Fake News Detection Task

Submission by:

Harsh Grover

Affiliation:

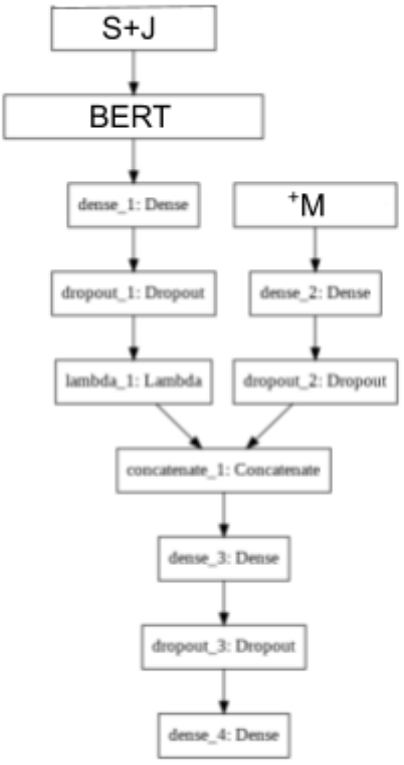
Birla Institute of Technology and Science, Pilani

Results (Best Validation[1] Accuracies)

Binary classification task - **70.3%**

Six-way classification task - **37.2%**

Approach



My model has a SJ branch for *statement(S)* and *justification(J)* and a +M branch for *metadata*. Finally output of these two branches were concatenated and passed to the classification branch. Took 6-7 hrs to train each model on GPU (Google Colab).

SJ branch:

I used BERT-base as a self-attention mechanism between the *statement(S)* and *justification(J)*. I found their relation analogous to question-paragraph[2] relation in Question Answering task. Question is related to the paragraph and the output is the answer, which is information from paragraph subjected to the question. Similarly, here statement is related to the justification and we need information (truthfulness) from the statement subjected to the justification. Hence, I used a similar approach. Finally used a dense layer to compress the output of BERT.

+M branch:

I one hot encoded the categorical variables and embedded the context using bert embeddings. Then I combined all the encoded metadata into a single vector. Finally, I input this vector into a neural network of dense layer to compress the information into lower dimension vector.

Classification branch:

I concatenated the outputs of the two branches to form a feature vector, which is then passed through another dense layer before the final sigmoid/softmax layer.

Training parameters:

Dataset - LIAR-Plus. I used half of the training set. 80% of which is used in training, 20% in validation.
Optimizer - AdamWarmup (Adam[3] optimizer with a warmup rate, recommended for training BERT)
Batch_size - 5, Epochs - 40, Loss - Crossentropy (Binary/Categorical)

[1]Validation set wasn't used for training at all, i.e. there was no leakage.

[2]Paragraph from which the question is asked.

[3]Stochastic Gradient Descent

Other approaches

1. I tried using a Bidirectional GRU after BERT instead of Dense. It just slowed down the training and degraded the accuracy a bit.
2. I tried using different branch for every metadata, concatenated their output, passed it through a GRU and then used it's output with SJ branch for classification. No improvement observed, slight degradation in accuracy and training was terribly slow.
In both the cases above, I didn't get a chance to train long enough to see
3. Bidirectional Attention Flow (BiDAF) mechanism should also work, given it's success in question answering task. Due to limited time I couldn't try that. It can be used instead of BERT if less parameter intensive model with faster inference time is required.

Libraries used/cited

- keras
- tensorflow
- keras-bert: https://github.com/CyberZHG/keras-bert (pip install keras_bert)
- sklearn
- Pandas
- Numpy
- NLTK
- re
- os
- matplotlib
- codecs
- livelossplot: https://github.com/stared/livelossplot (pip install livelossplot)
- tqdm