LOW LEVEL DESIGN AND IMPLEMENTATION DOCUMENT

"RUMOUR DETECTION AND VERACITY VERIFICATION USING SOCIO-LINGUISTIC DATA AND SOCIAL GRAPHS "

UE17CS490B – Capstone Project Phase – 2

Submitted by:

Sukanya Harshvardhan	**PES1201700214**
Sirisha Lanka	**PES1201700294**
Prajna Girish	**PES1201701261**

Under the guidance of

Prof. Bhaskarjyoti Das
Visiting Professor
PES University

January - May 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

TABLE OF CONTENTS

1. Introduction

1.1. Overview

Society has moved online. Although this has been majorly a boon owing to its accessibility and infinite resource power, there is a major downside which needs to be addressed. People nowadays acquire all their information from social media platforms, rather than conventional news channels, and usually believe most of what they see. With all social media becoming increasingly popular, it is possible that information posted by any user on the network, grasps the attention of almost a billion users, within a few seconds.

Since there are no official restrictions on the content of what is being posted on these social media platforms, there tends to be a lot of unverified information online, also known as rumours[1]. The low cost of information exchange on such social media sites help data and information to spread extremely rapidly, and the more a rumour spreads the more people tend to believe it. Hence, their early detection is of utmost importance.

Our focus for this research problem mainly is on Twitter data, considering that it is the primary social media platform used to post breaking news and other current affairs.

This document explains the low level implementation of our project. The entire problem has been broken down into 3 sub - problems, and each of these tasks have a separate implementation.

First, is the rumour detection aspect, where we determine if a tweet's content contains to be verified rather than it just containing an expression. Second, is the stance taken by a user and to determine his/her feelings towards this particular tweet. Finally, is the veracity verification where we determine if the rumour at hand is true, false or unverified.

1.2. Purpose

This document is necessary for a deeper understanding about the implementation of the project. We have built four different models, some based on the text present in the Tweet being considered, and others based on a social graph[2] perspective. This document explains the underlying implementation details, the different machine learning and deep learning models used, and what modules and functions we have used to achieve our goal.

1.3. Scope

The scope of our project includes only rumours present on Twitter. It does not take fake news into consideration, which is news that is propagated with an intention and a targeted audience. It is verifiably wrong information being spread. Rumours on the other hand could be true or false. Further, this application is only applicable to Twitter data. If the metric is successful enough, it could be extended to Instagram and Reddit data as well.

2. Proposed Methodology / Approach

We have tackled the task of rumour detection with two different methodologies. One being on the basis of the textual content in the tweet ie. on the basis of the socio-linguistic[3] data present in it. Second, is including a social graph perspective as well. For the task of text based rumour detection, we have used a BERT model. Text classification is one of the most common tasks in NLP. It is applied in a wide variety of applications, including sentiment analysis, spam filtering, news categorization, etc. The Transformer is the basic building block of most current state-of-the-art architectures of NLP. Its primary advantage is its multi-head attention mechanisms which allow for an increase in performance and significantly more parallelization than previous competing models such as recurrent neural networks. We have used a pre-trained BERT, one of the most popular transformer models, and fine-tuned it especially for the use case of rumour detection.

Second, for the task of graph based rumour detection, we analysed the ego centric network of the user that has posted the tweet and incorporated the propagation structure of the conversation thread. For this purpose we have used a random forest classifier to achieve high accuracies. We chose this type of classifier because the Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. The reason it works so well is because A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. Following this, we tackled the aspect of stance detection. We have used the Torchmoji model which detects the underlying emotion present in a statement and captures the tone of the user that replies to a source post. Along with this, Bi-LSTM[4] is used to achieve the same which captures the sequential structure of the conversation thread. Finally, we verify the veracity of the tweet and classify it as True, False or Unverified.

2.1 Algorithm and Pseudocode

Text Based Rumour Detection

For the purpose of rumour detection, we have used a BERT transformer. Our model is given all the related tweets and then performs various forms of data cleaning and data preprocessing. First off, all the Emoji characters are removed, making the text more easy to comprehend and understand. Next, the words in the Tweet are embedded using BERT which performs deep bidirectional embedding, trying to capture all token and position related information to produce a real-valued vector[5]. The elements of these vectors serve as parameters to our model.

For the tokenizer, we use the "bert-base-uncased" version of BertTokenizer. Using TorchText, we first create the Text Field and the Label Field. The Text Field will be used for containing the news articles and the Label is the true target. We limit each article to the first 128 tokens for BERT input. Then, we create a TabularDataset from our dataset csv files using the two Fields to produce the train, validation, and test sets. Then we create Iterators to prepare them in batches. We used the "bert-base-uncased" version of BERT, which is the smaller model trained on lower-cased English text (with 12-layer, 768-hidden, 12-heads, 110M parameters).
Finally, the embedded vectors in the Tweet are stored in a database, for easy data retrieval and access. The tweets are then fed into our transformer model, as part of the training data and our model learns the optimal weights for different parameters through backpropagation. These weights are fine-tuned further using Adam optimization.

We used a suitable learning rate to tune BERT for 5 epochs. We use BinaryCrossEntropy as the loss function. The output is passed through Sigmoid before calculating the loss between the target and itself. During training, we evaluate our model parameters against the validation set. We save the model each time the validation loss decreases so that we end up with the model with the lowest validation loss, which can be considered as the best model.

The next stage of our project is where the user and his/her input comes into play. A user wants to verify if a given tweet with a particular tweet ID contains information that is true or not. The user inputs the tweet into our model. The same procedure of Emoji character removal and word vectorisation is followed, and done as part of the data preprocessing phase. Following this, our model predicts whether the content of the tweet is agreeing/ denying/questioning a topic, rather than stating a topic.

Graph Based Rumour Detection

For every tweet - we extract 5 features from it, namely:

- Number of followers of the user
- Number of accounts the user follows
- Whether the user is verified on Twitter or not
- Tweet's favourite count
- Tweet's retweet count

An important feature that needs to be captured is the propagation structure of the tweet- how exactly it travels through the social network of Twitter and how it moves from one user to another. This can be captured by the Tweet's *favourite* and *retweet* count. For every tweet, we create a dictionary for all 5 of these features, with the key as the feature name and the corresponding value of that feature. The conversation thread is then represented as a dictionary of feature vectors in a tree structure. This dictionary and all associated features and then vectoried using the sklearn DictVectorizer. Following this, the obtained vector is being fed into a random forest classifier, which does the task of classifying it as a rumour or not.

Finally, we combine both the rumour detection models by assigning a weightage to each model on the basis of its accuracy. We also look into the confidence scores offered by both of the models and then make an informed decision on what has a higher weightage. After combining both models, we have successfully obtained an accuracy of 93%.

Stance Classification

Using Torchmoji and a Bi-LSTM

Here, we classify whether a given tweet is supporting, denying, querying or commenting the given source tweet. The stance of a tweet is being classified with the help of layers of LSTM units and uses factors such as cosine similarity and the number of negative words/ punctuations to determine how similar or different the reply tweet is from the source.

The input at each time step of the LSTM layers is a reply from the conversation thread and the output at each time step is the stance of the reply.

The layers of LSTM units help understand the sequential structure of the conversation tree which includes the source and reply tweets.

We have used a neural network architecture that uses layers of LSTM units to process the whole branch of tweets, thus incorporating structural information of the conversationThe input at each time step i of the LSTM layer is the representation of the tweet as a vector. We record the output of each time step so as to attach a label to each tweet in a branch5 . This output is fed through several dense ReLU layers, a 50% dropout layer, and then through a softmax layer to obtain class probabilities. We use zero-padding and masks to account for the varying lengths of tweet branches. The model is trained using the categorical cross entropy loss function. Since there is overlap between branches originating from the same source tweet, we exclude the repeating tweets from the loss function using a mask at the training stage.

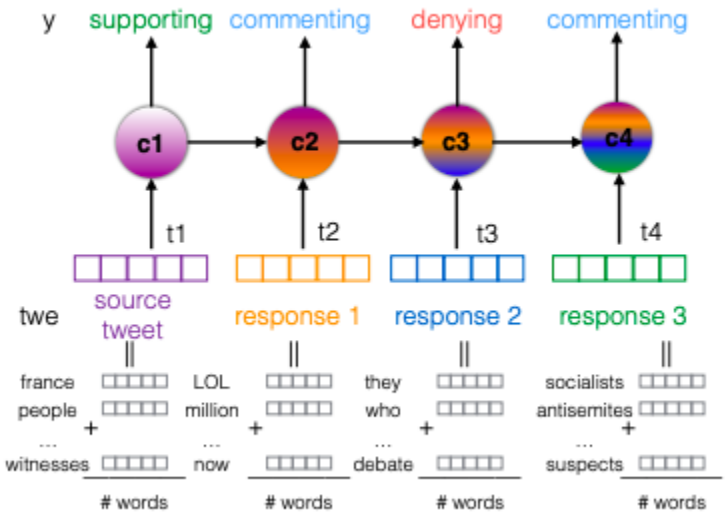

Figure 10: Illustration of the input/output structure of the Bi-LSTM branch model for Stance Classification

In addition to this, we are using the torchmoji module in python for better results. With the help of this module, emoticons are assigned to a particular tweet, based on the emotion present in the content of the tweet. The emotion associated with the majority of the emoticons help understand the stance of the speaker and hence classify the stance of the reply tweets.

Veracity Verification

To determine whether the Tweet is True, False or Unverified, we use a combination of the outputs obtained from the Torchmoji model and the Bi-LSTM in the stance classification phase, as well as anCaother Bi-LSTM which is solely focussed on predicting the veracity of the tweet. We have assigned a particular weightage to the outputs obtained from the different models, which helps us obtain one outcome. We have chosen a Bi-LSTM for this purpose because here we feed the learning algorithm with the original data once from beginning to the end to learn past information, and once from end to beginning, to learn future information. Also,it usually learns faster than a one-directional approach.

2.2 Implementation and Results

- Rumour Detection - Text Based : *BERT Classifier*
  Accuracy Obtained: 73%

- Rumour Detection - Graph Based : *Random Forest Classifier*
  Accuracy Obtained: 73%

- Stance Classification :
  Accuracy Obtained: 78%

- Rumour Veracity Verification : *Bi-LSTM Model*
  Accuracy Obtained: 92%

Appendix A: Definitions, Acronyms and Abbreviations

1. **Rumour**: Piece of unverified information. Has some element of ambiguity attached with it.

2. **Social Graph**: A social graph is a diagram that illustrates interconnections among people, groups and organizations in a social network.

3. **Socio-Linguistic**: The language spoken on social media platforms such as Twitter does not abide to the norms of the conventional English language- there is a limit on the number of characters, people use abbreviations and slang, and there might also be spelling errors. This type of language comes under the umbrella of socio-linguistic data.

4. **Bi-LSTM**: A model in which one particular layer learns the bidirectional long-term dependencies between time steps of time series or sequence data.

5. **Vectors:** They are commonly used in machine learning as they lend a convenient way to organize data. A vector is a tuple of one or more values called scalars. Vectors are built from components, which are ordinary numbers.

Appendix B: References

- MA, Jing; GAO, Wei; MITRA, Prasenjit; KWON, Sejeong; JANSEN, Bernard J.; WONG, Kam-Fai; and CHA, Meeyoung. Detecting rumors from microblogs with recurrent neural networks. (2016). *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*. 3818-3824. Research Collection School Of Information Systems.

- Ahsan, Mohammad & Kumari, Madhu & Sharma, T. (2019). Rumors detection, verification and controlling mechanisms in online social networks: A survey.

- MA, Jing; GAO, Wei; and WONG, Kam-Fai. Detect rumors in microblog posts using propagation structure via kernel learning. (2017). Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), Vancouver, Canada, 2017 July 30 - August 4. 708-717. Research Collection School Of Information Systems.

- L. Poddar, W. Hsu, M. L. Lee and S. Subramaniyam, "Predicting Stances in Twitter Conversations for Detecting Veracity of Rumors: A Neural Approach," 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), Volos, 2018, pp. 65-72, doi: 10.1109/ICTAI.2018.00021.

- Pavithra C P , Shibily Joseph, Detection and Verification of Rumour in Social Media: A Survey

- Shihan Wang, Takao Terano, Detecting rumor patterns in streaming social media

- Tian L., Zhang X., Wang Y., Liu H. (2020) Early Detection of Rumours on Twitter via Stance Transfer Learning. In: Jose J. et al. (eds) Advances in Information Retrieval. ECIR 2020. Lecture Notes in Computer Science, vol 12035. Springer, Cham. https://doi.org/10.1007/978-3-030-45439-5_38

- Zubiaga, A.; Liakata, M.; Procter, R.: Exploiting context for rumour detection in social media. In: Ciampaglia, G.L., Mashhadi, A., Yasseri, T. (eds.) Social Informatics, pp. 109–123. Springer, Cham (2017)

- Alsaeedi, A., Al-Sarem, M. Detecting Rumors on Social Media Based on a CNN Deep Learning Technique. *Arab J Sci Eng* (2020). https://doi.org/10.1007/s13369-020-04839-2