Deep Learning (COSC 2779) – Assignment 2

Yogesh Haresh Bojja (S3789918)

1. Problem Definition and Review : In this assignment we have to achieve stance classification on twitter dataset and categorize the tweets into 4 categories i.e., Comment, Query, Deny, Support.

2. Approach of Modelling : We will be using Electra-small BERT as the pretrained model for sentence classification and the output of BERT model will serve as the input of CNN and will help to predict the stance of the tweet.

3. Evaluation plan of the model: For evaluating our model we have used categorical accuracy as we are dealing with multi classification problem. Our data is imbalanced and to solve it we have Random over sampling technique hence we prefer to use accuracy instead of F1-score. We will be evaluating our model on the test set. We have split the twitter dataset into 70% train, 15% validation and 15% test data.

4. Detailed(step-by-step) explanation of the modelling:

a) Importing the data : We have used twitter dataset for training the model

b) Data Cleaning and Preparation : In data cleaning we have created a new column 'Source' which contains source tweet of the reply-tweet. We have removed twitter handles and URLs mentioned in the tweets. We have kept #, ?, ! in the tweets. Words in upper case is often used to showcase the strong emotions hence except those we have converted the whole string to lower case. Stop words like a, an etc. are trivial hence its better to remove them. We have converted the class column(query, comment, deny, support) to (0, 1, 2, 3). Data is imbalanced as shown below in fig1. We have done random over sampling on it. Overfitting generated by over sampling can be reduced by regularization and dropout further.

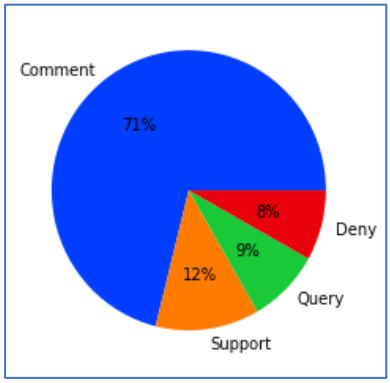


Fig 1

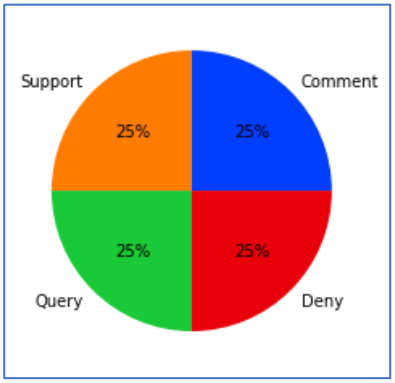


Fig 2

c) Creating Base Model : We have selected Electra-small as the pre-trained model. Pooled output from the BERT encoder is given to the dense layer with 4 units as there are 4 classes i.e., SDQC. Pooled output gives encoding of the whole string. After training the model, we evaluate the base model on test data and the accuracy is around 45% as shown below.

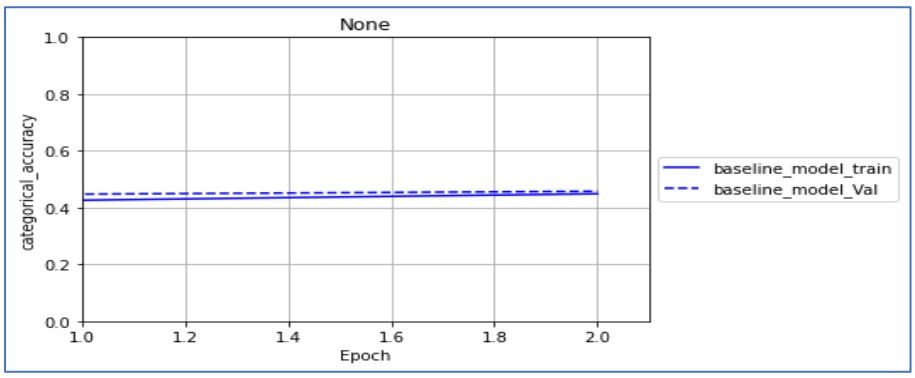
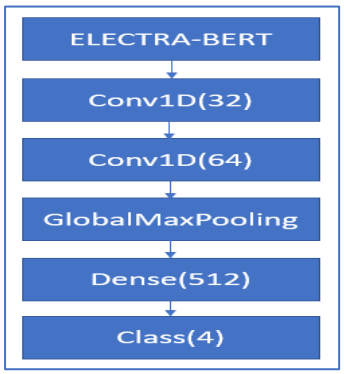


Fig 3

d) Creating Complex Model :



We will be Embedding CNN on top of the Electra model. CNN extracts the important features from the data. Hence we will use this technique to extract feature from the data. Sequence output gives the encoding for each word. We will feed sequence output to the convolution layer with 32 filters and then through the convolution layer with 64 filters. In order to reduce the data size and make it computation friendly we are applying global max pooling on it. Output of global max pooling will be given to MLP. MLP consists of 512 units in the first hidden layer and 4 units in the output layer.

e) Select Activation : We have tested the model on sigmoid, elu, relu, and leaky relu out of which elu gave the best result.
f) Select Regularization : Among L1 and L2 regularization, L2 regularization performed best hence we will select it for the further analysis.
g) Select Dropout : We tested the model with dropout 0.1 and 0.3. Model with dropout 0.3 performed better than the other one. Hence we will select the value of dropout equal to 0.3.
h) Selecting Optimizer : BERT uses 'adam' implicitly hence we will be using 'adam' for our model.
i) Evaluation on test data : Once the model trained we are predicting the classes of test data and evaluating it. Fig 4 shows training curve and validation curve. Fig 5 shows the confusion matrix generated by prediction over the test data.

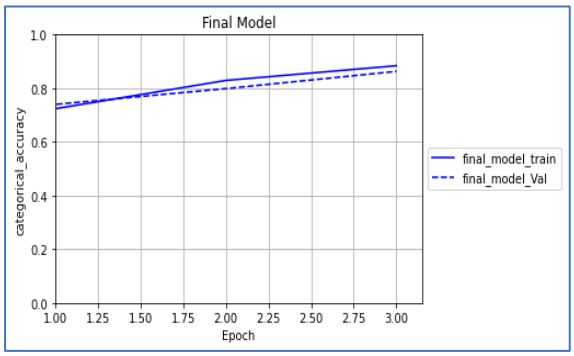


Fig 4

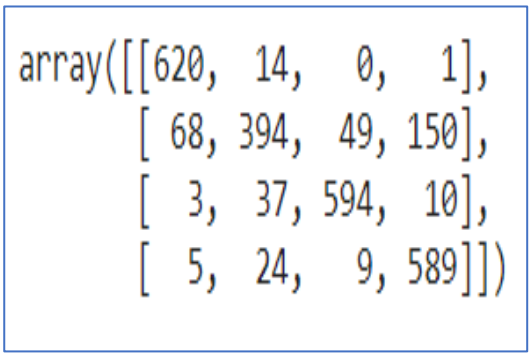


Fig 5

j) Evaluation on Independent dataset : For evaluating our model we have used Extra_reddit_dataset.csv and 3904163061532433784.csv from (COVID-19-rumor-dataset/Data/twitter at master · MickeysClubhouse/COVID-19-rumor-dataset (github.com)). We have got over 60% accuracy for both the testings.

Accuracy for Reddit dataset

```
75/75 [==============================] - 17s 221ms/step - loss: 1.1736 - categorical_accuracy: 0.6499
Loss =  1.1735889911651611
Accuracy =  0.6499372720718384
```

Accuracy for rumor dataset

```
10/10 [==============================] - 2s 243ms/step - loss: 1.2538 - categorical_accuracy: 0.6242
Loss =  1.2537873983383179
Accuracy =  0.6242038011550903
```

5. Justification :

- **Why Electra ? :** Electra has highest glue dataset percentage of 85.1. It is computationally very less costly as compared to ROBERTA and other BERT models. BERT only trains on the masked words but Electra is trained on each word by corrupting it.

- **Why Sentence pair approach ? :** Source tweet has connection with the tweets replied to it. Hence we have concatenated source tweet with the reply tweet by '|' operator.
- **Why we discarded '@' and kept '#,!?' in tweets :** Twitter handle starts with @ and twitter handle does not contribute to the information for text classification, hence, we discard it. On contrary, Tags are very important in grouping of the tweets for example tweets with tag #Peace can be identified in one group. Tags start with #, hence we do not discard them. ! is useful for identifying a tweet as comment, whereas ? is useful to classify queries.
- **Why did we change the case of sentence to lower except for those words in upper case ? :** It is been found out that people try to express their strong feeling by writing it in upper case for example "people need to STOP RACISM". Therefore, we convert sentences in lower case but words in upper case are left as it is.
- **Why oversampling ? :** Twitter data is imbalanced, to solve this issue we can apply SMOTE, ROS and RUS. SMOTE is computationally heavy as it simulates random samples and apply KNN to it. SMOTE can also give ambiguous results for data points falling close to two groups. ROS is oversampling technique and its fast. Problem with ROS is that it overfits the model, but we can solve this by introducing dropout and regularization.
- **Why sequence output selected ? :** Instead of pooled output we have selected sequence output. Pooled output gives the encoding for each sentence whereas sequence output gives encoding for each word in the sentence. Thus, I have preferred to take encoding for each word as we are applying CNN which will extract important words from the sentences.
- **Why CNN ?** : CNN extracts important features from the data. We have data with encodings of all the words in it hence CNN can identify important words from the sentences.

Ultimate Judgement : The final model gives good performance on the test data as compared to the baseline model with same epochs. Hence, I would recommend using final model for stance classification. When we perform testing on independent dataset (Reddit&Covid) accuracy comes down to 65%. It is also seen that model learns more how to classify comments rather than support, deny, and query.

Limitations : Most of the datasets found online is imbalanced having highest number of comment classes. Hence model is trained mostly on comment class.

Literature Review :

1. **Convolutional Neural Networks for Sentence Classification [1]**
   Findings : Simple CNN model with little hyper parameter tuning can give good results than other complex models.
2. **Rumour Stance Classification using A Hybrid of Capsule Network and Multi-Layer Perceptron [2]**
   Findings : This paper also mentions use of CNN for stance classification and performs good. Importance of words with upper case in the tweets, importance of '?' and '!' is mentioned in this paper.
3. **Revisiting Rumour Stance Classification: Dealing with Imbalanced Data [3]**
   Findings : Different techniques for solving problem of imbalanced dataset is mentioned in this paper. Random under sampling(RUS), Random over sampling(ROS), SMOTE, TM are discussed here. RUS loses important data; ROS solves this problem, but it introduces overfitting.
4. **eventAI at SemEval-2019 Task 7: Rumor Detection on Social Media by Exploiting Content, User Credibility and Propagation Information [4]**
   Findings : Importance of # and ? in the tweets is mentioned in this paper.
5. **SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours [5]**
   Findings : When source tweet and reply tweet both are passed to the model, model gives good results. This is called sentence pair approach. This paper has training using sentence pair approach.
6. **ELECTRA: PRE-TRAINING TEXT ENCODERS AS DISCRIMINATORS RATHER THAN GENERATORS [6]**
   Findings : This paper shows how Electra outperforms all the models like ROBERTA, BERT, ALBERT etc. Electra uses ¼ compute of what ROBERTA and XLNET uses and still performs better than them. GLUE score of Electra is highest among all i.e., 85.

Bibliography :

[1] (2021). Retrieved 19 October 2021, from https://arxiv.org/pdf/1408.5882.pdf

[2] View of Rumour Stance Classification using A Hybrid of Capsule Network and Multi-Layer Perceptron. (2021). Retrieved 19 October 2021, from https://turcomat.org/index.php/turkbilmat/article/view/9398/7232

[3] (2021). Retrieved 19 October 2021, from https://aclanthology.org/2020.rdsm-1.4.pdf

[4] (2021). Retrieved 19 October 2021, from https://aclanthology.org/S19-2148.pdf

[5] Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Hoi, G., & Zubiaga, A. (2021). SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. Retrieved 19 October 2021, from https://arxiv.org/abs/1704.05972

[6] Clark, K., Luong, M., Le, Q., & Manning, C. (2021). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. Retrieved 19 October 2021, from https://arxiv.org/abs/2003.10555