COMP90042 Project 2021:
Rumour Detection and Analysis on Twitter

1110298

Abstract

The exponential spread of news information through social media has become popular leading to rapid emergence of rumours (C P & Joseph, 2019). This was undoubtedly observed during the Covid-19 pandemic as many relied on social media platforms like Twitter for regular news updates during the times of uncertainty. Thus, this project aims to detect misinformation amongst noteworthy news events in contrast to reliable information using various methods of Natural Language Processing (NLP) on twitter datasets. Approaches for supervised binary rumour classification included logistic regression, naive bayes, feed forward neural networks, LSTMs, BERT and BERT ensemble models. This report shows that the BERT ensemble model performed better and is hence used to carry out a comprehensive analysis of Covid-19 twitter data to derive interesting insights about the characteristics of misinformation spreaders, popular rumour topics, hashtags and sentiments.

1 Introduction and Related work

Rumour is defined as information that has untrustworthy sources and may lead to confusion, anxiety, reduced trust in government, public panic and violence (Yadav & Purohit, 2019). As novel breaking news appear on social media, specifically in popular microblogs like Twitter, automatic rumour detection becomes a challenging task in NLP due to indistinguishable fake vs real information and user properties (Hamidian & Diab, 2015). This paper attempts to address this problem by using twitter metadata like tweet text, followers count, reply tweets to create a rumour classification model. It is divided into 5 sections starting from motivation and related work. Section 2 describes the various methodologies used for rumour detection, characteristics of labelled rumour data and preprocessing techniques for Covid-19 twitter data for analysis. Section 3 discusses the results of rumour classification and presents detailed analysis insights of rumours in Covid-19 test dataset. Error analysis and conclusion is presented in section 4 and 5 respectively.

(Alzanin & Azmi, 2018) perform a survey on supervised, unsupervised and hybrid approaches to detect rumours in Twitter where significant feature engineering was performed to build a classifier that automatically decides if a topic is related to a newsworthy event and performs credibility check. The 4 feature groups are message-based, user-based, topic-based and propagation-based features with manually annotated NEWS, CHAT and UNSURE labels to classify a dataset sample. Support Vector Machines (SVMs), decision trees and Bayes networks were used, 86% accuracy was achieved using J48 decision trees. The work by (C P & Joseph, 2019) show how Recursive Neural Networks (RvNN) perform significantly better than other neural networks like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). This is because CNN and RNN were content based consisting of just source and reply tweet contents whereas RvNN was not only used to capture the content semantics of tweets but also the responsive tree propagation structure.

2 Methodology and Data

To identify the best method for rumour detection, several algorithms were explored utilizing content based information of source and reply tweets. This included the following:

2.1 Labelled Rumour and Covid-19 data

The datasets provided for this task are already split into train, validation and test datasets which are json list (jsonl) files with each instance consisting of a source and a bunch of associated replies. Each source tweet can be classified as a rumour or non-rumour and labels are provided for the training and validation sets with tweet ids in a json file format. A summary of the health check or statistics of the provided data is in Table 1 which shows that the datasets are imbalanced as the size of non-rumours is nearly double the rumour size.

Data	#source	#src+rep	#rumor	#non
Training	4641	81120	1583	3058
Validation	580	10546	187	393
Test data	581		N/a	N/a
Covid-19	17458	254981	N/a	N/a

Table 1 Health Check of provided datasets

2.2 Preparing the datasets

In order to extract tweet text and other useful information from the jsonl and json files, they are loaded into a python script and converted into 'Data Frames' using the *json* and *pandas* libraries from python. The data frames are combined based on tweet ids to label each source tweet, the labels are one-hot encoded. Then, for every source tweet, the reply tweets are extracted based on **in_reply_to_status_id_str;** sorted based on time using ***created_at*** to form a new column in the data frame where each instance is a list of reply tweet texts sorted by time. This is achieved using python's *re* and *datetime* libraries. Finally, other numeric and binary metadata such as *followers_count* and *default_profile* is extracted to the dataframe for analysis in Section 3. Minimal preprocessing was done as this task has heavy reliance on sentence structure and context which preserved these elements. Hence, stemming, lemmatization and lowercase operations are avoided; only tags such as whitespaces and newlines are removed.

2.3 Rumour Detection and Classification

The goal of this task is to focus on detecting rumours and its associated properties. The accuracy is calculated for each model and predictions are passed to eval.py script to calculate F1-score, precision and recall.

Logistic regression: This was chosen as the baseline model as its performance was better than other machine learning models and even FFNNs and LSTMs. This was proved by the codalab scores for the unseen test instances scoring 82% with just this baseline model. This baseline model was trained, validated and tested using source tweet text and source+reply tweet texts where the replies are sorted by tweet creation time. Tokenization was performed and a Bag of Words (BoW) approach was used to fit the classifier (Wang, Li, Wang & Zhang, 2019).

FFNN Firstly, a simple FFNN was designed with three layers with a BoW input with 10 neurons in the hidden layer, a relu non-linear activation function and an output layer with sigmoid activation to predict probabilities. A second FFNN model was implemented where input was word embeddings obtained using cosine similarity, tokenize to word sequences and post 'pad' to have sentences of the same length. Thus, flatten function is added after the embedding layer and a similar architecture is followed for the hidden and output layer as model 1. An Adam optimiser, binary cross entropy loss and accuracy is used for evaluation.

LSTM The layers in LSTM resonate to the second model of FFNN with an additional LSTM layer and use similar hyperparameters.

BERT BERT stands for Bidirectional Encoder Representations from Transformers developed by Google in 2018. Here, pre-trained BERT based uncased is used and the model is first trained using just the source tweets and then the source+reply tweets in the form [CLS] source [SEP] reply1 [SEP] reply2 where CLS and SEP are classification and separator tokens respectively. BERT requires GPUs and thus the GPUs in Google Colab are used. Then, the transformer library of huggingface is installed that contains Pytorch Implementations of BERT and pre-trained model weights. After that, the encoder_plus method splits text into tokens, converts them to indices, pads to max_len and creates attention masks. A Pytorch dataloader, BERT-base uncased model is used consisting of 12 transformer layers and the output of the final layer CLS token is fed into the classifier consisting of a single hidden layer. The BERT classifier is initialised with an Adam optimiser and 2 epochs and a batch size of 10 due to memory limitations in codalab. During training, the operations carried out were load data to GPU, zero out gradients, forward pass for logits and loss and backward for gradients, norm-clip, update parameters and learning rate.

The model was saved and ensembles created using various seed values.

3 Results and Analysis

3.1 Rumour Detection and Classification

According to Table 2, the best classification model was BERT Ensemble which had 85% F1 score and performed the best with Codalab test set with a score of 0.84 thus outperforming the baselines. The BERT ensemble model was trained with seed values of 42 and 86 and a 50% threshold for predictions. This was trained with the source tweet text concatenated with the reply tweet text sorted by creation time. The performance metrics for other models significantly reduced when the reply tweet text was included and hence not included in the table. This was because cosine similarity was used, and dropout was harder to implement in models like LSTMs which is also sensitive to random weight initializations.

Model	**Acc**	**F1**	**Pr**	**Re**
LR Baseline	.87	.78	.85	.72
NB	.79	0.76	.86	0.67
FFNN	.85	0.77	.78	.76
FFNN + Embeddings	.86	.77	.80	.74
LSTM + Embeddings	.67	0	0	0
BERT	.87	.79	.80	.79
BERT(+ unsort_rep)	.87	.80	0.81	.78
BERT (+ sort_reply)	.89	.83	.82	.83
BERT Ensemble	.88	.85	.80	.90

Table 2 Dev set evaluation metrics for the models

BERT is trained in Wikipedia and Book Corpus. It is used to capture contextual representations using the concept of transfer learning transformers, multi head attention layers and positional encoding. This was used since RNNs are slow to scale to large corpora as they process one word at a time and in models like ELMo, the forward and backward language models are trained independently which provides a surface representation. But BERT uses the Transformer architecture and a masked language model to capture bidirectional contextual representation with a classification layer at the end, the input to which is a [CLS] token. BERT is smarter than the other models to comprehend the sentence structure and requires minimal preprocessing as compared to machine learning algorithms or recurrent networks.

3.2 Hashtag Analysis

As seen from table 3 the most popular hashtags include covid or coronavirus and other generic hashtags for both rumour and non-rumour tweets. Most of the hashtags overlap and there are minor differences such as #onpoli #masstesting in rumours vs #stayalert #staysafe in non-rumours. Hence, it is difficult to distinguish based on hashtags as this is out of domain data.

	#hashtags
Rumours	covid19 coronavirus breaking covid19ph staysafeug trackingandtracing wuhan inthistogetherohio masstesting arizona crushthecurve wearethecure covid19insa covid−19 stayhomeohio stayhome brazil china afc mufc coronavirusinsa watch assamcovidcount stayhomesavelives liveline onpoli lockdown txlege dominiccummings pmqs uganda citinewsroom india covid19sa hyderabad tablighijamaat indiafightscorona mlb
Non-Rumours	covid19 coronavirus breaking china coronaviruspandemic covid− stayhome cdnpoli covid maga lockdown stayhomesavelives trump socialdistancing stayathome trump staysafe blacklivesmatter sarscov auspol stayalert indiafightscorona hydroxychloroquine txlege covid pandemic onpoli florida kag familiesfirst ppe covidiots ridge takeresponsibility wuhan india tcot coronavirussa marr watch

Table 3 Top 40 hashtags

3.3 Sentiment Analysis

As shown in figure 1 where the left refers to non-rumour sentiments and the right refers to rumour sentiments. It is seen that both the sentiments are mostly neutral but non-rumour sentiments are more negative than rumour sentiments. The difference is however marginal. In a similar way emotion could be analysed from emojis to understand the difference between the two sentiment sets.

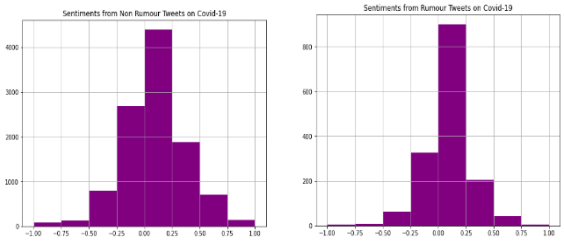

Figure 1: Sentiment Analysis plot

3.4 Topic Analysis

As shown in figure 2 and 3, bigram word cloud and network graphs are generated for analysis and topic modelling was performed as shown in Figure 4 which assigns weights to keywords. It is evident that the rumours talk about new cases and confirmed covid-19 cases that might not be existing, where as the non-rumours talk more about ex-president of the USA Trump, white house and New York as there were highest number of Covid-19 cases and black lives matter in the States then. The network graph for rumours is more densely connected than non-rumours which have clusters.

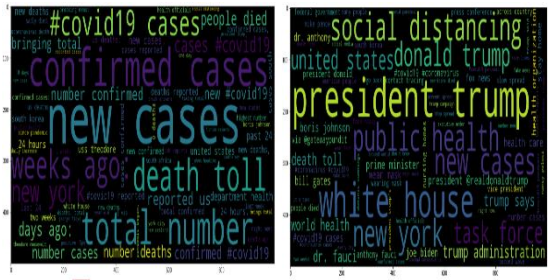

Figure 2 Bigram word cloud

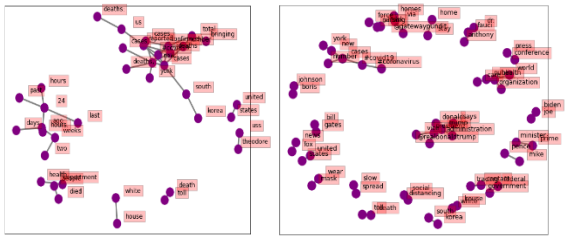

Figure 3 Network graph

```
RUMOUR TOPICS
(0, '0.049*"person" + 0.049*"SCREEN_NAME" + 0.026*"report" +
0.026*"breaking"')
(1, '0.037*"covid19" + 0.025*"mosque" + 0.025*"prayer" +
0.025*"neighbourhood"')
NON RUMOUR TOPICS
(0, '0.034*"coronavirus" + 0.017*"test" + 0.007*"covid-19" +
0.007*"week"')
(1, '0.023*"covid19" + 0.020*"coronavirus" + 0.010*"trump" +
0.010*"first"')
```

Figure 4 Rumour and non-rumour topics

From figure 5 is seen that there were a greater number of rumours on particular days than non rumours over a period of Jan 2020 to Aug 2020 for 2 days this showing the rumour creation patterns.

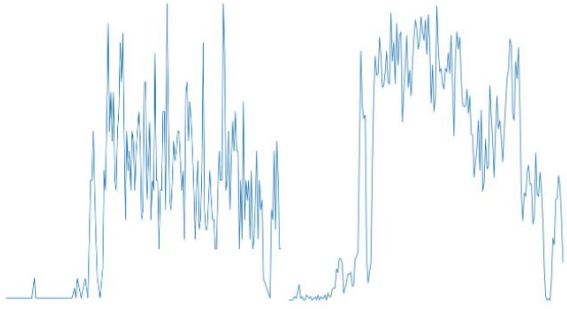

Figure 5 Non-rumour vs rumour tweets evolution

3.5 User Analysis

The median of the numerical values is computed and is shown in Table 4 which shows similar values due to out of domain data. But followers count and statuses count is significantly greater in rumour accounts. But there are lesser retweets, favourites and listed counts for rumours. Binary features were also analysed and there were more default_profiles but lesser had their geolocation enabled.

	Non-Rumour	Rumour
Followers count	387211	408710.0
Followers count	1106.0	1009.0
Statuses_Count	38263.0	59710.0
Listed_Count	2210.0	2143.0
Favourite_Count	2045.0	1305.0
Favourites_Count	3204	2761
Retweet_Count	781	600.0
UAccount Age	598wks,3days	599 wk,2 days

Table 4 User feature analysis

4 Error Analysis

For covid test set it is seen that number of tweets predicted rumours is 1947 using EN_BERT including tweets about vaccinations research in UniQueensland, home-made remedies to improve immunity. For dev set, many tweets that had #ottawa, #sydneysiege and stopblamingMuslims #terrorist were misclassified as rumours and there are 41 such tweets. Figure 6 shows loss plot for BERT used for tuning the model.

Figure 6 Train and Val loss per epoch: BERT

5 Conclusion

This project performed text classification rumours and Ensemble BERT is proved as the best performing model. A rumour analysis is performed on out of domain Covid-19 data to generate interesting insights. Further improvements could be made by using distilled BERT models, handling imbalanced data and also incorporating the propagation structure of tweets and graph convolution networks.

References

Alzanin, S., & Azmi, A. (2018). Detecting rumors in social media: A survey. *Procedia Computer Science*, *142*, 294-300. doi: 10.1016/j.procs.2018.10.495

C P, P., & Joseph, S. (2019). Deep Learning Approach For Rumour Detection In Twitter: A Comparative Analysis. *SSRN Electronic Journal*. doi: 10.2139/ssrn.3437620

Hamidian, S., & Diab, M. (2015). Rumor Detection and Classification for Twitter Data. *SOTICS 2015 : The Fifth International Conference On Social Media Technologies, Communication, And Informatics*.

Tian, L., Zhang, X., & Lau, J. (2020). #DemocratsAreDestroyingAmerica: Rumour Analysis on Twitter During COVID-19. *CEUR Workshop Proceedings (CEUR_WS.Org)*, *2699*.

Wang, S., Li, Z., Wang, Y., & Zhang, Q. (2019). Machine Learning Methods to Predict Social Media Disaster Rumor Refuters. *International Journal Of Environmental Research And Public Health*, *16*(8), 1452. doi: 10.3390/ijerph16081452

Yadav, S., & Purohit, A. (2019). Rumor Detection System for Twitter (A Micro-Blogging Site). *International Journal Of Recent Technology And Engineering*, *8*(4), 12287-12293. doi: 10.35940/ijrte.d4308.118419

Alexander V. Mamishev and Murray Sargent. 2013. *Creating Research and Scientific Documents Using Microsoft Word*. Microsoft Press, Redmond, WA.

Alexander V. Mamishev and Sean D. Williams. 2010. *Technical Writing for Teams: The STREAM Tools Handbook*. Wiley-IEEE Press, Hoboken, NJ.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. *Yara parser: A fast and accurate depen-dency parser*. *Computing Research Repository,* arXiv:1503.06733. Version 2.