Rumour Detection and Analysis on Twitter

Student ID: 739 674

Abstract

To help stopping the spreading of rumours on social media, a rumour detection algorithm was designed to identify the rumours on Twitter based on the given training set. During the development, LSTM and Graph Networks were applied respectively on the training set, and finally Graph Attention Networks was found to be the best performance based on the F score on the positive data. Furthermore, on COVID-19, the characteristics of the rumour twitters and their authors were inspected and analyzed, and the most frequently used hashtags, topics were also identified.

1 Introduction

Rumours is typically defined as unverified statement or news circulating from person to person (Jey Han Lau, 2021). Nowadays, with the increasing impacts of social media, the spreading of rumours has significant damage to the society and world. Donald Trump, the former US president, used to spread rumours and fake news on his Twitter. Although his account was disabled after causing hundreds of people wounded and a series of chaos, the damage he made to the world won't be recovered. Therefore, with the development of the Artificial Intelligence and Data Science, it is feasible to implement a rumour detection algorithm on social networks, such as Twitter, to identify and prevent the unverified statements or news from circulating.

2 Data Preprocessing

The project-data.zip includes the data.jsonl for training, development and testing. They have similar structure where the source and its reply twitters are stored as objects and clustered together. The raw data can be obtained by calling key specified in Twitter API documentation (Twitter Inc., 2021). The zip file also includes the label data for training and development. These json files only include the unique twitter id ["id_str"], followed by whether the corresponding twitter is "Rumour" or "Non-Rumour".

In the development the following procedures were followed to preprocess the data:

1. Read the .data.jsonl as clusters of twitters.
2. **Sort** all the twitters in a cluster according to their publishing time.
3. **Find the source**: Set the first twitter as the source in the cluster
4. **Fetch the required data:** Get the unique id of each twitter ["id_str"], the unique twitter id which the current twitter replies to ["in_reply_to_status_id_str"], the author's user id ["user"] ["id_str"] and content of the current twitter ["text"]
5. **Filtering and tokenization**: Apply regular expressions to filter the ["text"], including replacing the abbreviations with full phrases, removing the "@" "\n" "http://" "www" and other the non-English characters.

After the filtering, undertake lemmatization and tokenization using *TweetTokenizer()* from *nltk*

6. **Padding**: add empty spaces to the end of each ["text"] to make every ["text"] the same length as the longest ["text"]
7. **Adjacent Matrix**: Work out the adjacent matrix of each cluster from the information obtained in step 4
8. **Encoding:** The *tokenizer* from *TensorFlow* was applied to fit and encode the data obtained in step 5. In the .label.json, we use ["id_str"] to find the if each twitter is rumour or not and set
    a. "Rumour" to 1
    b. "Non-rumour" to 0

9. **Oversampling:** After investigating the label data, we found ratio between the positive and negative data is:

$$rumour : non - rumour = 0.5177$$

To achieve the balanced positive and negative data, we can change the weight of the positive samples from 1 to 2 in the *WeightedRandomSampler()* from *Torch*.

3 Feature Engineering

Based on the data obtained in section 2, several features can be obtained to help the model make the prediction:

1. **Whether the User Name is Pure Number / Unmeaningful words**

For ordinary people, when they register the twitter account, they are likely to use something that has unique meaning to them or even their real name as their twitter name, which are word and characters. But for abnormal accounts, which are normally batch registered, the owner tends to use pure number as user name for easy registration and management.

2. Number of Followers and Friends

People tend to use their alternative accounts to publish something they are unsure, so that if that twitter was found to be false, it will not damage the trustworthy of main account and they won't be responsible for that. This kind of accounts normally have very few followers and friends. However, some famous celebrities with a lot of followers would still spreading rumours. Donald Trump would be a convincing example.

3. The Time When the Account was Created

The alternative accounts are normally created relatively recently compared with the normal ones. As people would like to hold multiple alternative accounts and these accounts are frequently closed for inactivity or rules violation.

4 Modeling

4.1 LSTM Modeling

LSTM networks are well suited for handling sequential data for its long and short memory mechanism. Meanwhile, words and sentences are exceptional example of the sequence data. Both of these make it remarkable for handling the twitters.

In this task, as shown in Fig.4.1 & Fig.4.2, LSTM classifiers are applied respectively to train and predict the rumour twitter. Both of them are taken the tokenized ["text"] data as input, using *tokenizer.texts_to_sequences* from *TensorFlow*, combined with the features we obtained from section 3. LSTM model in Fig. 4.1 only takes the source twitter, while LSTM in Fig. 4.2 takes concatenated twitters from both source and replies.

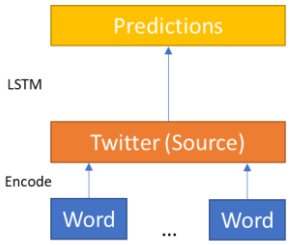

Fig. 4.1 LSTM Applied on Source Twitter Only

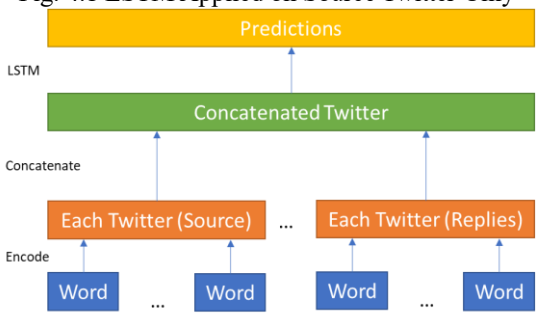

Fig. 4.2 LSTM Applied on Concatenated Twitters

Keras.Sequential from *Tensorflow* was used in setting up the model, with the parameters tuned as below:

```
  # Python 3.6
  model = tf.keras.Sequential([
      tf.keras.Input(shape=(None,),
dtype="int32"),

tf.keras.layers.Embedding(20000,
128),

tf.keras.layers.Bidirectional(tf.ke
ras.layers.LSTM(64,
return_sequences=True)),

tf.keras.layers.Bidirectional(tf.ke
ras.layers.LSTM(64)),
      tf.keras.layers.Dense(64,
activation='relu'),
      tf.keras.layers.Dropout(0.5),
      tf.keras.layers.Dense(1,
activation='sigmoid')
  ])
```

4.2 GAN Modeling

Different from LSTM, graph networks are suited for modeling the dependency and relationship among the nodes, which the source and reply twitter have. As shown in fig. 4.3, the relationships between the source and replies tweet (as shown in orange arrows), the authors and their tweet (as

shown in blue lines) and the timeline of all the tweet can be modeled as a graph network.

As shown in the heterogeneous graph Fig 4.3, the time line is a vital factor, as all the previous twitters are visible when a user is posting something, and it is the reason why we are applying GAN to all the twitters on the time line to identify the effects of the reply (twitter 2-4 in the example) on the trustworthy of source twitter (twitter 0 in the example). From the time line prospective, the source twitter will affect the replies in the following twitter, and the information flows from the source to replies. However, in the modelling, the information we obtained in the replies will affect the trustworthy of the source twitter, so the information will propagate in the other direction of the timeline.

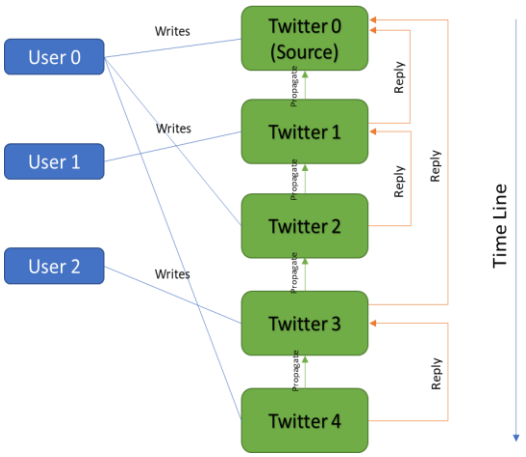

Fig. 4.3 The Graph representing the Tweet Networks

In data the preprocessing, the adjacent matrix has already been obtained, together with encoded text and labels. A *Pytorch* implementation of the Graph Attention Network model (Veličković et. al, 2017) was fit and trained with the tuned parameters below:

batch_size = 128; n_epochs = 300
learning_rate = 2e-3; weight_decay = 1e-3
nhid = 64; dropout = 0.1
t_num_layers = 2; t_num_head = 2
g_num_head = 8; g_alpha = 0.25

5 Performance

After the preprocessed training data was obtained, it was fitted into different models and then tested by the development data several times to tune the parameters, and finally applied the processed test data into the trained models. The predicted result was uploaded to *codalab* to evaluate the performance of each model:

Modeling	F Score	Recall	Precision
LSTM with Concatenated Twitter	0.65	0.59	0.73
LSTM with Source Twitter	0.79	0.80	0.79
Graph Networks (GAN) Ongoing	0.81	0.77	0.84
Graph Networks (GAN) Final Evaluation	0.80	0.77	0.82

Table 1 Performance of the Models

As shown in table 1, in general, the LSTM model is worse than the GAN model, that is because it neglects the relationships among the source, reply and users. Although the concatenated LSTM model takes in all the tweets and considers all the information, it is the worst of all because it considered all the tweets equally, and neglects effects of the replies on the source. The GAN in the final evaluation is slightly worse than the ongoing evaluation, that is probably because the overfitting of model and the uneven distribution of the test data. The full set of hidden test data is required to undertake further investigation.

6 COVID-19 Analysis

Using the trained model obtained in section 4.2, the COVID-19 related twitter was preprocessed and predicted as rumour or non-rumour, on which an in depth analysis was based to formalize the characteristics of the COVID related rumour tweet.

Hashtags can be obtained from ['entities'] ['hashtags'] of each tweet.

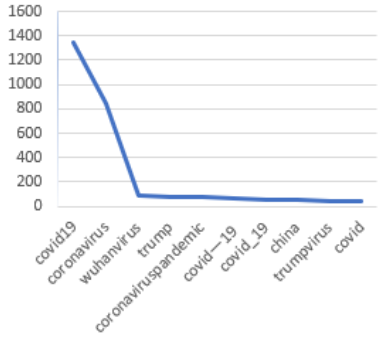

Fig. 6.1 Top 10 Hashtags Counts in COVID related rumours

	Hashtag(Rumor)	Counts(Rumor)
0	covid19	1345
1	coronavirus	842
2	wuhanvirus	95
3	trump	83
4	coronaviruspandemic	73
5	covid−19	61
6	covid_19	56
7	china	53
8	trumpvirus	43
9	covid	39

	Hashtag(Non-Rumor)	Counts(Non-Rumor)
0	covid19	32230
1	coronavirus	17973
2	trump	2579
3	coronaviruspandemic	1550
4	trumpvirus	1471
5	wuhanvirus	1421
6	maga	1200
7	covid_19	1199
8	covid	1057
9	covid−19	1043

Fig. 6.2 Top 10 Hashtags in COVID related rumours and Non-rumours

As there are only 1485 twitters out of 17458 twitters are predicted as rumours, the hashtags out of top 10 was not considered to reach a more generalized conclusion.

As shown in Fig.6.2, the top 10 hashtags in COVID related rumours and non-rumours share a lot in common. However, #China appears only in the top 10 of the rumour twitter, and #maga only appears in the non-rumour twitter.

User characteristics

As discussed in the feature engineering, the users generating the rumours are most likely to have random / pure number user name, has few friends or followers and are recently registered.

Word Counts

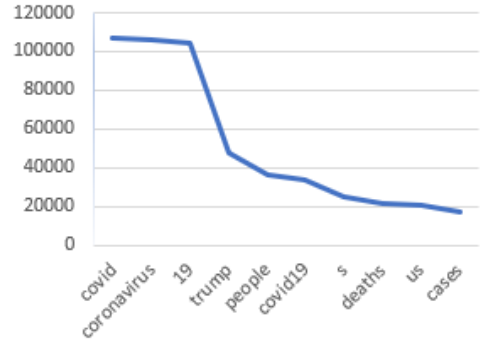

Fig. 6.3 Top 10 Word Counts in COVID related rumours

As shown in Fig.6.3, we can identify the most frequently discussed topic in COVID-19 related tweets are about trump, people, deaths and cases.

6.1 Conclusions

Compared with LSTM, the GAN network was more suitable for identifying the rumours. Furthermore, on COVID-19, the characteristics of the rumour twitters and their authors were successfully inspected and analyzed, and the most frequently used hashtags, topics were also identified.

References

Jey Han Lau. 2021. *The Project Specification to COMP90042 Natural Language Processing*. The University of Melbourne, Carlton VIC, AU.

Twitter Inc. (n.d.). *Twitter API Documentation. Twitter API.* Retrieved May 15, 2021, from https://developer.twitter.com/en/docs/twitter-api.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

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.