Rumour Detection and Analysis on Twitter

Parikshit Diwan

1 Introduction

In this report we explore the problem of classifying the tweets as rumour and non-rumour, and subsequently analyze the classified tweets. We have divided the objective into two parts - (i) Developing a model to classify tweets as rumour or non rumour (Task A); (ii) Using the developed model to classify tweets centered around the COVID-19 pandemic and perform analysis on these tweets (Task B). For Task A a model based on BERT (Devlin et al., 2019) was used with which a final evaluation F1 score of 0.835 and a rank of 55 was achieved on the CodaLab Leaderboard. This model has been further used in Task B to classify tweets based on COVID19. We conduct an in depth analysis based on the textual properties and the propagation-based properties of the classified tweets.

2 Glossary

In the sections below certain terms are used whose meaning and context is described below:

- **Source Tweet** : The first tweet among a group of tweets.
- **Reply** : A source tweet may have multiple tweets in reply.
- **Tweet Set** : a set of source tweets and replies.

3 Task A : Rumour Detection

3.1 Description

We have been provided a collection of source tweets with their replies. Our task is a binary classification task i.e to develop a model which can predict whether the source tweet is a rumour or non-rumour.

3.2 Data set

The data set has been divided into train, development and test set containing 4641, 580 and 581 tweet sets respectively. The distribution of the train and development set into rumours and non-rumours can be seen in table 1.

	Rumour	Non-Rumour
Train Set	1583	3058
Development Set	187	393

Table 1: Distribution of the Tweets in the Train and Development set

3.3 Approach

The development of the model has been through the eyes of using techniques which capture different properties to generate fixed size representation for the twitter text data. The subsequent representations are then used to train a classifier.

3.3.1 GloVe

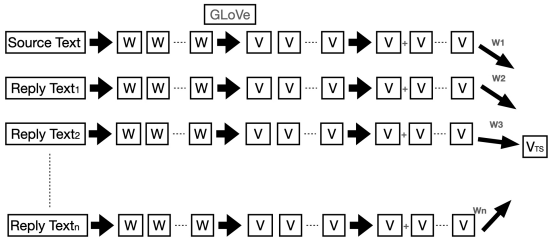

Figure 1: Using GloVe to generate Tweet set representation

Using the GloVe (Pennington et al., 2014) pre-trained word embeddings, a fixed representation of a tweet set is generated. The GloVe embeddings have been trained from 2 billion english language tweets and represent each word as vector of length 200. This is done as following (refer fig 1):

- Text from each source tweet and its replies are pre-processed and tokenized according to the procedures outlined in the specifications.

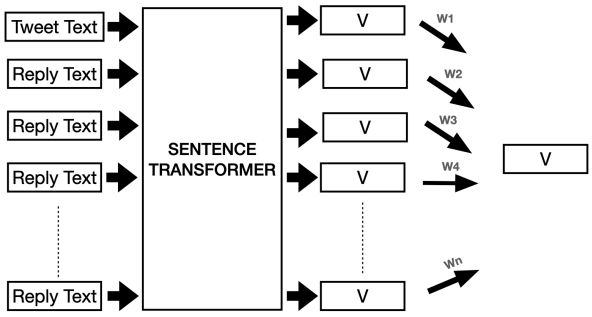

Figure 2: Using Sentence Transformer to generate Tweet set representation

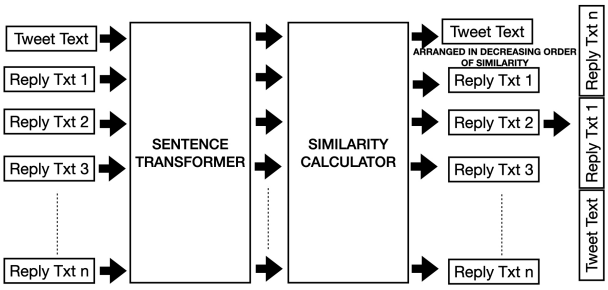

Figure 3: Generating input for BERTweet

- A fixed size representation for a tweet is generated by adding the individual representations for each token.
- A fixed size representation for a tweet set (V_{TS}) is generated by taking the weighted$W$ average of each tweet representation(V_{TT_i}) in the set.

$$V_{TS} = \sum W_i \cdot V_{TT_i} \quad (1)$$

- We experimented with different types of weights including text length, retweet count, user follower count. The best performing weight type is the retweet count (refer to table 2). Additionally, we can observe from table 2 that popularity based weights (retweet, user follower) significantly enhance the model performance and as a result of that should be incorporated.

3.3.2 Sentence Transformers (ST)

Retweet count based weighted vectors prove to be successful in the classification of rumours and non-rumours. We improve on the above method, by changing the base embedding from GloVE to Sentence Transformers (Reimers and Gurevych, 2019) sentence embeddings. This change further enhances the performance of the models, achieving an F1-score of 0.851.

Next, to incorporate the sequence of replies to a source tweet, we trained a Bi-LSTM model. We used the source tweet as the first representation and subsequently the reply tweet's representation were fed to the Bi-LSTM in the order of their time of reply. However this did not improve the overall performance resulting in an F1 Score of 0.798 (refer table 2).

3.3.3 BERTweet

We formulated a method which allows our model to detect a rumour based upon the best argument available for or against the the source tweet. We use BERTweet (Nguyen et al., 2020) which is a BERT model trained on English language tweets. We adopt the following methodology on a tweet set (fig 3):

- First, we generate a fixed size representation of each tweet in a tweet set using sentence transformers.
- For a tweet set, we rank the replies based on the cosine similarity between the representations of source tweet and reply tweet.
- Next, the source tweet text (S) and the reply text with the highest similarity (R_H) and the lowest similarity (R_L) are used to form a representative text for a tweet set (RTS) as follows :

$$RTS = S + R_H + R_L \quad (2)$$

- Finally, RTS was used as input for fine tuning the BERTweet model with a classification layer on top.

3.4 Results & Observations

From the results from the Codalab competition it can be concluded that model 7 performs the best achieving a F1 score of **0.835**. The model has performed the best for the following reasons:

- It has utilized most number of features among all the models
- It directly incorporates the features of the source tweet without diluting them.
- The features have been generated using a model which has been trained upon tweets, unlike sentence transformers.

Additionally from the table 2 we can observe that:

- Models involving GloVe under perform as GloVe uses word embedding which do not account for semantics in a sentence.

#	Embedding	Weights	Model	F1 Score	Precision	Recall
1	GloVe	# text token	Random Forest	0.62	0.759	0.524
2	GloVe	# text token	LightGBM	0.724	0.783	0.673
3	GloVe	# retweet	LightGBM	0.796	0.831	0.764
4	GloVe	# user follower	LightGBM	0.758	0.811	0.711
5	ST	# retweet	LightGBM	0.851	0.894	0.812
6	ST	# retweet	LSTM	0.792	0.798	0.786
7	BERTweet	-	MLP	0.831	0.808	0.855

Table 2: Performance comparison for Task A

- Models involving Sentence Transformers are less generalized, because even though they have a higher F1-score than model 7 they performed badly on the test set as evidenced by the final scores on Codalab.

4 Task B : Rumour Analysis

Non-Rumour(NR)	Rumour(R)
16783	675

Table 3: Predicted labels for Covid-19 data set

Using our best model for Task A we perform the classification on the data provided in the file *covid.data.jsonl*. We obtain the distribution mentioned in Table 3.

4.1 How long does the conversation go on?

Measure	Statistic	NR	R
Thread Life(hrs)	Avg	70.2	20.3
	Median	11.0	2.8
Retweet Count	Avg	3712	1855
	Median	766	563

Table 4: Predicted labels for Covid-19 data set

We define the life of a thread life of a tweet set as the difference in the time of the first source tweet and the last reply tweet in the thread. A higher thread life indicates that the conversation on source tweet lasts longer. Table 4 shows the mean and median thread life of the classified rumours and non-rumours. We observe that the conversation around rumours is significantly shorter as compared to non-rumours. This may indicate that (i) Rumours tend to be debunked quickly (ii) After a period of time, participants do not engage in threads they suspect or know are rumours.

4.2 Can number of replies be considered as an indicator for a rumour?

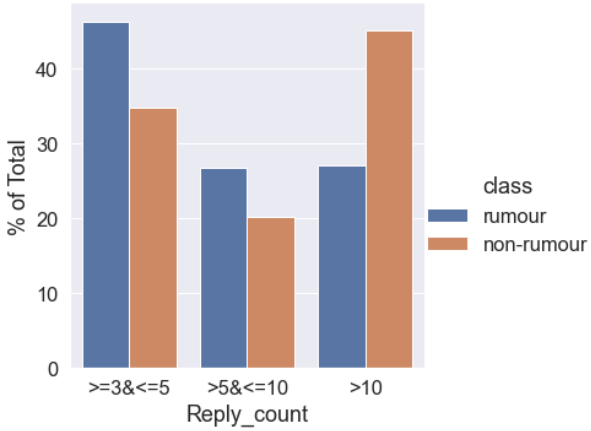

Figure 4: Distribution of Reply count across two classes

To explore this question the reply count has been binned as seen in fig.4. Each bar represents the percentage of tweets belonging to the bin in that class. Rumours tend to have shorter threads while non-rumours seem to have longer threads. This supports our assertion that COVID19 rumours tend to be debunked quickly on twitter.

4.3 What sentiments do the source tweets exhibit in the two classes?

Class	Positive	Neutral	Negative
NR	0.17	99.51	0.31
R	0.14	99.48	0.29

Table 5: Predicted labels for Covid-19 data set

Using NLTK's Sentiment Analyser on the source tweets we get the distribution illustrated in the table 5. We observe that a significant majority ($> 99\%$) of the source tweets are of a neutral temperament. However, for both rumours and non-rumours, there are twice more negative tweets than positive tweets.

4.4 What kind of hashtags are seen across the two classes?Are they different?

NR	R
'covid19', 'coronavirus', 'breaking', 'covid', 'china', 'coronaviruspandemic', 'stayhome', 'cdnpoli', 'lockdown', 'maga', 'stayhomesavelives', 'trump', 'socialdistancing', 'stayathome', 'trump2020', 'staysafe', 'auspol', 'txlege', 'stayalert', 'covid19ph', 'sarscov2', 'blacklivesmatter', 'indiafightscorona', 'covid2019', 'florida', 'wuhan', 'onpoli', 'pandemic', 'hydroxychloroquine', 'kag', 'covidiots', 'covid19nigeria', 'ppe', 'familiesfirst', 'foxnews', 'coronavirussa', 'nhs', 'india', 'takeresponsibility', 'ridge', 'watch', 'marr', 'wearamask', 'tcot'	'covid19', 'coronavirus', 'breaking'

Table 6: Hashtags with freq>10 in source tweets

If we look at the hash tags contained in the source tweets in the source tweets with frequency >10 (refer to table 6). It is interesting to observe that the source tweets classified as rumour seem to be strictly dealing with COVID19 while the ones belonging to the non-rumour set seemed to be touching on more topics than COVID19 like the black lives matter protest and the Trump 2020 election campaign. It can be seen that the three hashtags found in the Rumour tweets are present in the non-rumour hashtags as well which point to the fact that often times the hashtags are incorporated to gain traction for rumours.

4.5 What happens to the length of replies as we go further down the thread?

We expect 'longer a conversation goes the less interest it ignites'. As a result, we expect length of tweets to reduce overtime. So to capture this trend we would expect to see a negative gradient on a time series chart of reply text lengths. We conduct this experiment as follows -

- All the reply tweets texts of a tweet set are arranged according to their *creared_at* field. Tweet text with with less than 5 tokens are ignored (this is done because analysis showed that these mostly contain urls, hashtags and user tags).

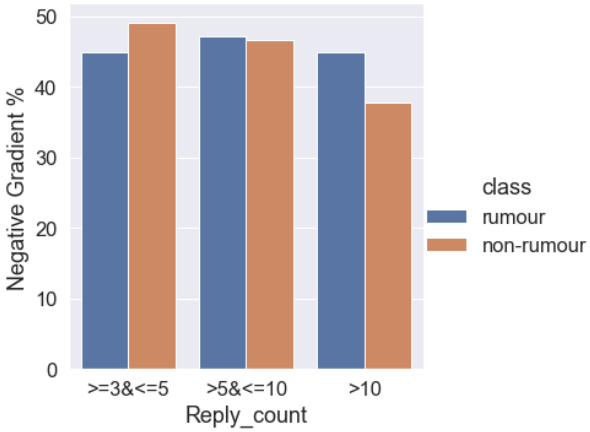

Figure 5: Analyzing the length of replies in a thread

- The word count of the text in these replies is calculated and put into a linear model to calculate the slope or *gradient* of the series.
- These gradients are binned depending on the no. of replies in the tweet set and finally fig. 5 is generated which shows the percentage of negative gradients in a bin.

Looking at fig. 5 we can observe that as the length of threads (i.e number of replies in a thread) grow longer, the lengths of rumours decrease more because a higher percentage of tweets sets that belong to that bin have negative gradient. This supports the idea that the main objective in the case of rumour threads is the propagation of source tweet, rather than meaningful contributing to the discussion by the replies.

5 Conclusion

So through this project we have found BERTweet to be a high performing model for the task of rumour detection.further analysis of the COVID-19 twitter data set shows that rumours tend to have a short life span and the longer their thread goes on we are less likely to see any meaningful conversation. to be added.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.