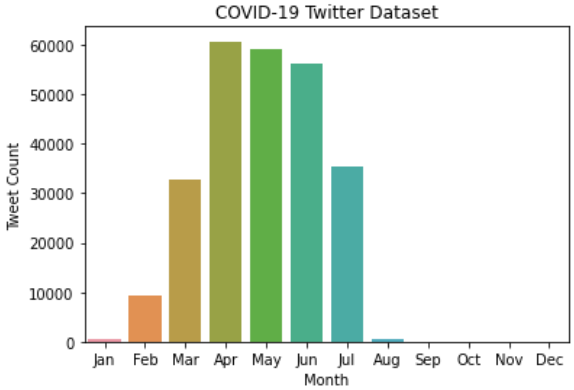

Figure 1: COVID-19 Twitter dataset by month

Model	Acc (Dev)	F1	Recall	Precision
Logistic Regression	0.8275	0.6914	0.5957	0.8235
Multi-layer Perceptron	0.8431	0.7486	0.7128	0.7882
* BERT raw text	0.8741	**0.8265**	0.8617	0.7941
BERT raw text (re-train)	0.8972	0.8063	0.8191	0.7938
BERT /w PP	0.8948	0.7960	0.8511	0.7477
* BERT /w PP & MM	0.8715	**0.8187**	0.8404	0.7980

Table 3: Comparison of different Rumour Classifier with * indicating used as the final model

We can see from the figure that most of the COVID-19 related tweets fall between March to June 2020 which can be linked to the declaration of pandemic by the WHO [11].

Pre-processing is similar with cleansing method done in the second iteration on section 2.4, only the hashtags is now being kept for further analysis. NLTK's TweetTokenizer is used for tokenising the tweets with consideration of removing English stopwords downloaded from NLTK. Some stopwords, such as 'covid19' and 'coronavirus' were added manually to the stopwords list in order to better understand the conversation of the tweets.

The focus of this task will be text analysis. One of the methods used for the analysis is Latent Dirichlet Allocation (LDA). LDA is a generative probabilistic model that can calculate probabilities of text corpora and produce similarities which then can be bucketed into clusters of common topics [12]. The LDA library is using both from ScikitLearn and Gensim [13], which on the ScikitLearn, GridSearchCV class is used to find the optimum number of topics. When using Gensim, a coherence method using CoherenceModel is used to determine the optimum number of topics.

LDA model required a bag of words representation as the input. The data are divided based on the corresponding class (*rumour* or *non-rumour*). The text data fed into ScikitLearn LDA is vectorised using ScikitLearn's CountVectorizer, while the input data for Gensim LDA is converted using Gensim's Dictionary class. Only the top 15 words from each topic resulting from LDA model will be taken into analysis.

Since the class of the COVID-19 Twitter data is using the classifier built in this project, room for error will be assumed when analysing *rumour* and *non-rumour* COVID-19 related tweets.

3 Results and Analysis

Notice that there are two BERT model using raw text shown in Table 3. It is unexplainable that different training session yield different results. This might be caused by the complexity of the unsupervised model inside BERT which results in its "black box"-like behaviour. Despite that, the difference is not too big, so in this report, the BERT (re-train) result is used to justify the comparison of result between different models.

3.1 Rumour Classifier

The performance comparison of each models and pre-processing method (in BERT case) detailed in Table 3.

Logistic regression act as a baseline model in this task. With minimal processing, logistic regression has already had a relatively high accuracy when tested against development set. Although it shows some overfitting problem when tested against test set. With the same data structure, MLP shows some improvement, and finally BERT as the current SOTA scores the highest. This is due to the capability of BERT extracting contextual representation of the tweets between source and reply that makes BERT able to classify better between *rumour* and *non-rumour* tweets.

As this is expected, more pre-processing techniques are done to the train dataset when feeding to BERT. Doing some data cleansing make BERT model perform lower than the raw data. Upon inspection, this was likely due to the hashtag removal and poor tokenisation done by the AutoTokeniser. One example is for tweet "@Heresy_Corner #ImCharlieHebdo" when cleaned become "ImCharlieHebdo" and tokenised into "[101, 10047, 7507, 12190, 2666, 5369, 2497, 3527, 102]". This is not what intended as there should be at max 4 tokens so that BERT can read the contextual representation of the tweet. To tackle this problem, max match was used when encounter hashtags. The result become higher than other models. Therefore, the model used for classifying COVID-19 tweets related data is BERT with pre-processing and max match.

	Rumour	Non-Rumour	Total
# of records	812	16,646	17,458
avg replies per record	9	14	-
# of hashtags	5,158	175,758	180,916

Table 4: Distribution of predicted COVID-19 data.

3.2 Topic Analysis

The classifier predicted only 4.65% of the total records as *rumour* as detailed in Table 4. It is also interesting to see that the average number of replies for tweets classified as *rumour* is lower (9) when compared to *non-rumour* tweets (14).

Another interesting result is when using the log likelihood score as the metrics, 5 topics was chosen as the most optimal number of topics (Figure 2). In contrast, a considerably higher number of topics that considered as optimal when using coherence as the metrics, which is 30 topics (Figure 3). We will discuss both the LDA result from ScikitLearn and Gensim.

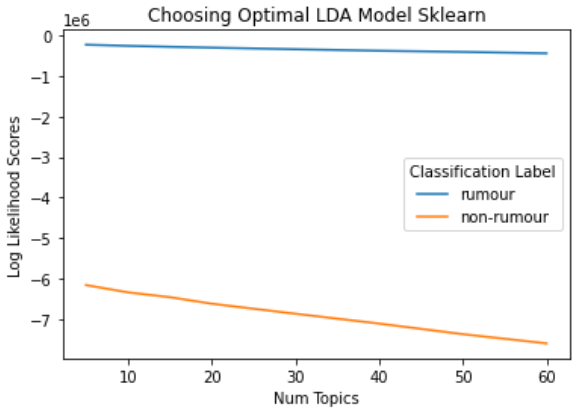

Figure 2: Log-likelihood scores in choosing optimal number of topics for ScikitLearn's LDA.

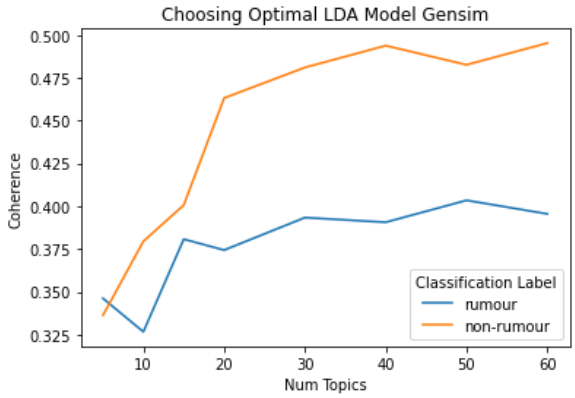

Figure 3: Coherence scores in choosing optimal number of topics for Gensim's LDA.

Mutual Key Topics	Rumour Keywords	Non-Rumour Keywords
COVID-19 Announcement	'pandemic', 'cases', 'deaths', 'tested', 'positive'	'India', 'mask', 'please', 'wear', 'one'
China	'#wuhanvirus', 'outbreak', 'chinese', 'breaking'	'chinese', 'wuhan', 'flu', 'world', 'people'
America	'trump', 'president', 'american', 'response', 'war'	'trump', 'cases', 'country', 'states', 'fauci'
Healthcare	'get', 'positive', 'tested', 'going'	'patients', 'nursing', 'help', 'homes', 'money'

Table 5: Common related topics from LDA produced by both ScikitLearn and Gensim.

The result from ScikitLearn shows a promising distinction between *rumour* and *non-rumour* (Table 5). We can see from the table that there are different keywords, for example related to COVID-19 announcement, on the *rumour* side people tend to talk about 'cases' and 'deaths' where on the *non-rumour* side people are talking about wearing mask and cases in India.

The increased topic size in Gensim LDA make the keywords richer. Distinct keywords, such as 'vaccine', 'fauci', 'riots' and 'floyd' in the *non-rumour* class shows that *non-rumour* class tweets are more related to news, when compared to *rumour* class tweets. The *rumour* class tweets also tend to show negative sentiment, such as '#bailouthumans', 'lost', 'job'.

The takeaway from this result is that coherence when used as a metric in determining the most optimum number of topics yields better result than the log-likelihood in terms of variety of conversations and distinct tweets between classes.

4 Conclusion

We have discussed different Rumour Classifier models and its performance. We also have explored COVID-19 related Twitter dataset using the built classifier and gain insights that characterised *rumour* and *non-rumour* tweets.

References

[1] Shearer, E. and Mitchell, A., 2021. News use across social media platforms in 2020.

[2] Twitter, Inc. (2021). Twitter API Documentation. Available at: https://developer.twitter.com/en/docs/twitter-api (Accessed: 14 May 2021).

[3] Bisong, E., 2019. Google colaboratory. In Building Machine Learning and Deep Learning Models on Google Cloud Platform (pp. 59-64). Apress, Berkeley, CA.

[4] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. and Desmaison, A., 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

[5] J. S. Cramer, "The origins of logistic regression," 2002.

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, pp.2825-2830.

[7] Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[8] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M. and Davison, J., 2019. HuggingFace's Transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

[9] Loper, E. and Bird, S., 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

[10] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[11] Cucinotta, D. and Vanelli, M., 2020. WHO declares COVID-19 a pandemic. *Acta Bio Medica: Atenei Parmensis*, *91*(1), p.157.

[12] Blei, D.M., Ng, A.Y. and Jordan, M.I., 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, *3*, pp.993-1022.

[13] Radim Rehurek, and Petr Sojka 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45–50). ELRA.