Rumour Detection and Analysis on Twitter

Student Number: 1154197

Abstract

In the 21st century, when any kind of information or misinformation travels at a speed faster than lightning, it is very interesting to know what kind of information are people most interested in propagating at the surge of unexpected pandemic COVID-19. Various studies have already been done to capture the spread of rumours through twitter, yet each one had something to contribute to our knowledge. In this paper, we classify a set of tweets as rumour or non-rumour. Various deep learning models are built and used for the classification task. We also present our analysis on the classified tweets. It is not only interesting to know how collectively we as humans react to rumours but it also gives an insight into what genre of topics or what kind of people influence the general opinion of a community.

Introduction

To understand the context of this paper, let us understand the two main aspects of it - firstly, what is COVID-19 and secondly, what is considered as rumour?

COVID-19 is a disease caused by coronavirus [2]. First reported in Wuhan, China in December 2019, the virus is known to cause respiratory infections and is highly contagious. This caused lockdowns and an increased use of social media to stay connected.

Rumours at a very high level can be defined as a piece of incorrect or unverified information (Cai et al. [5]; Liang et al. [6]). As per Oxford Dictionary it is ‘‘a currently circulating story or report of uncertain or doubtful truth’’ (‘‘rumor,’’ n.d.).

Related Work

Most of the work done in rumour detection is supervised, by training models on labelled datasets. (Tian et al. [9]) used Twitter15, Twitter16 [10], PHEME [11], and SemEval2019 [12] as training data to train binary rumour classification models. Although there were 4 labels to the tweets, they were shrunk to ‘rumour’ and ‘non-rumour’ to do the binary classification. Tian et al. ([9]) primarily used BERT model on twitter text and user information of the inceptor of a tweet chain.

Castillo et al. [13]; Yang et al., [14]; Liu et al., [15] used post contents, user profiles and propagation patterns to train their model. Later on Friggeri et al., [16]; Hannak et al., [17] used features such as those representing rumor diffusion and cascades of comments to quash the rumours. Kwon et al. [7] added a time-series-fitting model on the tweets timeline. Ma et al. added more sequenced features. All these works have one thing in common, i.e, heavy preprocessing work. Subsequently, Ma et al. did various works in this field to reduce the heavy work on preprocessing by using Neural Networks. Recurrent NN was one of the most interesting one; however, falling short of the ways to represent how the posts propagated.

Some of the works show use of various other classifiers like SVM by Wu et al. [19] and RvNN(Socher et al., [18]). One of the most recent work to deal with vanishing gradients was done using LSTM integrated onto RvNN (Hochreiter and Schmidhuber, [20]).

One of the most recent papers by Ma et al. [8] proposes the idea of 2 variants of RvNN to capture structural and textual properties. This structure has proven to be very efficient in classification of tweets as rumours or non-rumours and also in early detection tasks.

Problem Statement

Given a set $C$of tweet chains (as described below), classify each tweet chain as rumour or non-rumour.

A tweet chain can be described as $\{r_i \rightarrow x_{i1}, x_{i2}, x_{i3},..., x_{in}\}$, where r_i is the root tweet and $x_{ij}$are the reactions on the tweet; reactions can be a retweet or comment on the tweet.

Datasets

To perform rumour detection on COVID-19 tweets, we have used datasets provided by the University of Melbourne. A decent sized dataset (4,641 tweets) of labelled tweets along with their reactions (re-tweets and replies) are used to train our models. The validity of these trained models are tested over a development set of 580 tweets for fine tuning the models. Finally these tuned models are used to generate labels for 581 tweets and assert their efficiency to detect rumour. Finally these models are applied on 17,458 COVID-19 related tweets to classify them as rumours and non-rumours and do analysis on them for the kind of rumours.

Baseline

A baseline of tweet labels as provided by the University, gave the F1 score of 37.5% on development set.

Pre-processing Data

The data is huge and too many attributes to consider besides text. Initially, only the bag of words in the tweet text was considered for this experiment. Fine tuning the text further, all the *stopwords* in the text were removed. For this, *nltk stopwords* were downloaded, and a match of the languages present in the *nltk* package and tweet language was established. We also used the *tweet-preprocessor* library to do most of the cleaning up tasks.

Moving further, two kinds of text to vector representation of *bag-of-words* was considered; (1) TF-IDF mapping and (2) mapping using *texts_to_matrix* method provided by *keras.preprocessing.text.Tokenizer.* Two machine learning models, namely, Naive Bayes Model and Logistic Regression Model were used to do the binary classification task using both the kinds of text to vector representation. Which gave the following results in terms of accuracy:

	Naive Bayes	Logistic Regression
TF-IDF	66.03 %	79.48 %
Keras *texts_to_matrix*	66.55 %	80.86 %

From the analysis, the idea of using TF-IDF was dropped.

Apart from the above mentioned details, the user features were at times extracted and appended with the pre-processed tweet text (more about it is in the following section).

For this experiment following attributes of tweet chains are used: Text, User id, User verification check, User allow geolocation check and Length of the tweet chain

Deep Learning Models

We used Recurrent NN based models to do the classification task at hand. It has the following structure at the core for processing data.

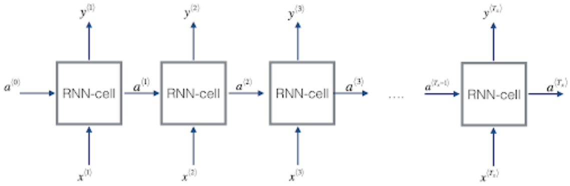


Fig 1. RNN structure

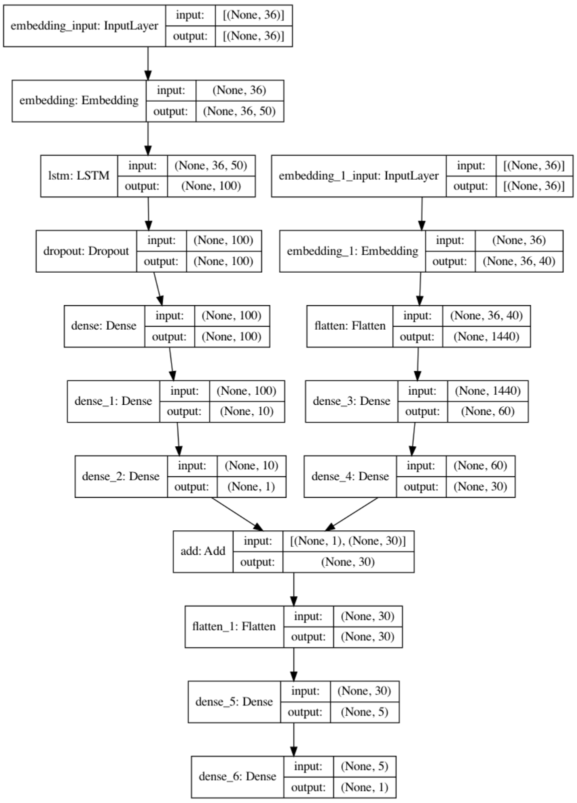
Empirically it has been found that *relu* activation in most of the inner layers enhances the efficiency of the models and since our task is binary classification, *sigmoid* has been our constant choice for the final layer. Also, *binary_crossentropy* has been our primary loss function for all our models.

Because of this simple recurring function, even the simplest feedforward network of just 3 layers with a bag of words representation as count matrix gave an accuracy of more than 80%. In the next step we added 2 more layers to the model and used the text_to_sequence method of keras preprocessing kit to capture the sequential information and it did result in about 87% accuracy with development data.

However, as we know about the vanishing gradient problem that RNN suffers from, for the next step, we decided to classify our data with the LSTM model. This model initially gave a very low accuracy, however, having an *ensembled* model (Fig 2) and using the entire root tweet text gave a better result, 84% accuracy on development dataset. In this ensemble model, we are using two different models, one to work on root text features and the other on user (tweet initiator) features.

Finally, we developed a BERT Model to do the classification task. For this model, apart from the preprocessed tweet texts of the tweet chain we have also considered collective users' validity of each tweet chain. Since the BERT Model only supports 512 tokens, it was a challenging task to fit all data into the Model. As we know the first tweet is the most important tweet for the classification criteria, we can go ahead and truncate any part of later text.

However, we followed Sun et al.[21], who mention an interesting fact that retaining the front and end portion of text for classification with the BERT Model gives the best result. As part of preprocessing for the model, each x_{i1} tweet reaction was concatenated with the root tweet r_i after cleanup, and then '[CLS]' token was concatenated with 256 front tokens and 255 rear tokens.



embedding_input: InputLayer | input: [(None, 36)] | output: [(None, 36)]
embedding: Embedding | input: (None, 36) | output: (None, 36, 50)
lstm: LSTM | input: (None, 36, 50) | output: (None, 100)
embedding_1_input: InputLayer | input: [(None, 36)] | output: [(None, 36)]
dropout: Dropout | input: (None, 100) | output: (None, 100)
embedding_1: Embedding | input: (None, 36) | output: (None, 36, 40)
dense: Dense | input: (None, 100) | output: (None, 100)
flatten: Flatten | input: (None, 36, 40) | output: (None, 1440)
dense_1: Dense | input: (None, 100) | output: (None, 10)
dense_3: Dense | input: (None, 1440) | output: (None, 60)
dense_2: Dense | input: (None, 10) | output: (None, 1)
dense_4: Dense | input: (None, 60) | output: (None, 30)
add: Add | input: [(None, 1), (None, 30)] | output: (None, 30)
flatten_1: Flatten | input: (None, 30) | output: (None, 30)
dense_5: Dense | input: (None, 30) | output: (None, 5)
dense_6: Dense | input: (None, 5) | output: (None, 1)

Fig 2. ensembled model

[The BERT Model was developed and run on Colab Notebooks (by Google), and due to their GPU usage restriction, the batch-size was limited to 20.]

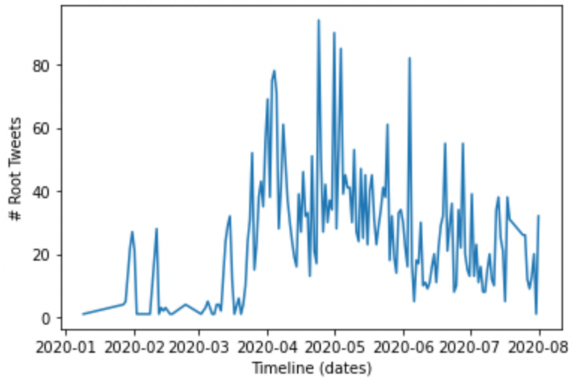
Even with few limitations, the power of transformers were clearly visible, where the bidirectional context is captured very well. One epoch gave a decent accuracy with the development tweet set, and increasing them to 2 gave better results. With the best model we had the following metrics on the development dataset.

Precision	78.46 %
Recall	87.70 %
F1	82.82 %

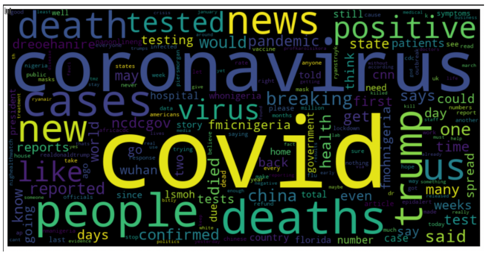
Using the best performing BERT model we decided to do the classification of COVID-19 tweet set.

Analysis

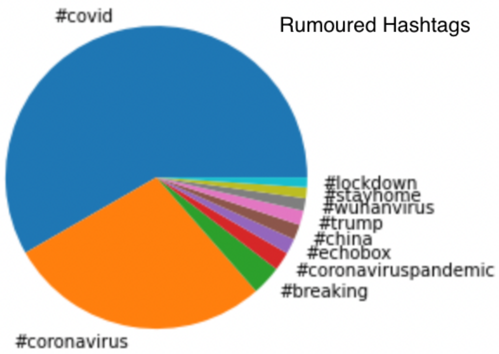
The ratio of rumoured tweets to non rumoured tweets was about (986 : 16471) which shows that there is more true news spread in the community than false news. The following line chart shows the timeline of most active covid rumours propagation.



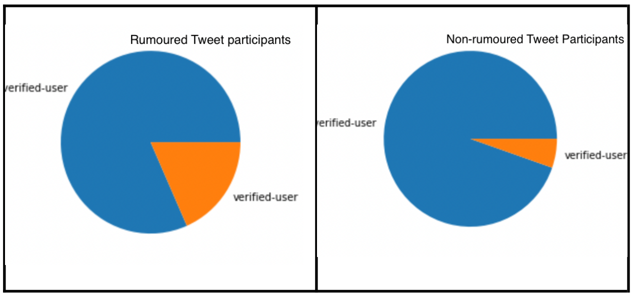
We applied unigram and bi-gram models on preprocessed twitter text to find the most talked about topics. Following word cloud shows the most talked about rumoured topics .



Following pie-charts show the most used hashtags in rumoured tweets.

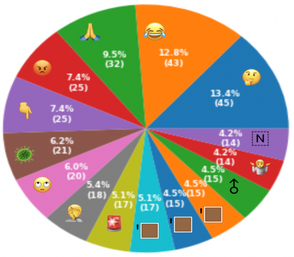


From the hashtags we can say that apart from #covid and #coronavirus, people were much interested in topics related to U.S. president Trump. In rumoured tweets #breaking shows that not every #breaking that we see on Twitter deliver factual news.



After analysing the comparison of verified to non-verified user ratio in rumoured and non-rumoured tweets, we can say that the number of verified users are much less in twitter (in general); and the proportion of verified users propagating false news is much higher than that in propagating verified correct news. This indicates one of the major problems in the society currently, i.e, people not being informed and thus can be swayed away by any news in the market without verifying it and worse say it to other people (by retweeting).

Analysing the most commonly used emojis we found that people were rather confused about the rumours in tweet as the most common emoji was '🤔'. Also, it is interesting to notice how fast twitter came up with coronavirus emoji '🦠'.



We also applied *Rule Based* information extraction methods on rumoured tweets and found following interesting rumours. One of the most interesting 'such as' relation applied with

Noun-such as-Pronoun pattern was "U.S. Senator Kelly Loeffler and her husband dumped millions of dollars in stocks, selling shares in retail stores such as Lululemon and T.J. Maxx and investing in a company that makes COVID-19 protective garments."

Conclusion

We used a number of deep learning machine models to understand their working capabilities and found that RNN is quite robust. We found that transformers in BERT model are capable of retaining bidirectional contextual information for classification

References

[1] - Wani, M., Agarwal, N. and Bours, P., 2020. Impact of Unreliable Content on Social Media Users during COVID-19 and Stance Detection System. *Electronics*, 10(1), p.5.

[2] - Australian Government | Department of Health. 2021. *What you need to know about coronavirus (COVID-19)*. [online] Available at: <https://www.health.gov.au/news/health-alerts/novel-coronavirus-2019-ncov-health-alert/what-you-need-to-know-about-coronavirus-covid-19> [Accessed 8 May 2021].

[3] - En.wikipedia.org. 2021. *Severe acute respiratory syndrome coronavirus 2 - Wikipedia*. [online] Available at: <https://en.wikipedia.org/wiki/Severe_acute_respiratory_syndrome_coronavirus_2> [Accessed 8 May 2021].

[4] - Al-Sarem, M., Boulila, W., Al-Harby, M., Qadir, J. and Alsaeedi, A., 2019. Deep Learning-Based Rumor Detection on Microblogging Platforms: A Systematic Review. *IEEE Access*, 7, pp.152788-152812.

[5] - G. Cai, H. Wu, and R. Lv, "Rumors detection in Chinese via crowd responses," in Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining, Piscataway, NJ, USA, Aug. 2014, pp. 912–917. [Online]. Available: http://dl.acm.org/citation.cfm?id=3191835.3192014

[6] - Y. Liu, X. Jin, and H. Shen, "Towards early identification of online rumors based on long short-term memory networks," Inf. Process. Manage., vol. 56, no. 4, pp. 1457–1467, Jul. 2019. doi: 10.1016/j.ipm.2018.11.003

[7] - S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang, "Prominent features of rumor propagation in online social media," in Proc. IEEE Int. Conf. Data Mining (ICDM), Dec. 2013, pp. 1103–1108. doi: 10.1109/ICDM.2013.61.

[8] - Ma, J., Gao, W., Joty, S. and Wong, K., 2020. An Attention-based Rumor Detection Model with Tree-structured Recursive Neural Networks. *ACM Transactions on Intelligent Systems and Technology*, 11(4), pp.1-28.

[9] - Tian, L., Zhang, X. and Lau, J., 2021. [online] Ceur-ws.org. Available at: <http://ceur-ws.org/Vol-2699/paper32.pdf> [Accessed 10 May 2021].

[10] - J. Ma, W. Gao, K.-F. Wong, Detect rumors in microblog posts using propagation structure via kernel learning, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 708–717.

[11] - E. Kochkina, M. Liakata, A. Zubiaga, All-in- one: Multi-task learning for rumour verifica- tion, arXiv preprint arXiv:1806.03713 (2018).

[12] - G. Gorrell, E. Kochkina, M. Liakata, A. Aker, A. Zubiaga, K. Bontcheva, L. Derczynski, Semeval-2019 task 7: Rumoureval, determining rumour veracity and support for rumours, in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 845–854

[13] - Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In Proceedings of WWW. pages 675–684.

[14] - Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on sina weibo. In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics. MDS '12, pages 13:1–13:7.

[15] - Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time rumor debunking on twitter. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. CIKM '15, pages 1867–1870

[16] - Adrien Friggeri, Lada A Adamic, Dean Eckles, and Justin Cheng. 2014. Rumor cascades. In Proceedings of ICWSM.

[17] - Aniko Hannak, Drew Margolin, Brian Keegan, and Ingmar Weber. 2014. Get back! you don't know me like that: The social mediation of fact checking interventions in twitter conversations. In Proceedings of ICWSM.

[18] - Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In Proceedings of the 28th international conference on machine learning (ICML-11). pages 129–136.

[19] - Ke Wu, Song Yang, and Kenny Q Zhu. 2015. False rumors detection on sina weibo by propagation structures. In Data Engineering (ICDE), 2015 IEEE 31st International Conference on. IEEE, pages 651–662.

[20] - Sepp Hochreiter and Jurgen Schmidhuber. 1997. ¨ Long short-term memory. Neural computation 9(8):1735–1780.

[21] - Sun, C., Qiu, X., Xu, Y. and Huang, X., 2021. [online] Arxiv.org. Available at: <https://arxiv.org/pdf/1905.05583.pdf> [Accessed 12 May 2021].