Detecting 2020 paranoia: COVID-19 on Twitter

1015501

Abstract

This document contains de description of methods to detect COVID-19 rumour on twitter. The whole problem is divided in two task using Bert and Recurrent neural networks to train a classifier and the explore the results for the classifier in an unlabelled dataset using sentiment analyst, topic modelling and other unsupervised techniques.

1 Introduction

Rumour detection in social media is a relevant task. The internet that has been able to extend human capabilities of communication, it also extends human capabilities for misinformation. Several resources are invested to this problem trying to automatically predict and stop misinformation in social networks such as Twitter. Given the recent events of COVID-19 pandemic, stopping the spread of misinformation related to public health issues has the additional impact of saving lives and economies.

In the context of COMP90042 we aim to build an analysis tool to perform such predictions and understand some key features of the rumour spread around COVID-19.

2 Related work

The task of rumour detection can be found over a decade of research (Vosoughi, 2008) (Hamidian & Diab, 2019).

Before the irruption of Recurrent Neural Networks (RNN), most models relied on the construction of features whether textual or not. (Zhou et al., 2019) Among the features utilized textual, user centred, and conversational structure were employed, as can be seen in detail in the next section. (Vosoughi, 2008).

The most challenging aspect is to deal with the tree structure of the information, attempt using graph neural networks (Yuan et al., 2019).and maxpooling with RNN to capture sequential features (Ma et al., 2018) are used.

These techniques were surpassed using sequential networks Transformers, archiving superior evaluation metrics relying only on the tweet text, without the necessity to engineer additional features (Alzanin & Azmi, 2018).

3 Methodology description

The available data correspond a set of tweets representing an event. These tweets relate to each other forming a tree-like conversation.

The entire task was divided in two faces. First, a binary classifier model was developed using a labelled data of twitter events. For this step, the actions considered three lines, namely data pre-processing, model selection and fine-tuning. Models were trained and evaluated in three sections of the data set. For the last section labels are unknow and evaluated by a competition in Codalab, only partial results are returned, only the highest scoring model can have full results by the end of the competition.

In a second face the model was applied to a similar unlabelled dataset containing COVID-19 related twitter events. The resulting groups of rumour and non-rumour are analysed trying to describe properties of their structure, user, or text features. Aspects such as sentiment score, unsupervised topic modelling and clustering are used to attempt to explain the nature of COVID-19 rumours.

4 Developing rumour detector face

The first group of labelled datasets is used to develop a model. The datasets consist in a training set with 4641 events. About 1/3 correspond to rumour events.

4.1 Data pre-processing

The data for each tweet considers information about the users, the text and response information. The latter was used to rebuild a tree structure of tweet and responses for each event. In the case of

missing tweets, a blank tweet was created and assumed to be linked to the source or root tweet.

These data types allow to create different variations of features, likewise, versions of the datasets applying diverse processes were created.

Textual data: The tweet text is the main feature utilized. To make the data consumable by the models the text of every tweet in an event was concatenated. As the order of the sequence is quite important, four strategies were tried, starting with the default order of tweet in the dataset, then order by date and hour of the tweet creation, then using the tree structure and flatten it in pre-order. Ties among child nodes were solve by ordering them by the number of grandchildren or by children 'like' count.

Additional features: After the text set were tested, additional features based on literature were incorporated in the last layers of the model to attempt to improve its performance. The following features were incorporated:

1. The tree heigh normalized by # of nodes.
2. The # replies where the replying user is more popular than the source, normalized by # total replies in the tree
3. The ratio between the # of leaf nodes versus all nodes.
4. The source user friends versus follower's ratio.
5. The proportion between sources # 'likes' and the average 'likes' of answers.
6. Boolean if source tweet has hashtags
7. Boolean if source tweet has hyperlinks.

4.2 Model selection

Following the literature, the most advanced models are neural network in the form of RNN and Transformers. Two types of models were tried. On one hand, a bidirectional stacked Long Short Term Memory (LSTM) network was implemented with additional pre-processing to turn tokenized words to Glove vector (100 dimensions is utilized). This model is selected given de double sequential nature of the data. A Recurrent Neuronal Network can capture the sequential nature of tweeter conversation. However, this model suffers from vanishing gradients, where initial tweet and words tend to have reduced importance in the model. To compensate this the LSTM keeps cells with information to future states, also a bidirectional network is utilized to compensate for the quite long concatenation of texts, increasing the chances of the source tweet to make a greater contribution.

On the other hand, a Bert based model was implemented. Tokenization is produced by same tokenization tool of the pre-trained Bert model. Two variations were tried: one using vanilla Bert model as embedding without fine tuning and an another fine tunned. On top of Bert a bidirectional Grated Recurrent Unit with two layers and a linear layer a was used for classification. On the understanding that an additional sequence layer may improve the classification and that GRU is simpler to calculate than LSTM but achieving similar scores. Both models were implemented using PyTorch.

For results in the first round of tries, I tested simultaneously both Models and datasets, the results using F1 score are summarised in Table 1, where simple, date sort, Tree-depth and Tree-like stand for sorting criteria where datasets were trained. Two version of the models were incorporated: One without additional pre-processing and one where *stopwords* were removed (sw).

Model	Simple	Date sort	Tree Depth	Tree Like
BI-LSTM	0.727	0.72	0.714	0.728
BI-LSTM sw	0.77	0.7611	0.742	0.743
BERT biGRU	0.78	0.784	0.775	0.774
BERT biGRU sw	0.74	0.741	0.743	0.735

Table 1: F1-score for table and dataset

Surprisingly, the tweets order did not impact the score significantly but removing *stopwords* did, but in opposite scale for both models. The most likely explanation is Bi-LSTM improved its score eliminating *stopwords* as it reduced noise and reduced the length of the units, capturing more information of the remaining tokens. Whereas Bert did not gain information as is more robust to capture information from the whole sequence, on the contrary it was originally trained with those tokens.

Fine Tuning Bert variations where no tested with all dataset given its long times to fit a model. The simple dataset, as it performed more consistently was selected for the remaining tasks.

4.3 Hyperparameter and results

Regarding fine-tuned version of Bert, the first attempt increased the F1-score of previous models, therefore it was selected to search for additional parameters. A double layer and Bidirectional GRU with a dropout rate of 0.25 was used fed with the full last hidden state of the Bert mode. Single layer, unidirectional and lower and higher of dropout values were tested but no configuration outperformed the above described.

A Bert model with only a single flat layer for classification was trained as baseline. a Batch size of 16 and learning rate of 2e-5 (Adam) following creator recommendations (Devlin et al., 2019).

Regarding the BERT model, the maximum sequence length of the data loader was systematically tested for every variation model. In theory the longest sequence, the most information Bert can handle, however the training time increases. The maximum length that the original model can handle is 512 (Devlin et al., 2019).

For Bi-GRU the removal of *stopwords* helps to improve the F1-score until the model starts to lose some information when passing longer sequences. The vanishing gradients effect comes in, despite GRU capabilities as some aspects may be left out the memory cells which affects its performance.

Fine tunned Bert can take advantage of *stopwords* removal as it can adjust the parameter for the missing words outperforming its static counterpart. Surprisingly, Baseline model achieve similar scores than its Bi-GRU alternative, although it takes a longer time to train.

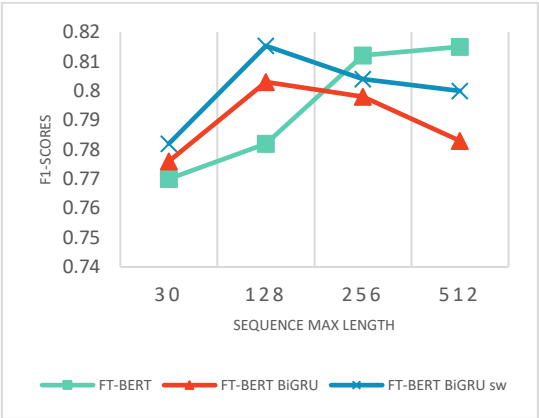

Figure 1: F1 Score Fine tune Bert model vs max sequence length.

Regarding the additional features to help to improve the model. The features were incorporated one by one concatenating them in a linear layer before the final classification layer. However, none of them help to improve the model score, on the contrary, most only added noise to the system and none were incorporated in the final model.

5 COVID-19 dataset classification

In this task the classification model was applied to COVID-19 events dataset and explored how different are rumour 'candidates' of their counter parts. These explorations use off-the-shelf tools which are not perfect, but they have shown excellent performance with tweet data before (Hutto & Gilbert, 2014).

The final distribution of rumour shown that 2564 out of 17458 events were classified as rumours. This proportion (14%) is even smaller than the original training set (34%). This disproportion add some challenges to the analysis, as it can be easy to miss certain trends.

5.1 Analysis of textual features

To capture snippets of content bi-grams and tri-grams were built to extract common concept of the data. These is relevant for the model as it express common sequential features in the text.

Regarding the non-rumour group, the most frequent combinations of words have multiple names of politicians and governmental organizations. These may signal that non rumour may have or mention some institutional framework that support those claims.

On the other hand, while its share some of the non-rumour keywords, rumour include words of recent events and dead, and comparison (less than, higher than, so far, last week, days ago, past hours) but also claims of fake news, as some response tweet may be doubting about the information presented.

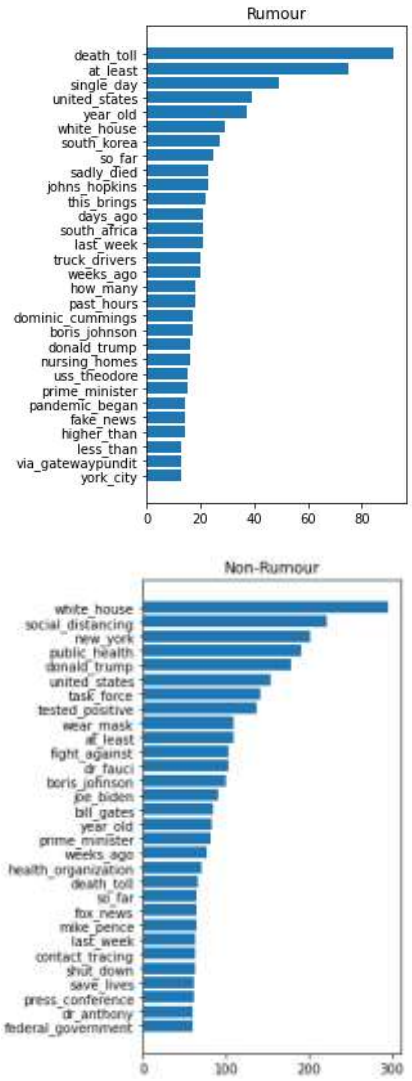

Figure 2: most common phrases by label

5.2 Sentiment

Regarding the overall sentiment of the events, two lines of process where followed. For one side the sentiment of the concatenated text was calculated. For the other, the sentiment score of each source tweet was calculated. To calculate these scores Vader polarity tool within the NLTK packages was utilized (Hutto & Gilbert, 2014).

As it observable from figure 3 the sentiment polarity is similar for both groups, but rumours tend to be proportionally slightly more positive for both text and source.

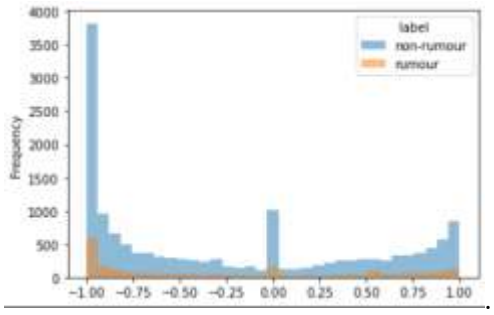

Figure 3: Polarity score distribution by label.

5.3 Topics

To summarize the content of the event a topic model was built using LDA algorithm with MALLET implementation (McCallum and Kachites, 2020). Is an unsupervised technique but requires to select the number of latent topics. Coherence score was computed to determine the proper number of topics, testing several values and selecting the one that increases the score within fewer topics, similar to the 'elbow' technique in clustering tasks (Campbell et al., 2015).

As is shown in figure 4, the topic where rumour have higher proportion Topics 25, 19, 15 and 9, corresponding to how COVID-19 dead are counted, if celebrities got tested for COVID-19, number of COVID-19 cases in Africa, particularly in Nigeria, regarding an increase in the number of cases and taking bleach and vitamins to cure the virus in Florida, US, respectably.

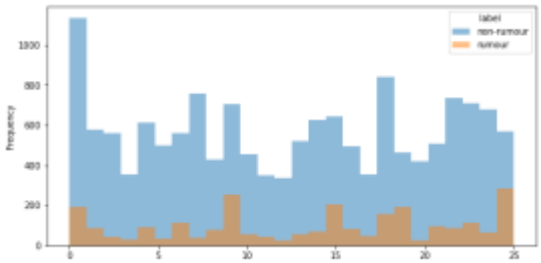

Figure 4: Frequency by topic code by label.

6 Discussion and further improvements

Finally, regarding the classifier development, in order to try to maximize the sequential information multiple [SEP] token was included in Fine tuning Bert model, however the model only accepts two sequences. It does produces better models in development set, but It does not on test data, therefore it was discarded. Further experimentation is required following these lines as more carefully selection of tweets within an event can be potentially critical to achieve better performance.

References

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

Campbell, J. C., Hindle, A., & Stroulia, E. (2015). Latent Dirichlet Allocation: Extracting Topics from Software Engineering Data. *The Art and Science of Analyzing Software Data*, *3*, 139–159. https://doi.org/10.1016/B978-0-12-411519-4.00006-9

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*(Mlm), 4171–4186.

Hamidian, S., & Diab, M. (2019). Rumor Detection and Classification for Twitter Data. *ArXiv*.

Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, *January*, 216–225.

McCallum, Andrew Kachites. (2020) "MALLET: A Machine Learning for Language Toolkit."

http://mallet.cs.umass.edu.

Ma, J., Gao, W., & Wong, K. F. (2018). Rumor detection on twitter with tree-structured recursive neural networks. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*, 1980–1989. https://doi.org/10.18653/v1/p18-1184

Vosoughi, S. (2008). Automatic Detection and Verification of Rumors on Twitter by Soroush Vosoughi. *Massachusetts Institute of Technology*, *2008*. https://dspace.mit.edu/handle/1721.1/98553

Yuan, C., Ma, Q., Zhou, W., Han, J., & Hu, S. (2019). Jointly embedding the local and global relations of heterogeneous graph for rumor detection. *Proceedings - IEEE International Conference on Data Mining, ICDM*, *2019-Novem*, 796–805. https://doi.org/10.1109/ICDM.2019.00090

Zhou, K., Shu, C., Li, B., & Lau, J. H. (2019). Early rumour detection. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 1614–1623. https://doi.org/10.18653/v1/n19-1163