Rumor Detection and Analysis on Twitter

Anonymous ACL submission

1 Introduction

Twitter is an open platform used for information exchange on which users can freely post and interact through brief messages known as "tweets". Considering the volume of data, rumors, or unverified statements, come part and parcel. According to an MIT study, falsehoods on twitter are likely to spread 20 times faster than fact. [1]

The objective of this project is two parted. The first task is to build a rumor detection model using python. This model was trained and tested on tweets, which were extracted through crawling the twitter API. Three models were developed and tested, each implemented by one of our team member, these models being (1) recurrent neural network (RNN), (2) hierarchical neural network (HRNN), and (3) Bidirectional Encoder Representations from Transformers (BERT) language model. In the second part of the project, the best performing model was selected to label a set of COVID related tweets, and rumour analysis was carried out to understand the nature of COVID rumours.

2 Literature Review

The classification approach to detect rumour veracity can be divided into three groups: machine learning based, deep learning based and propagation-based (Bondielli and Marcelloni, 2019; Cao et al., 2018).

Two of the most widely used deep learning paradigms for rumour detection are Recurrent Neural Networks (RNN) and Convolutional Neural Network (CNN) (Bondielli and Marcelloni, 2019). Due to the inherent structure of RNN, it performs effectively for sequential data and it is reported in (Ma et al., 2016) that RNN-based methods produce more accurate and efficient detection of rumours. CNN-based approaches are also proposed by Chen et al. (Chen et al., 2017) to solve rumour veracity and stance classification for Twitter data. A number of hybrid architectures combining RNN and CNN are also explored in (Ajao et al., 2018; Song et al., 2019).

Propagation-based approaches incorporate the relations among instances and make predictions based on the whole event (Cao et al., 2018). Ma et al. (Ma et al., 2018) proposed a tree-structured recursive neural network which considered the non-sequential propagation structure of tweets. Recent works (Anggrainingsih et al., 2021; Devlin et al., 2018) also proposed methods utilizing Bidirectional Encoder Representation from Transformer (BERT) for rumour detection on twitter.

3 Task1: Rumour Detection

The first part of this project is to develop a model that automatically detects whether a source tweet is a rumour or non-rumour. We are provided with around 2,500 tweet events in the form of the ID of the source tweet followed by the IDs of its reply tweets. Labels indicating whether these tweet events are rumours or non-rumours are also provided. Among these tweet events, around 77.5% of the source tweets are non-rumours and about 22.5% of them are rumours. In order to build a binary classifier for rumour detection, tweet objects associated with each tweet ID are crawled with Twitter API and tweepy. We were able to retrieve most of the tweet objects and in the case where the source tweet is no longer available, we simply removed them from the dataset. Three models were implemented and tuned for this task, which will be described in detail in this section.

3.1 Data Cleaning

Differing from standard texts, Twitter allows people to type emojis, mention other users, and tag external links for reference. However, it increases the difficulty for the computer to understand the

[1] https://news.mit.edu/2018/study-twitter-false-news-travels-faster-true-stories-0308

context correctly. Therefore, before training the model, a series of data cleaning was performed on the texts of the tweets:

- URLs and Twitter handles were removed
- Convert emoji to its textual description
- All characters were lowercased
- Lemmatization was applied
- Punctuations (except for apostrophes) were removed

3.2 Recurrent Neural Network (LSTM)

Each preprocessed tweet was then tokenized with Keras function `Tokenizer()` and was vectorized into a sequence of integers. Each word sequence of tweet text is padded to the maximum number of words in one tweet with 0. Then all the padded word sequences in each tweet event were concatenated which was then padded with 0 to match the length of the longest tweet event. Pre-trained word embedding GloV-twitter-50 was used which is a model pre-trained on 2 billion tweets (27 billion tokens, 1.2 million vocab) [2] Unseen words were assigned with 0 weights.
To deal with class imbalance, different weights were assigned to rumours and non-rumours when calculating the loss. The minority class was weight up proportionally to balance with the majority weight.

When fitting the LSTM model, parameters were tuned by experimenting with different sets of parameters and the model that produce the best performance was found to be: Adam optimizer with the learning rate of 0.008 and binary cross entropy loss, dimension 128 for the LSTM layer, a batch size of 64 and 10 epochs.

3.3 Hierarchical Recurrent Neural Network (HLSTM)

A hierarchical LSTM is also implement. Different from LSTM, the HLSTM has two layers of LSTM. Each word in the tweet was passed to the first (lower level) LSTM layer and the output of the last word of each tweet in the same event was then passed to the second (higher level) LSTM layer.
Parameters were also tuned by experimenting with different sets of parameters and the model that produce the best performance was found to be: Adam optimizer with the learning rate of 0.01 and binary cross entropy loss, dimension 64 for the first LSTM layer and dimension 32 for the second LSTM layer, a batch size of 64 and 10 epochs.

[2] https://nlp.stanford.edu/projects/glove/

3.4 BERT Model

3.4.1 Experimental Setup

The experiment of BERT requires to access GPU, and regarding to that, part of training BERT model have to be implemented on an online server, which providing a specific online programming environment with RTX3090 GPU.

To feed valuable data to pre-trained Bert-base-uncased model, we have to process as following procedures:

3.5 Text Preprocessing

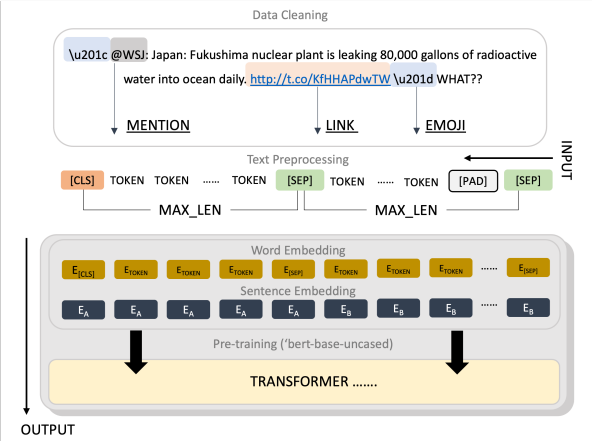

Figure 1: What happens after text preprocessing?

(1) Split the sentence into tokens.(2)Add the special '[CLS]' and '[SEP]' tokens.(3)Add the special '[AUT]' and '[REP]' token to identify whether the sentence is source tweet or not.(4)Add the special '[VER]' and [NON] token to represent the verified status of tweet creator.(5)**Input ids:** apply convert_tokens_to_ids() to Map the tokens to their IDs, the function is based on the 'bert-base-uncased' pre-trained model (6)**Attention Mask:** Create the attention masks which contains a list of binary number, where 1 stands for tokens rather than '[PAD]'.

We did hyper-parameter tuning by grid searching for the maximum f1 score since that is the same rubric for the Kaggle competition. We set the learning rate to be one of [3e-5, 5e-5], choose batch size from a set of [16, 32, 64], and let the number of epochs be four which the author of the BERT model

recommends. To better regulate the real-time runs, connecting to Weights & Bias API helps with mapping logistic loss and accuracy trends when steps moving. In that way, we could directly check exact accuracy values or get line charts for each match attempt (as shown in Table 1). The table illustrates all mix parameters and corresponding f1 scores evaluated by the validation data sets. The highest score is evidence of identifying the best-performance parameters, and then we used them to train the final model. Ensemble fine-tuning works for iterating and developing accuracy, and meanwhile, we get each epoch's outputs and do majority voting to get the final predictions. It is effective to see that the score was getting higher.

	batch_size	epochs	learning_rate	val_accuracy
1	32	4	3e-5	0.9184
2	32	4	5e-5	0.9412
3	16	4	3e-5	0.9338
4	16	4	5e-5	0.9485
5	8	4	3e-5	0.9534
6	8	4	5e-5	0.9478

Table 1: the hyper-parameter tuning outcome

3.6 Model Selection

For the experiments, we are trying to implement three different models. They are the BERT model, recursive neural network processing tree-structure propagation, and recurrent neural network. We will cover them in detail.

4 Discussion

The comparable results of three models are displayed in table 1 below, the results show the performance with regards to the highest F1 score after fine-tuning techniques were applied. The evaluation was based on the validation datasets. The evaluation results combined with the scores on Kaggle show the BERT model being the optimal model, meaning it makes the most accurate predictions. During implementation of the BERT model, the way of data cleaning and text preprocessing was customized as to boost the outcome accuracy.

Model	Accuracy
BERT Model	0.9534
LSTM	0.8949
HLSTM	0.7769

Table 2: The accuracy of three implemented models

5 Task 2: Rumour Analysis

5.1 The topics of COVID-19

In order to analyze the following, the topics had to be retrieved. This was done through collecting the top 10 most common words in both the rumor and non-rumor tweets. Retrieving usable data meant that the tweets first had to be tokenized and lemmatized.

Below are the top 10 rumour topics: [('trump', 501), ('coronavirus', 452), ('covid', 232), ('president', 225), ('19', 218), ('death', 139), ('say', 133), ('u', 119), ('american', 111), ('people', 100)]

Below are the top 10 non-rumour topics: [('coronavirus', 3874), ('covid', 2344), ('19', 2208), ('trump', 2041), ('new', 1176), ('case', 1000), ('president', 986), ('say', 947), ('u', 897), ('death', 868)]
Upon analysis, it is evident that the rumor and non-rumor topics differ more in popularity than subject. For instance, 'trump' is the highest rated rumor topic yet places fourth as a non-rumour topic.It is evident from the count difference of a common topic in rumors and non-rumors that the rumors branch from the non-rumors. This means that factual information gives rise to the spread of rumors. This with the exception of the following topics: 'american','people' in rumour and 'new','case' in non-rumour.

The highest rating rumor topics hint towards shared public opinion.This meaning that the rumors are central to the current states of affairs of that time, for instance, trump being president, the spread of covid-19, the president's involvement in such and so on. The difference in topics also underlines this, 'american' and 'people' give the notion of public opinion in comparison to 'new', 'case' which give way to fact.

5.2 How do COVID-19 rumour topics or trends evolve over time?

The tweets- by time sorting was achieved through creating a dictionary relating the tweets to their respective time and then sorting the times in ascending order.

Below please find a snippet of the earlier tweets retrieved with the topic 'coronavirus':

Mon Jan 08 08:05:57 +0000 2007 Madonna censured by Instagram after sharing video about a coronavirus conspiracy theory to her 15m followers https://t.co/QCYRdWcp5n Sun Jan 28 01:58:49 +0000 2007 White House ordered NIH to cancel coronavirus research funding, Fauci says https://t.co/TfN8koIn7O by @BethMarieMole Sat Mar 10 23:52:36 +0000 2007 The 2020 New York City Marathon, scheduled for November 1, has been canceled due to the coronavirus https://t.co/ypTTkxq2Nj Sun Mar 11 19:58:10 +0000 2007 A notable shift downward in projected deaths from coronavirus is already being spun as "experts were wrong!!" inste… https://t.co/rRt4dXy8FI

Below please find a snippet of the later tweets retrieved with the topic 'coronavirus':

Tue Mar 06 15:53:26 +0000 2018 Trump is such a psychopath that if a coronavirus vaccine was discovered bu the USA in the next few months, he'd del… https://t.co/wbEx7Jljjc Fri Sep 21 17:10:30 +0000 2018 Since CNN and MSNBC won't show Trump's coronavirus briefings - the White House shouldn't allow any of their reporters to attend it. Wed Mar 11 10:49:01 +0000 2020 coronavirus got my peoples going crazy with the cuts man https://t.co/HmWes7DUlR Fri Mar 27 15:28:09 +0000 2020 If you think the Trump administration's coronavirus response is a disgrace, reply with MoreLiesMoreDie.

The more recent tweets show Trump being the main focus whereas the earlier tweets are more focused on the general recent covid events, this shows an evolution in trends with Trump being a trending topic.

The earlier tweets have more impersonal,professional language use, words like 'due to', 'A notable shift' amongst others. Later tweets however make use of emojis ' ', unprofessional language use, such as: 'crazy with the cuts man', 'Trump is such a psychopath', and are more personal in nature 'the White House shouldn't allow', 'If you think…'. This shows an evolution in trends, the trend of tweet style changed drastically, from the topic of coronavirus being treated somewhat professionally to it becoming susceptible to discussion and public opinion, as well as related to a popular public figure.

5.3 The Popular Hashtags of Rumours and Non-rumours

Rumour	Non-Rumour
#COVID19	#COVID19
#coronavirus	#coronavirus
#Coronavirus	#Coronavirus
#Trump	#Covid19
#covid19	#BREAKING
#China	#covid19
#BREAKING	#CoronaVirus
#Covid19	#China
#BlackLivesMatter	#lockdown
#TrumpPressConference	#StayHome

Table 3: Top 10 most common hashtags in rumour and non-rumour source tweets

It is noticed that both rumour and non-rumour tweets share some of the most popular hashtags: #COVID19, #coronavirus, #covid19, #Covid19, #China and #BREAKING which is understandable as these are closely related to the topic of COVID. There are also some popular hashtags unique to rumours, such as #Trump, #TrumpPressConference and #BlackLivesMatter which seem less relevant to the topic. Some of the popular hashtags that only appear in non-rumour tweets are #StayHome and #lockdown.

5.4 Sentiment of Source Tweets

Sentiment analysis was performed on the source tweets with TextBlob (Loria, 2018). The sentiment is categorized into three groups: positive, neutral and negative. Figure 2 demonstrated the percentage of positive, neutral and negative tweets for rumours and non-rumours respectively. It can be seen that a higher proportion of rumour tweets have a negative sentiment and a higher proportion of non-rumour

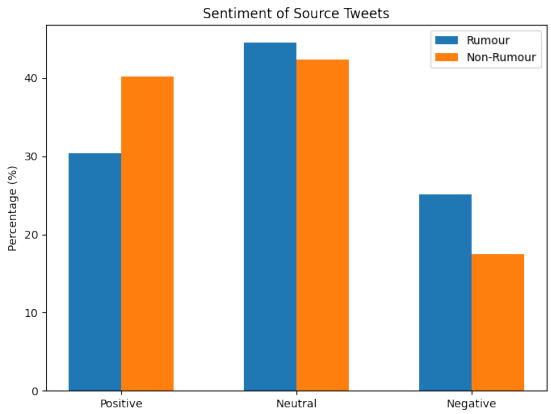

Figure 2: Distribution of the sentiment of source tweets for rumour and non-rumour

tweets have a positive sentiment. The portions of neutral tweets are roughly the same.

5.5 The characteristic of rumor creators

Before exploring the proper characteristic of rumor creators, we speculated that unverified users might be the majority, and most of them prefer to disable the location detection. However, after collecting information from tweet objects, the revealed pattern was not expected. We could not find any considerable difference between rumor creators and normal users regarding these three attributes. The majority of covid19 tweets (21982, approx. 76.88% of total) is predicted as non-rumor. Verified status(rumor): True: 2073, False: 495; Verified status(nonrumor): True: 10568, False: 2826; location enabled(rumor): True: 1408, False: 1160; location enabled(nonrumor): True: 8188, False: 5206; profile_image_status(rumor): False: 2568; profile_image_status(rumor): False: 2568.

6 Conclusion

In this paper, we have discussed three different approaches to rumor detection, these being hierarchical neural networks, recurrent neural networks (RNN) and the BERT model. All three models were successful in execution and detection. The most optimal modal, the BERT model, was then selected for Task 2. The BERT model achieved a score of 0.87500 on kaggle, position 81 on the leader board. The tweet analysis with regards to covid rumors was then based off of the BERT model's results. All assignment objectives were successfully achieved.

7 Contribution

During the project, we have organized 5 group meetings for discussion. For task 1, each of us took one of the proposed model as below: 1. 1307017: BERT Model 2. 980940: LSTM 3. 1315703: HLSTM. FOR task 2, two of listed analysis questions were assigned to each of the group members.

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

Oluwaseun Ajao, Deepayan Bhowmik, and Shahrzad Zargari. 2018. Fake news identification on twitter with hybrid cnn and rnn models. In *Proceedings of the 9th international conference on social media and society*, pages 226–230.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Rini Anggrainingsih, Ghulam Mubashar Hassan, and Amitava Datta. 2021. Bert based classification system for detecting rumours on twitter. *arXiv preprint arXiv:2109.02975*.

Alessandro Bondielli and Francesco Marcelloni. 2019. A survey on fake news and rumour detection techniques. *Information Sciences*, 497:38–55.

Juan Cao, Junbo Guo, Xirong Li, Zhiwei Jin, Han Guo, and Jintao Li. 2018. Automatic rumor detection on microblogs: A survey. *arXiv preprint arXiv:1807.03505*.

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Yi-Chin Chen, Zhao-Yang Liu, and Hung-Yu Kao. 2017. Ikm at semeval-2017 task 8: Convolutional neural networks for stance detection and rumor verification. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 465–469.

Huong Dang, Kahyun Lee, Sam Henry, and Ozlem Uzuner. 2020. Ensemble bert for classifying medication-mentioning tweets. In *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*, pages 37–41.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Steven Loria. 2018. textblob documentation. *Release 0.15*, 2.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics.

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

Changhe Song, Cheng Yang, Huimin Chen, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2019. Ced: Credible early detection of social media rumors. *IEEE Transactions on Knowledge and Data Engineering*, 33(8):3035–3047.