IE643: Deep Learning: Theory and Practice **July-Dec 2021**

ColBERT: Using BERT Sentence Embedding for Humor Detection

Team Name: Team YOLO *Team Members: Gopalan Iyengar, Om Prabhu*

Abstract

This project report contains the details on our project which aims to detect humor in a given excerpt. We describe the literature review of the ColBERT model and some of its precursors. We also describe the deep learning models and the associated training heuristics and optimization strategies involved in the original paper, as well as the improvements suggested by us. We have implemented convolutional layers in place of the pre-existing dense layers in the model, and superior results are obtained with respect to the number of parameters involved, and thus, the model training time, with a slight decrease of approximately 2% accuracy from the presented 98% in the paper.

1 Introduction

Humor is a major component of our daily lives. If we are ever to strive for chat bots, or intelligent agents which must communicate with humans in order to understand and perform their objectives, there is a need for an interface between the two which can accurately grasp the underlying intended humor in the user's speech. This agent could also be trained to use such humour in conversation with a human, which would make the user experience more comfortable and interesting.

To that extent, a deep neural network is trained as a humor classifier on sentence embeddings which provide a numerical basis for a computer to understand languages. The ColBERT model uses dense layers for this task, whereas in this paper, we will compare those results with our model's results, which uses convolutional layers in their place.

We also propose an extension model concept, which greatly reduces the training time and number of parameters involved during training.

We provide a literature survey in Section 2. Our proposal for the project is described in detail in Section 3. We details on experimentation in Section 5. A description of follow-up work which could be implemented is given in Section 7. We conclude with a short summary and pointers to forthcoming work in Section 8.

2 Literature Survey

Our project draws inspiration heavily from work by Annamoradnejad et al.[1]. In their paper, the authors describe a model where an input string is fed into a transformer model BERT[3] which generates embeddings for the sentences. The input contains a maximum of 5 sentences. Each sentence is fed into BERT separately, along with the entire text, which produces six embedding features. These features are then used in an eight-layered parallel dense (fully-connected) network, which is used as a classifier. This process is illustrated in Figure 1.

The authors discuss how the parallel network extension to BERT represents the model's intuition of the linguistic structure of humor in a passage. An analogy is drawn to how a humorous excerpt consists of a few 'set-up' sentences followed by a 'punchline', none of which alone characterize the humor in the passage. Only in each others' context do the sentences seem funny.

The authors also published a dataset (ColBERT dataset) of 200,000 labelled excerpts containing equal amount of humorous and non-humorous samples. We have made use of this dataset in our work.

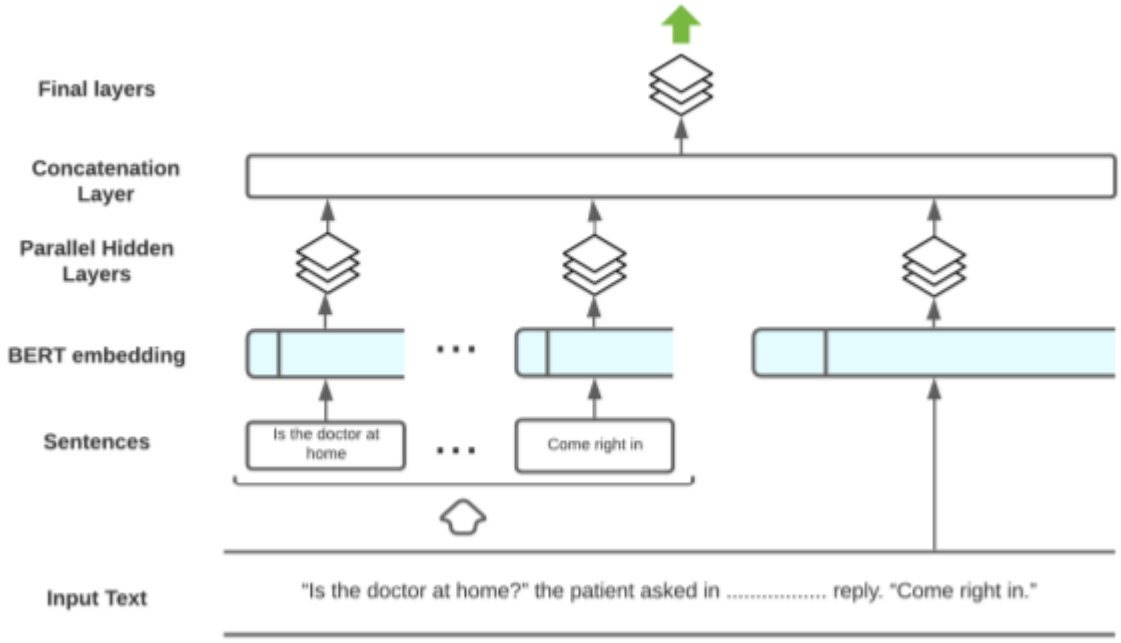


Figure 1: ColBERT Model representation given in the original paper

Another intriguing research paper we referred to was Humor Recognition Using Deep Learning, Chen and Soo[2]. This paper utilizes convolutional neural networks with varying filter sizes and usage of highway layers to predict presence of humor in a sample of text using GloVe embeddings[6]. This multilingual model had high prediction accuracy (approx. 90%) in both English and Chinese humor samples.

We observed that CNNs show much promise in humor detection tasks from Chen and Soo's work.

We also reviewed Humor Detection: A Transformer Gets the Last Laugh, Weller and Seppi[7]. This paper describes a model consisting of BERT to generate embeddings for the entire sample, followed by a single dense layer with a softmax activation. This model's performance (approx. 93% accuracy) surpasses Chen and Soo's work described above on multiple datasets.

We noticed that the embeddings generated by BERT (State of the Art transformer model) were much better at encoding textual/linguistic relationships in a numerical space than other embeddings.

In an effort to combine these concepts and integrate them with the concept of the linguistic structure of humor discussed in the ColBERT paper, we propose a change to the structure of the ColBERT model, where we replace the dense layer extension to BERT with a parallel CNN extension.

3 Methods and Approaches

The original ColBERT model is shown in Figure 2. It takes 3 feature vectors of size 20 per sentence as input, and another 3 of size 100 corresponding to the entire excerpt. These features are generated using BERT's pre-trained tokenizer with maximum sequence length of 100. Using these 18 feature vectors, we generate embeddings using BERT and perform Global Average Pooling, providing us 6 embedding feature vectors of size 768, one for each sentence.

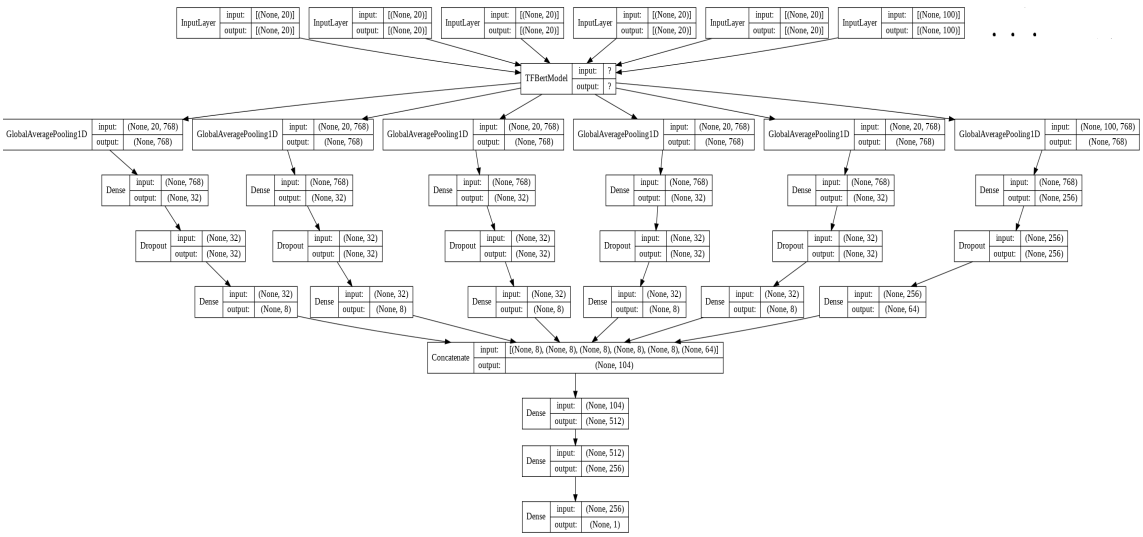


Figure 2: ColBERT Model detailed

Notice the six parallel dense layers, which are eventually concatenated and processed in further dense layers. These represent the separate processing and featurization of each sentence's struc-

ture. Dropout layers are used extensively in the original model to prevent over-fitting and increase regularization.

Since we are not training the BERT Head of the model, we separate the extension layers from the BERT layers, saving the embeddings to a file and using them as features for only the extension model. This is illustrated in Figures 3 and 4.

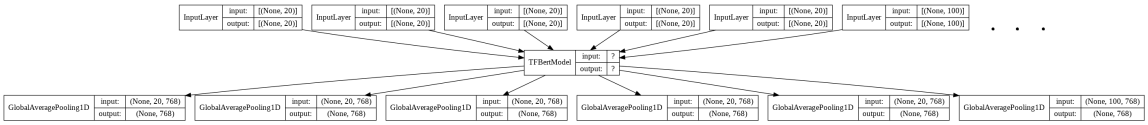


Figure 3: BERT Head

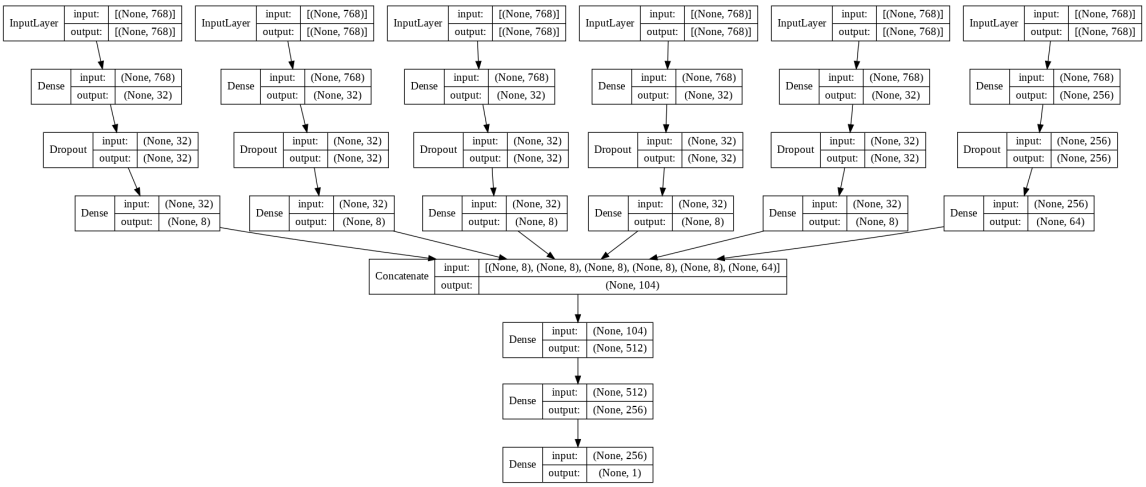


Figure 4: Original Extension to BERT

Henceforth, all the models we describe will use the Global Average Pooled embeddings generated by the BERT Head as features. We refer to these as extension models. Our goal now is to replace the original extension model to BERT with a convolutional extension model.This 'extension model' heuristic greatly reduced the average training time from 2.5 hours per epoch to approximately 1 hour per epoch for the dense extension layer. Multiple extension models have been tested, described in Section 3.2.

3.1 Work done before mid-term project review

- Gained broad understanding of the problem statement and solution proposed in the ColBERT paper.
- In-depth analysis of the dataset collected and published by the author of the ColBERT paper.

- Learned about the model technical details and the associated training heuristics implemented.
- Extensive literature review of the three papers described in Section 2.
- Exploratory data analysis and initial rudimentary implementation of code.

.

3.2 Work done after mid-term project review

- Conducted some initial experiments with TensorFlow's default Natural Language Processing toolkit and shallow dense models, achieving 73% test accuracy.
- Implemented ColBERT as a black box for testing accuracy on a test dataset (original model).
- Implemented ColBERT in Python using dense layers (original model).
- Implemented multiple BERT extensions for humor classification in Python using convolutional layers (proposed model).
- Trained and tested these models on the original ColBERT Dataset.

The significance of choosing convolutional layers can be explained by an analogy similar to that described by the author of the ColBERT paper. The author describes a setup for parallel processing of each sentence, and the entire passage separately. This was to capture the sentence-wise linguistic structure of the joke.

Similarly, in our implementation, we have a setup for parallel sentence processing for this purpose, as well as the feature extraction (convolutional) setup to extract local relationships between words, which appear in a sequential manner and hold a special relationship with each other based on their ordering too. This helps our model better generalize over the sentence-wise and sequential structure of humorous passages.

Our models make use of 1D convolutional layers as our embedding features are 1 dimensional. Pre-concatenation convolutional layers are padded to preserve feature size in convolution. Max pooling layers with a maximum kernel size of 4 are used after most pre-concatenation convolutional layers. These perform the task of downsampling the extracted features. Post-concatenation convolutional layers are not padded, and are downsampled by convolution. The first five parallel layers are similar, and the final layer is a little less downsampled, as it holds information about the entire passage as opposed to the other parallel layers individual sentences. The final layer is also a convolutional layer which behaves as a fully connected layer, with sigmoid activation. All other activations are ReLU.

The following convolutional extension models were developed and tested on the ColBERT Dataset. Dropout was used extensively in all the models following suit from the original model, but they are absent from the images for concise representations. The dense extension model contains 523,017 trainable parameters. The models were developed and improved in the same order they are listed:

- **Single Filter Model**: Henceforth referred to as Model 'A'. All kernels used were of depth 1, i.e one filter for each feature which was subjected to convolution. This model was developed by trying to emulate the depth and feature sizes of the dense model, and some trial and error. This model has 524 trainable parameters.

- **Multiple Filter Model**: Henceforth referred to as Model 'B'. Multiple filters were used for one feature undergoing convolution, for all intermediate layers before concatenation. All kernels used after the concatenation were of depth 1. Another convolutional layer was added before the concatenation layer. This model was developed by increasing the complexity of Model A, and trial and error. This model has 4,041 trainable parameters.
- **Maximum Parameters Model**: Henceforth referred to as Model 'C'. This model was built as an improvement to Model B, and contains two extra convolutional layers: one after the input layer, and another in the post concatenation layers. More features were allowed to pass through the concatenation to increase the 'concatenated feature' contribution. Still, all kernels used after the concatenation were of depth 1, i.e one filter for each feature which was subjected to convolution. As the name suggests, this model has the largest number of trainable parameters, 19,630.
- **Best Performing Model**: Henceforth referred to as Model 'D'. Multiple filters were used for one feature undergoing convolution, for all intermediate layers before **and** after concatenation. This model was developed by trying to decrease the number of parameters in Model C, while still giving more learning capability to the concatenated features. This model has 17,220 trainable parameters.

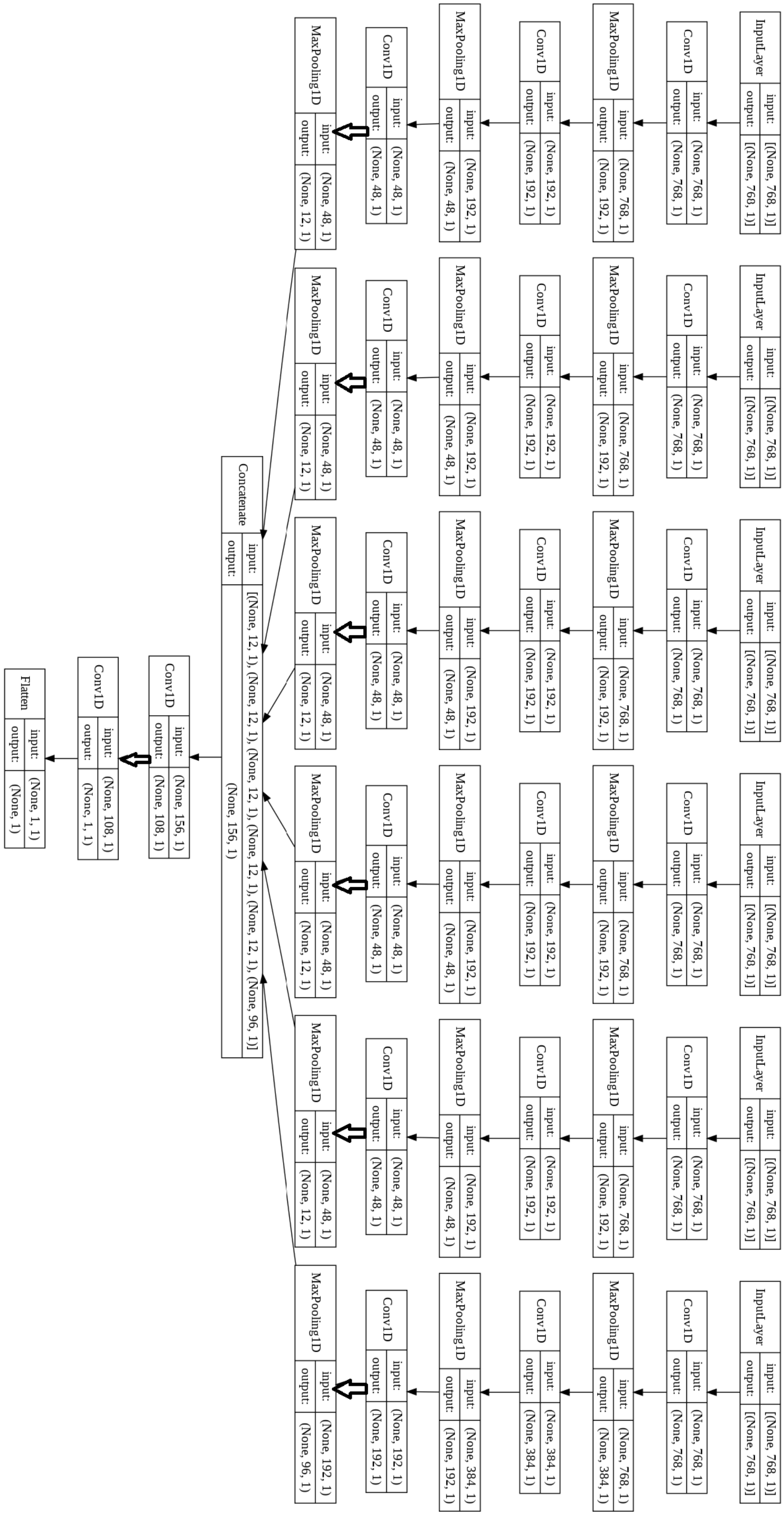


Figure 5: Model A

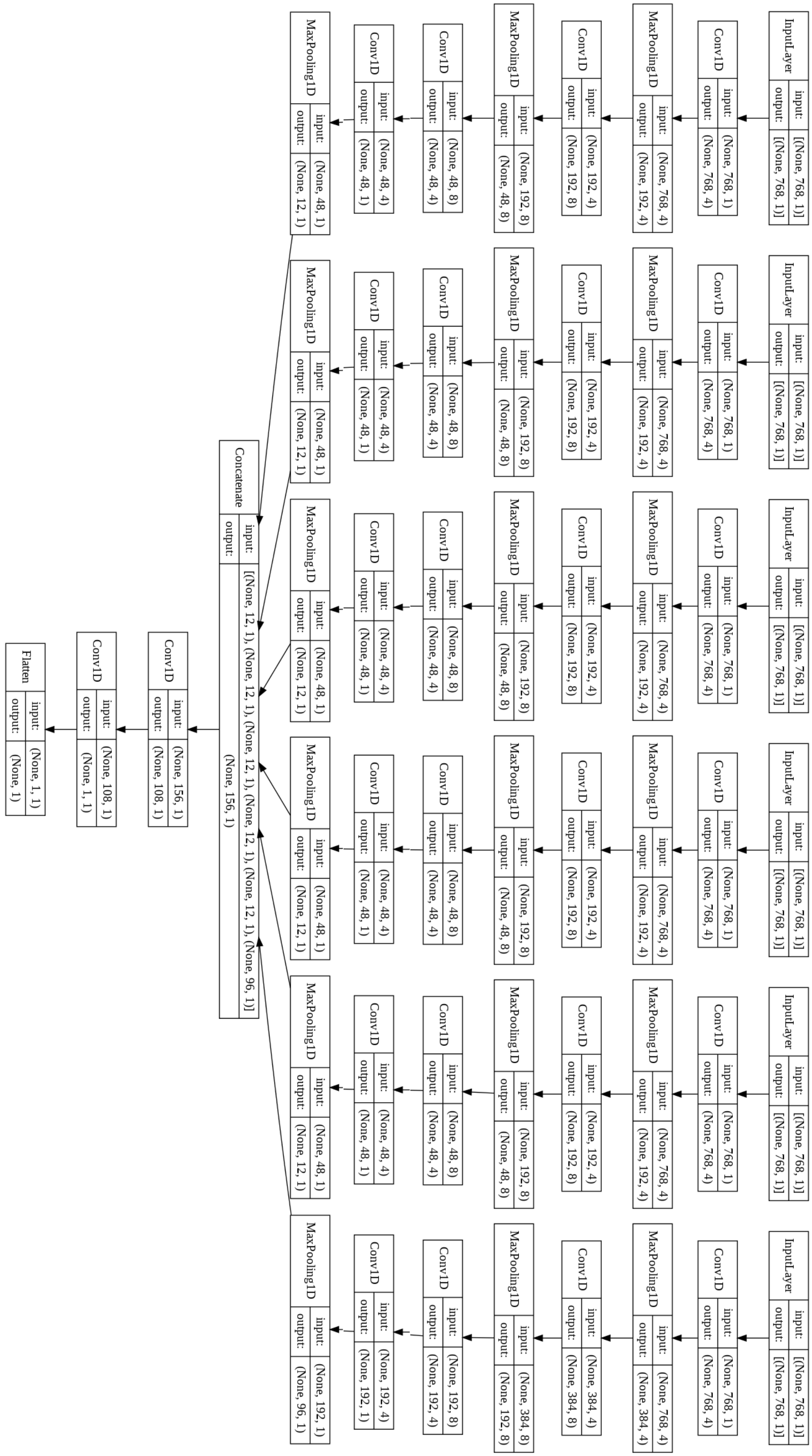


Figure 6: Model B

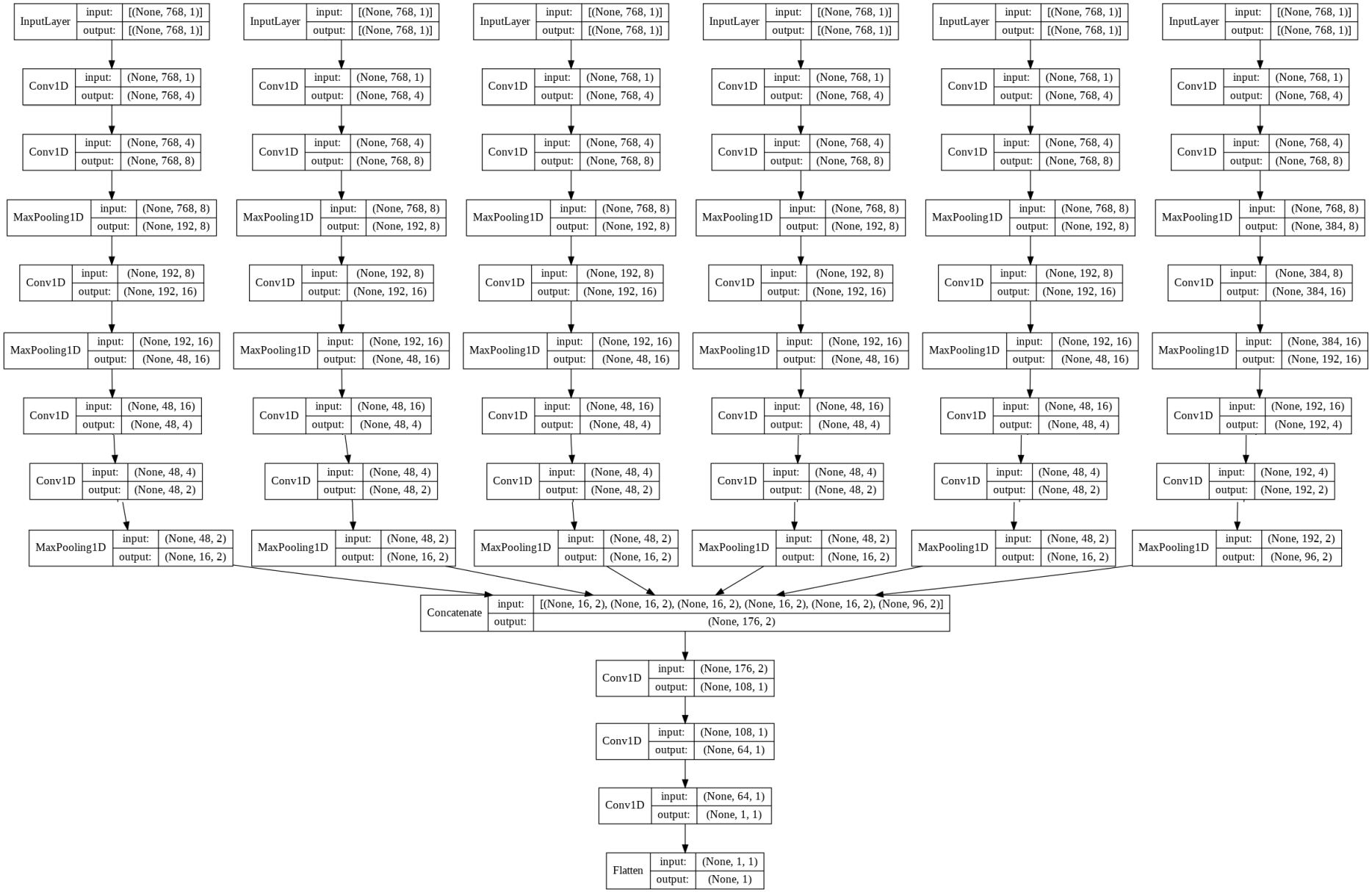


Figure 7: Model C

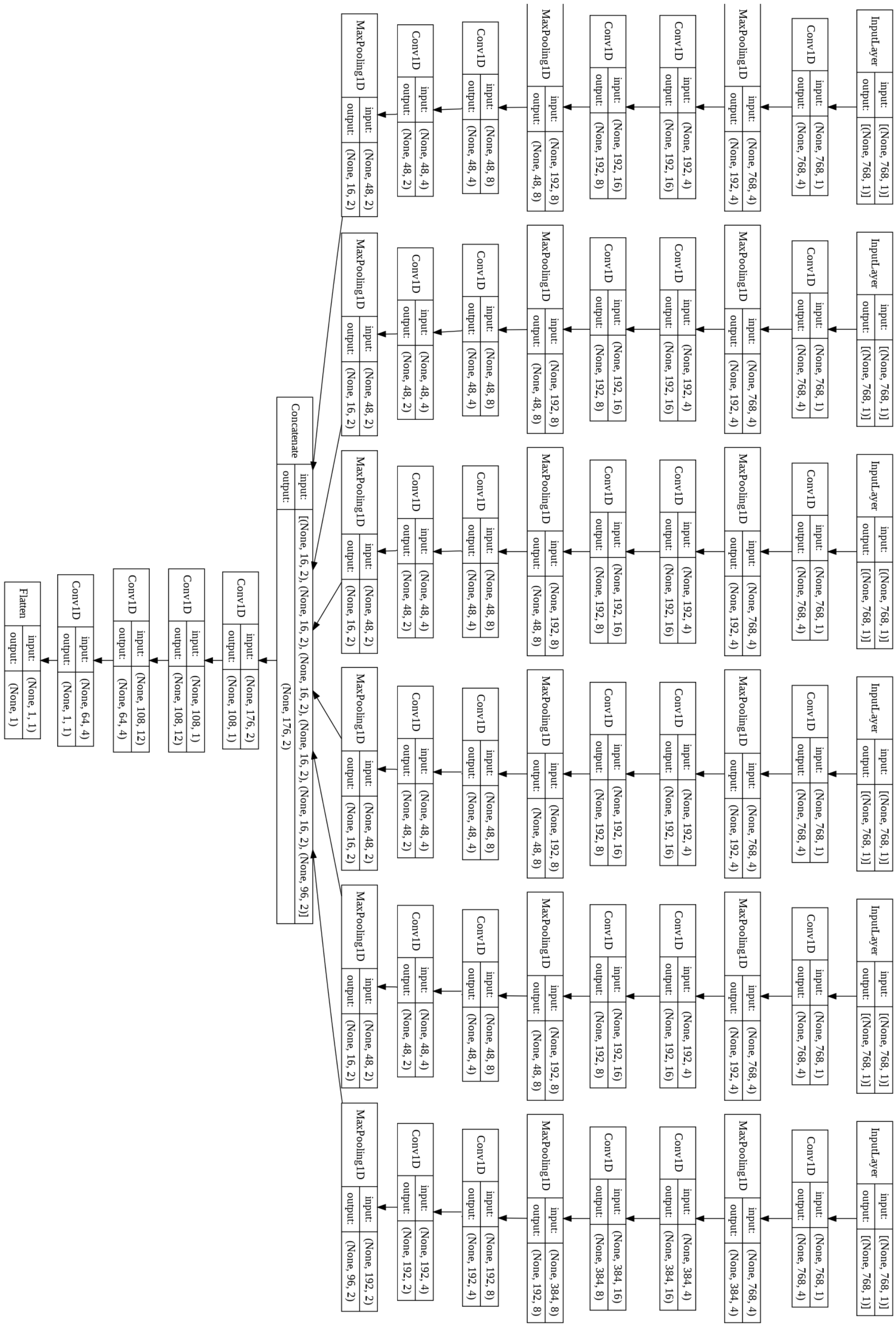


Figure 8: Model D

4 Dataset Details

The ColBERT Dataset is used for training and validating our model. This dataset was compiled, processed and published by the author of the ColBERT paper. It is available on GitHub[5] and Kaggle[4].

The dataset contains 200,000 sample texts. The dataset is balanced, i.e. there are exactly 100k humorous samples and 100k non-humorous samples. Figure 9 illustrates some statistics about the dataset.

GENERAL STATISTICS OF THE COLBERT DATASET (100K POSITIVE, 100K NEGATIVE)

	#chars	#words	#unique words	#punctuation	#duplicate words	#sentences	sentiment polarity	sentiment subjectivity
mean	71.561	12.811	12.371	2.378	0.440	1.180	0.051	0.317
std	12.305	2.307	2.134	1.941	0.794	0.448	0.288	0.327
min	36	10	3	0	0	1	-1.000	0.000
median	71	12	12	2	0	1	0.000	0.268
max	99	22	22	37	13	2	1.000	1.000

Figure 9: ColBERT Dataset Statistics

To use this data on our extension models, we reshaped the tokenized text data, then used these tokens to create embeddings using our BERT Head model. We then saved these embeddings to external loadable files, and used these to train our extension models.

Of these 200k samples, 160k samples are used for training, whereas the other 40k are used for validation/testing. K samples produce **six** embedding features of shape (K, 768).

Outputs or the labels are simply binary vectors, i.e. True/False or 1/0 of shape (160000, 1) for training labels or (40000, 1) for validation labels.

5 Experiments

For all experiments, Binary Cross Entropy loss was used. Additional metrics such as accuracy, precision and recall were calculated for most experiments. Adam optimizer with default optimizer hyper-parameters was used for all experiments. Batch size of 32 was used. All learning rates were chosen by trial and error. The learning rate which gave maximum validation accuracy over 5 epochs was chosen. Models consistently settled/plateaued at around 10 epochs. Keras callbacks were used to store the model with maximum validation accuracy while training.

- **Model A:** LR = 0.0004, Epochs = 50, 100s execution time per epoch (with GPU).
- **Model B:** LR = 0.001, Epochs = 15, 1100s execution time per epoch (without GPU)
- **Model C:** LR = 0.0005, Epochs = 30, 220s execution time per epoch (with GPU).
- **Model D:** LR = 0.0004, Epochs = 30, 190s execution time per epoch (with GPU).

Hardware Specifications: Tesla K80 GPU; Intel Xeon CPU 1 core @ 2.30GHz

6 Results

First, we compare the best validation accuracy models with the baseline models described in the original ColBERT paper, and the dense ColBERT model itself.

Evidently, the **worst** performing convolutional model 'A' outperforms the **best** baseline model, XLNet, by a small margin.

However, the **best** convolutional model 'D' is outclassed by the original ColBERT dense model by a margin of 0.026. Let us examine the probable cause for this. However, we *can* conclude from Figure 10 that the models using BERT embeddings outperform all baseline models, proving the superiority of BERT embeddings.

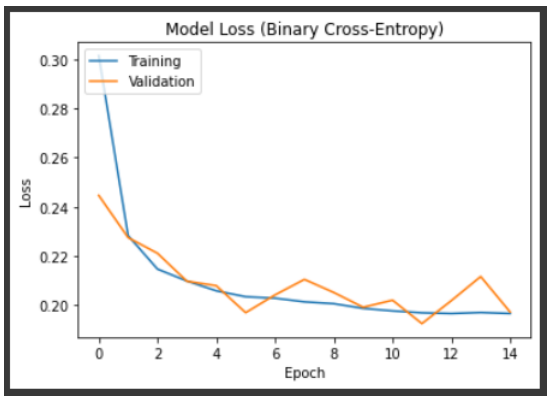
MODEL	CONFIGURATION	ACCURACY	PRECISION	RECALL	F1
Decision Tree		0.786	0.769	0.821	0.794
SVM	gamma = 1.0	0.872	0.869	0.88	0.874
Multinomial NB	alpha = 0.2	0.876	0.863	0.902	0.882
XGBoost		0.72	0.753	0.777	0.813
XLNet	Large-Cased	0.916	0.872	0.973	0.92
ColBERT	Dense	0.982	0.99	0.974	0.982
ColBERT	Model **A**	0.922	0.914	0.932	0.923
ColBERT	Model **B**	0.943	0.956	0.929	0.942
ColBERT	Model **C**	0.949	0.956	0.942	0.949
ColBERT	Model **D**	0.956	0.954	0.959	0.956

Figure 10: Baseline Models Comparative Statistics

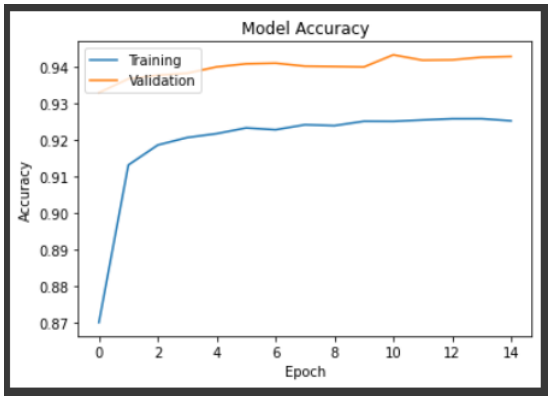
Now, we compare the complexity of the models. All the parameters stated below are the number of trainable parameters in the extension models. As we can see from Figure 11, the convolutional models all have at least a 25x reduction in the number of parameters, with only a maximum of 6%, and minimum of 2.6% reduction in accuracy.

MODEL	PARAMETERS	ACCURACY
ColBERT Dense	523,017	0.982
ColBERT Model A	524	0.922
ColBERT Model B	4041	0.943
ColBERT Model C	19630	0.949
ColBERT Model D	17220	0.956

Figure 11: ColBERT Models Comparative Statistics

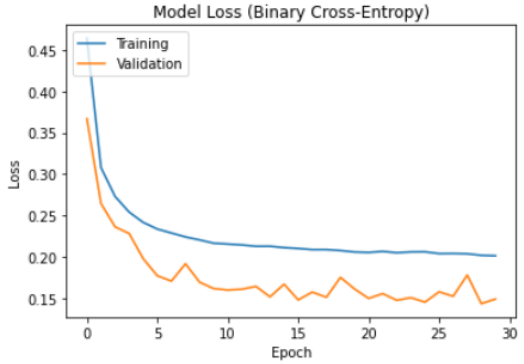


(a) Loss

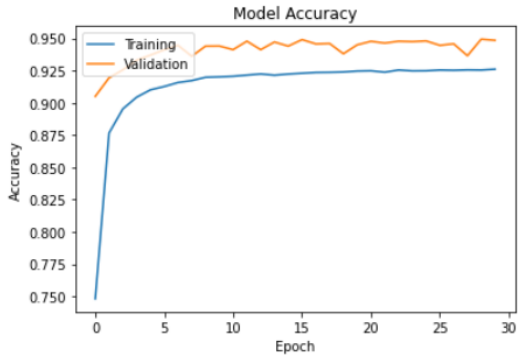


(b) Accuracy

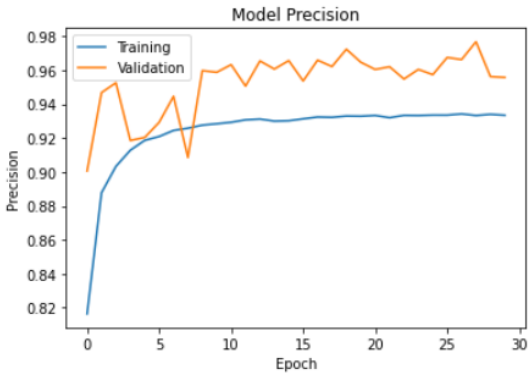
Figure 12: Model B Training History



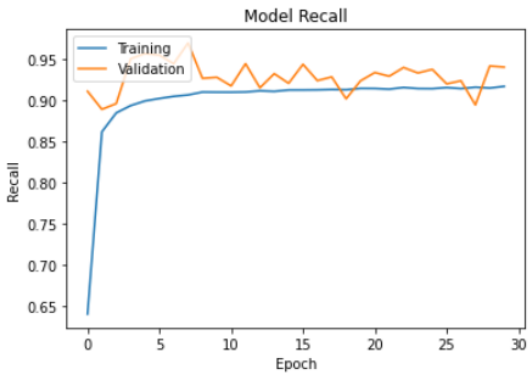
(a) Loss



(b) Accuracy

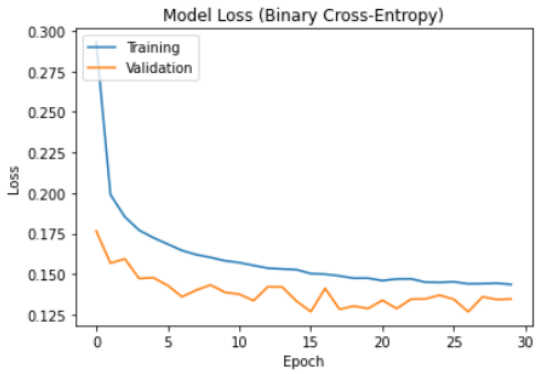


(c) Precision

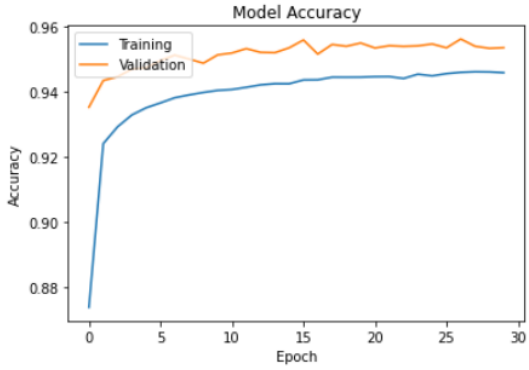


(d) Recall

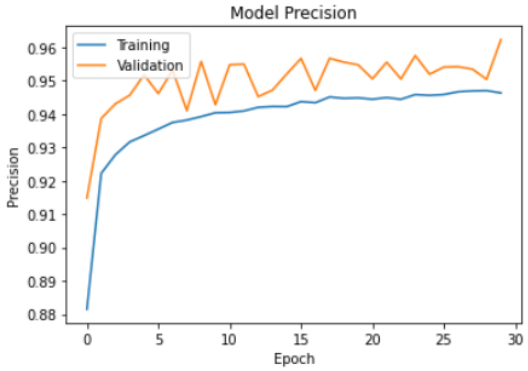
Figure 13: Model C Training History



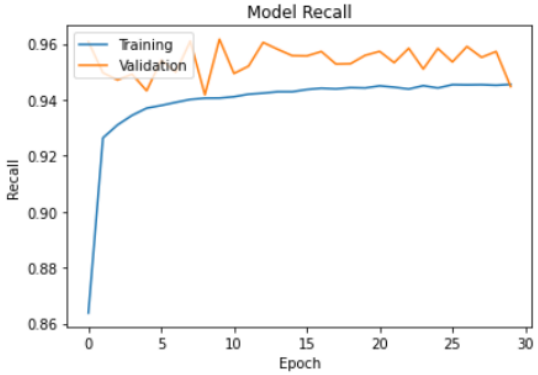
(a) Loss



(b) Accuracy



(c) Precision



(d) Recall

Figure 14: Model D Training History

7 Future Work

Some improvements which could be made to our work are:

- Can use AlBERT, RoBERTa, DistilBERT variants for faster embedding feature generation.
- Deeper and more complex convolutional extension models.
- Highway layers could be used in the convolutional models.
- Ensemble of multiple low trainable parameter models could be used for extremely fast training and predictions.

8 Conclusion

To conclude, we have performed an extensive study of the methodology and results of the original ColBERT paper, and replicated their results for humor classification in text.

We proposed the extension model heuristic to a BERT head, which greatly decreased training time. We have also proposed a convolutional extension model in place of the original dense extension model in ColBERT, which has led to an extreme reduction in the number of trainable parameters and training times.

We have obtained this extremely fast model at the cost of slightly less accuracy than the ColBERT model by approximately 2.6%. Our model has also outperformed all the baseline models described in the ColBERT model, which only proves that the combination of BERT embeddings, sentence-wise parallelization and sequential feature extraction by convolution performs amazingly.

We can also conclude from the relative performances of models C and D, that the more the contribution of the concatenated features (more convolutional layers after the concatenation), the more the model's potential to learn effectively.

As for improvements to our work in the future, the addition of highway layers seems like a very plausible option, implemented as described in the paper by Chen and Soo.

9 Contributions

- Pre-Midterm Review: Equal Contribution
- Initial Experiments and Exploratory Data Analysis: Om Prabhu
- Black Box Implementation: Gopalan Iyengar
- Implementation of Proposed Convolutional Models: Gopalan Iyengar
- Report: Gopalan Iyengar
- Presentation: Equal Contribution

References

[1] Issa Annamoradnejad. Colbert: Using BERT sentence embedding for humor detection. *CoRR*, abs/2004.12765, 2020.

[2] Peng-Yu Chen and Von-Wun Soo. Humor recognition using deep learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 113–117, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[4] Issa Annamoradnejad. 200k short texts for humor detection. https://www.kaggle.com/moradnejad/200k-short-texts-for-humor-detection, 2020.

[5] Issa Annamoradnejad. Colbert pretrained model and dataset. https://github.com/Moradnejad/ColBERT-Using-BERT-Sentence-Embedding-for-Humor-Detection, 2020.

[6] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[7] Orion Weller and Kevin D. Seppi. Humor detection: A transformer gets the last laugh. *CoRR*, abs/1909.00252, 2019.