Deep Learning (COSC 2779) – Assignment 2 – 2021

Karthi Narendrababu Geetha (S3835901)

1 Problem Definition and Review

Automating the detection of 'fake news' associated with the news in social media is an important task in order to prevent the spread of rumours amongst the people. The problem statement involves in the classification of response tweets given the source tweets corresponding to their stances such as 'support', 'deny', 'query', 'comment'.

2 Dataset Provided

- Dataset is a part of a workshop named Semantic Evaluation on NLP, which is a series occurring every year. The given dataset is gathered in the workshop occurred in 2017.
- Two sets of social media data were provided such as Twitter and Reddit.
- Twitter data comprises of 6634 tweets whereas the Reddit data comprises of 2471 tweets.
- Both the datasets were combined, since the provided data is very less.
- Since the comments class in both the dataset is higher, the class is dropped from reddit dataset while merging.
- The length of tweet's texts is between 10 and 480.
- As most of the tweets were classified as 'comment', there is a huge imbalance between the classes.
- Not all the topics have all the classes, however all of them have comment class.
- Also, there is no specific topic name provided in the reddit dataset.

3 Evaluation Framework

3.1 Loss Function – Categorical Cross Entropy

The given problem statement is a multi-class classification, and hence the Categorical Cross Entropy loss is being used for the whole development process.

3.2 Metrics

Since there is huge data imbalance in the given dataset, the model's performance measure should not be fully based on the accuracy. Hence, the macro averaged F1 score is being used as a performance measure. Also, the precision and recall rate for each epochs is being calculated.

3.3 Epochs

At the Initial stages, the number of epochs used for the problem is 30, however it is identified that 30 epochs are over training the model and hence it has been reduced to 15 epochs.

3.4 Optimizer

As per the literature survey, the best optimizer to be used is the Adam optimizer and hence it has been used for the whole process. Adamw is also considered, however the performance measure from both the optimizer is equal and hence Adam optimizer is used for the whole experiment.

4 Approach & Justifications

4.1 Approach:

Approach for this problem is categorized into 2, deriving solution with and without pre-processed text. The reason for this approach is that, it is identified that the tokenizer is able to handle punctuations and the symbols, also the twitter data is abundant of symbols and punctuations.

4.2 Class Imbalance

With the help of EDA analysis, it is identified that there is a huge class imbalance problem with the given data where most of the data are skewed towards the 'comment' class. Hence the following Experiments were tried:
Data Augmentation, Class Reweighting, Learning rate scheduler, increasing the Dropout and decreasing the training layers in the model.

4.3 Handling Source text

The dataset consists of both the source and reply texts that are available, however for this problem the source text will not be useful. Instead of dropping the whole source texts, they were reused. If the source text contains any '?', then the texts were annotated as 'query' class and if it contains negative words & an '!', then the texts were annotated as 'deny' class and every other texts as 'support' class.

4.4 Data Preprocessing

- Created a dictionary of apostrophe words with the corresponding full words, for example {'won't' : 'will not'} and replaced all the words correspondingly.
- With the help of regular expressions parsing functions from 're' package, the 'http' & 'https' URLs were replaced with blank space.
- Python's String.punctuation variable contains all the punctuations and hence with the help of that variable the punctuations has been replaced with the blank space.
- The NLTK package contains the array of Stopwords and from that array. The negative words such as (not, nothing, etc.,) were removed from the stopwords array. Then the remaining stopwords were removed from the whole dataset.

4.5 Model Architecture

4.5.1 Base model

4.5.1.1 Tokenizing:

In order to convert the sentences into words using the Tokenizer method from keras package is being used. With the help of tokenizer object the sentences in the training and testing dataset has been tokenized, if the word is not available on the object then it will be annotated as OOV Token and assigned an index 1. Then the tokens were converted into sequence of numbers based on the index numbers from the tokenizer object. Also, the sequences are post padded to certain length.

4.5.1.2 Embedding layer:

The input dimension of the embedding layer is the total number of words in the tokenizer's **word_index+1** and the output dimension is set to 64 which outputs a 3-dimensional tensor. Most of the texts length is between 10 & 50, however overall texts length is between 10 and 490 and hence the sequence of texts is embedding into the size of 200 which does not lose much data from the overall data. Also, the weights of the data retrained by setting the 'trainable' to true.

4.5.1.3 Bidirectional LSTM layer:

Bidirectional LSTM is considered for this problem, as the unidirectional LSTM will not be able to predict the sentence from both forward and backward directions. The sequence of 2 bidirectional LSTM layers is resulted in overfitting of the model. The number of units in the layer is considered between 32, 64 and 128. However higher the increase in the layer's units, the model's complexity increased.

Along the above layers, 2 Dense layers with the units between 32, 64 and 128 is being added to model and 2 dropouts with 50% of dropout rate, inorder to reduce the model's complexity.

4.5.2 BERT model

4.5.2.1 Tokenizing:

From the hugging face transformers package, the AutoTokenizer class gets pre-trained word models such as Bert, XLTokenizer, etc. Then the words are converted to sequence of numbers using the pre-trained tokenizer. The tokenizer returns the sequence of numbers corresponding to each word and the attention mask, using which the model gives attention to those values, if the attention mask is 1. The special characters were also added to distinguish the starting and ending of the sequences. Hence the output of the tokenizer consists of inputs and the attention mask.

4.5.2.2 Embedding layer:

Pretrained embeddings from the Bert model is being used where the input to the layer is the tensor array of inputs and attention mask from the tokenization process. Other layers are exactly similar to the base model.

4.5.3 Glove model

4.5.3.1 Embedding layer:

Tokenization process for this model is similar to the base model. The embedding vectors for each word in the given dataset are obtained from the pretrained glove text file and then feed them as weights to the embedding layer. Other layers are exactly similar to the base model.

5 Experiments & Tuning

5.1 Data Augmentation

5.1.1.1 Synonym Replacement:

Wordnet is a lexical database of English. The synonym lemmas for each word can be obtained with the 'synsets' method from wordnet. The new set of data is being created with different synonyms for around 3 words, for the minority classes in order to solve the class imbalance problem.

5.1.1.2 Word Swapping:

Words are randomly swapped from the sentences of minority classes and created a new of data.

5.2 Transfer Learning

Since the provided dataset is low, transfer learning is one of the best options to obtain better performance from the model and so pretrained models such as BERT and GLOVE is used as an experiment. However, BERT model improves the performance of the model.

5.3 Class Reweighting

Class reweighting is the concept of influencing the loss function to set higher cost to the data from minority classes. It is done with the help of compute_class_weight function the scikit learn package, which computes the class weights using n_samples / (n_classes * np. Bincount(y)).

5.4 Dropout

Dropout is a regularization technique, where the randomly selected convolutional neurons will be ignored during the training process, which helps the model to be more generalized. 50% of Dropouts were used to reduce model's complexity, thereby reducing the overfitting of data.

5.5 Learning rate scheduler

With the help of Learning rate scheduler, the optimizer's learning rate is modified during the training process and it works based on the loss function values from each epochs. Scheduler, increases the performance of the training data, however the performance on the validation data is not improved

6 Ultimate Judgment, Analysis & Limitations

6.1 Judgement:

6.1.1 With Pre-processing (Appendix 8.1):

- Since the provided dataset is low, performance of the base model is not improving, also it overfits a lot. Tried to reduce model complexity, added more data, however the overfitting problem is not reducing.
- Tried BERT based transfer learning, the performance of the model does increases but it still doesn't reach expected score.
- Need to try building a model without pre-processing steps, in a way the lot of data will not be lost.

6.1.2 Without Pre-processing (Appendix 8.1):

- Although the base model's accuracy is higher, the performance of the model is very low, the model is clearly overfitting. The length of comments in the dataset is much higher and hence the model was only able to predict the comments.
- GLOVE model was able to provide better performance while training, however it's performance on the unseen data is lower.
- BERT model does performs well, because of imbalance in the data, the model is slightly overfitting. Even the learning rate scheduler unable to reduce the overfitting problem.

6.2 Final model:

On comparing the performance of 3 models (Bi-LSTM, GLOVE, BERT) without the pre-processing steps, BERT model is considered as the best model for this problem. Because of the data imbalance problem, the model was slightly overfitted towards the comments, however the loss is very much less. The F1-Score on the unseen data is around 50 % – 53 %, the model's performance was fluctuating as some of the texts were easy to classify.

6.3 Limitations:

- Model's performance on predicting 'comment' & 'support' classes is higher
- On analysis, it can be seen that if the sentence is larger and the sentence have a question mark, the model is able to predict that sentence as a query.
- However if the sentence is small, even though the sentence has a question mark, it is either classified as comment or support.
- Model's performance is very poor on 'deny'.

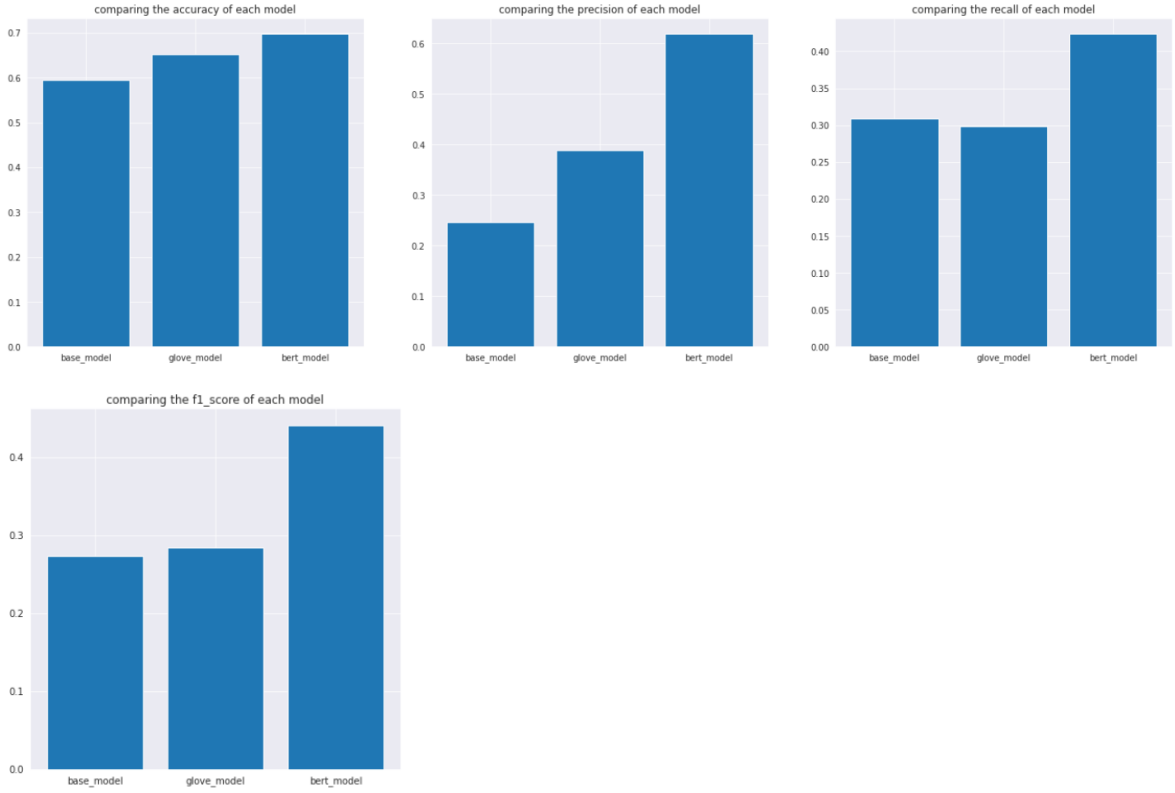
7 References

1. Derczynski, L., Bontcheva, K., Liakata, M., Procter, R., Wong Sak Hoi, G. and Zubiaga, A., 2017. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, [online] Available at: <https://aclanthology.org/S17-2006> [Accessed 19 October 2021].
2. Gorrell, G., Kochkina, E., Liakata, M., Aker, A., Zubiaga, A., Bontcheva, K. and Derczynski, L., 2019. SemEval-2019 Task 7: RumourEval, Determining Rumour Veracity and Support for Rumours. Proceedings of the 13th International Workshop on Semantic Evaluation, [online] Available at: <https://aclanthology.org/S19-2147> [Accessed 19 October 2021].
3. Rajendran, G., Chitturi, B. and Poornachandran, P., 2018. Stance-In-Depth Deep Neural Approach to Stance Classification. Procedia Computer Science, [online] 132, pp.1646-1653. Available at: <https://www.sciencedirect.com/science/article/pii/S1877050918308640> [Accessed 19 October 2021].

4. Fabien, M., 2020. Data Augmentation in Natural Language Processing. [online] Available at: <https://maelfabien.github.io/machinelearning/NLP_8/#> [Accessed 19 October 2021].
5. Briggs, J., 2020. *Build a Natural Language Classifier With Bert and Tensorflow*. [online] Medium. Available at: <https://betterprogramming.pub/build-a-natural-language-classifier-with-bert-and-tensorflow-4770d4442d41> [Accessed 19 October 2021].
6. Team, K., 2021. Keras documentation: Using pre-trained word embeddings. [online] Keras.io. Available at: <https://keras.io/examples/nlp/pretrained_word_embeddings/#download-the-newsgroup20-data> [Accessed 19 October 2021].
7. Gupta, R., 2021. Build and Compare 3 Models—NLP Sentiment Prediction. [online] Medium. Available at: <https://towardsdatascience.com/build-and-compare-3-models-nlp-sentiment-prediction-67320979de61> [Accessed 19 October 2021].
8. Nltk.org.. NLTK :: nltk.corpus.reader.wordnet module. [online] Available at: <https://www.nltk.org/api/nltk.corpus.reader.wordnet.html?highlight=wordnet#module-nltk.corpus.reader.wordnet> [Accessed 19 October 2021].
9. Huggingface.co.. The tokenization pipeline — tokenizers documentation. [online] Available at: <https://huggingface.co/docs/tokenizers/node/latest/pipeline.html> [Accessed 19 October 2021].
10. Cheng, M., Wang, S., Yan, X., Yang, T., Wang, W., Huang, Z., Xiao, X., Nazarian, S. and Bogdan, P., 2021. A COVID-19 Rumor Dataset. Frontiers in Psychology, 12. Available at: < https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8200409/>

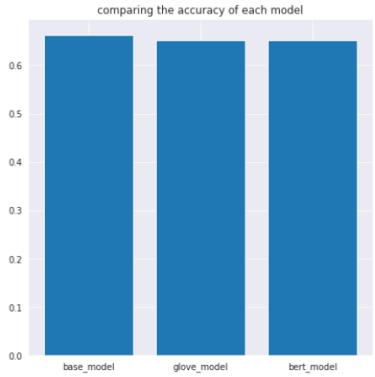
8 Appendix

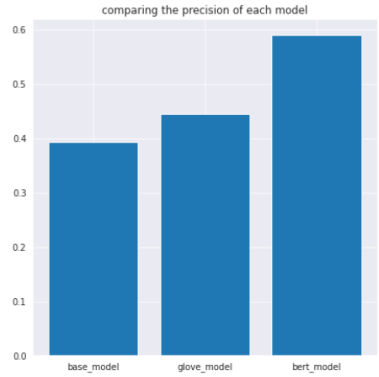
8.1 With Pre-processing:

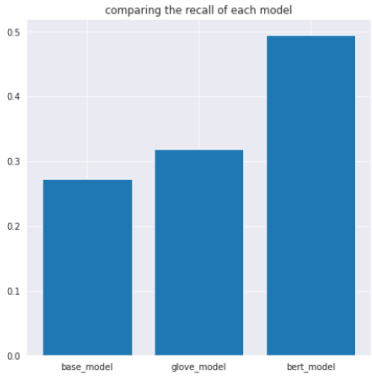


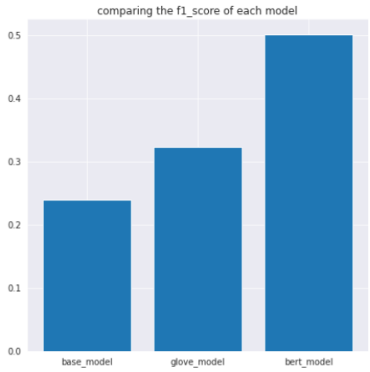

```
Text    @ nijatk @ vegasrebs have fun defending your home with a baseball bat while waiting 30 minutes for the cops to show up.
Original        3
Predicted       3
Text    @ rockrt66 @ nbcnews true. however if clothes were different or the same would do it also - i doubt that he the time to change.
Original        3
Predicted       3
Text    @ marcepa49 @ flightradar24 @ isobelroe it was obvious it was being manually controlled by the changes in descent rate. also no
Original        3
Predicted       3
Text    @ sandratxas @ carrieksada @ specialkmb1969 @ choosetobfree @ lrihendry @ phil200269 @ steph93065 @ drmartyfox
Original        3
Predicted       0
```

8.2 With Pre-processing:










```
Text    @ MarkyMark5D @ PrisonPlanet Bill knows
Original        3
Predicted       3
Text    But, but, but... looters? Virgin Islands Allows National Guard To Seize Guns, Ammo Ahead Of Hurricane Irma
Original        1
Predicted       0
Text    @ DIANAZOGA @ AntonioFrench How were the police so quick to indicate Mike Brown if all of this took place in a matter of minutes?
Original        1
Predicted       3
Text    @ SandraTXAS @ carrieksada @ SpecialKMB1969 @ ChooseToBFree @ Lrihendry @ phil200269 @ steph93065 @ DrMartyFox thank you for the valid
update......
Original        3
Predicted       3
```