Cornell Yoav Artzi NLP & ML Research Team Application

Method:
For the spring applications, I tried out a method described in a research paper by Prof. Thorsten Joachims, on using SVMs for text classification. That method used a basic tokenizer that did not utilize any of the advantages that advances in Deep Learning have made possible. For the applications this time, my aim was to use more advanced models to better my results overall.

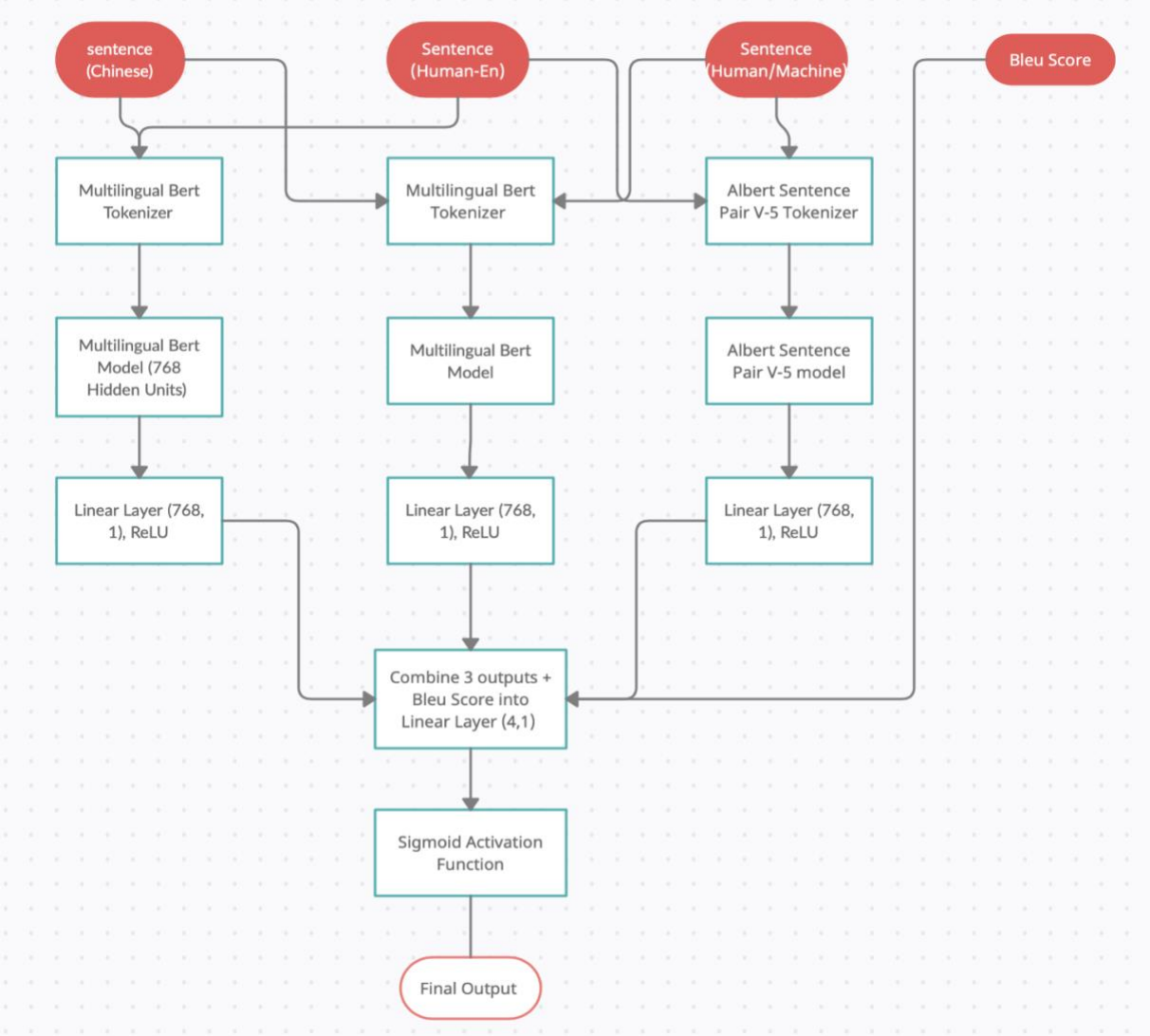
For this project, I have used the concept of Transfer Learning and finetuning to come up with a model that has an F1 score of 85% after optimizing the models for the highest validation accuracy.

I used two different Bert transformers: Multi-Lingual Bert and Albert v-5. Each of these models were used for sentence pair tokenization. Multi-Lingual Bert is used for sentence pair tokenization between multiple languages. Albert v-5 is for same language tokenization. I then feed the tokenizations forwards to their respective models. This is as shown in the model given below. Each of these models returns an output with 768 hidden units, and those are each put into 3 different linear layers that output a single value. We get three such values (using the ReLU activation) from each sentence pair combination and that combined with the bleu score is sent into a linear layer with 4 inputs and 1 output. This is sent into a sigmoid activation function, that we use to optimize a Binary Log loss function with the labels that were converted from (H, M) to (0,1).

As a note, each epoch of training took 45 minutes to an hour to train using sequential SGD minibatch with an Adam optimizer. Thus, I have only trained this model 6 times for a total of 30 hours of training. I started off with by keeping the Bert layer frozen and updating the linear models. However, I underfit a lot on the validation set and thus increased the number of epochs, but that still resulted in some underfitting. I decided to try unfreezing the Bert models and fine tuning its parameters. To prevent overfitting, I reduced the learning rate and the number of epochs. However, I still ended up overfitting on the validation set. Finally with a learning rate of 2×10^{-5} with a learning rate and a linear learning rate scheduler, for 4 epochs. I achieved an F1 score of 0.944, which would mean that either I overfit on the validation set, or the test set has a very different distribution to the training dataset (from which we drew the validation dataset).

Overall, I achieved an F1 score that was 0.1 higher than my model from spring which is a non-trivial improvement. In addition, I was limited by the time taken to train the model, which significantly limited my capabilities for training these models on my computer. With more hyperparameter tuning using something like Bayesian Optimization of hyperparameters, I could achieve better results.

Github link:
https://github.com/Sidv2001/NLP-LIL-Application.git


sentence (Chinese)
Sentence (Human-En)
Sentence (Human/Machine)
Bleu Score
Multilingual Bert Tokenizer
Multilingual Bert Tokenizer
Albert Sentence Pair V-5 Tokenizer
Multilingual Bert Model (768 Hidden Units)
Multilingual Bert Model
Albert Sentence Pair V-5 model
Linear Layer (768, 1), ReLU
Linear Layer (768, 1), ReLU
Linear Layer (768, 1), ReLU
Combine 3 outputs + Bleu Score into Linear Layer (4,1)
Sigmoid Activation Function
Final Output