Advanced Model Architecture:

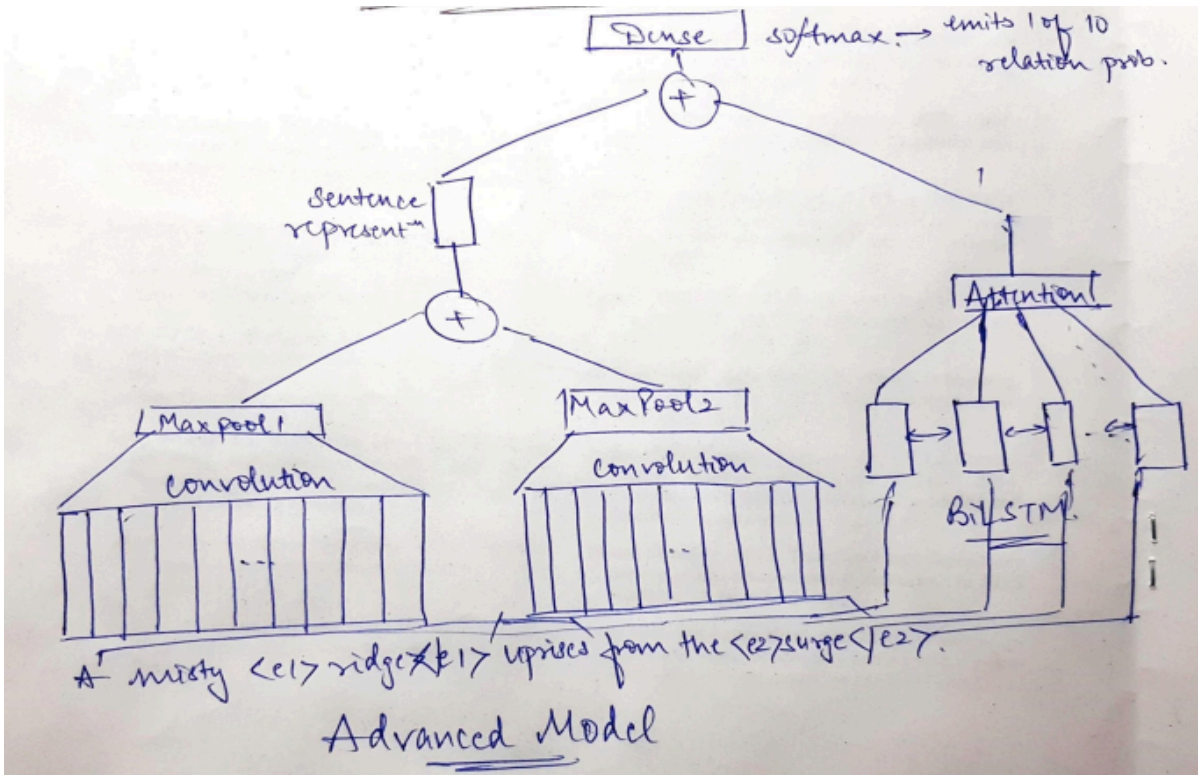

Fig: Advanced Model (Ensemble of CNN with Att-BiLSTM)

Implementation:
We have used 2 Convolution Layers in parallel, followed by 2 MaxPool on top of each. The result of the 2 pool layers is then concatted together to generate a 3D coarse representation of the original sentence. The result is flattened out and returned as an output for the custom ConvNet.

ConvNet Architecture:

Conv1 : layers.Conv1D(filters=512, kernel_size=2, input_shape=(None, 2 * embed_dim), activation='relu')
Conv2 : layers.Conv1D(filters=1024, kernel_size=3, inpu_shape=(None, 2 * embed_dim), activation='relu')

MaxPool Layers:

- **MaxPool1** : layers.MaxPool1D(pool_size=<sent_len-1>)
- **MaxPool2** : layers.MaxPool1D(pool_size=<sent_len-1>)

where sent_len = N-f+1 (output of Convoluted batch_sentence)

For the **Ensemble** part,I have used Bidirectional LSTM on the original word_embeddings , followed by attention(same as one employed in Basic Model,defined earlier).

The outputs from ConvNet and BiLSTM with attention is then concatted together and fed to the final classification Dense layer, to predict probabilities of Possible relations, derived from nominals in the input sentences from SemEval dataset.

BiLSTM Architecture:
LSTM Layer: layers.LSTM(6*self.hidden_size,return_sequences=True)

Bidirectional LSTM : layers.Bidirectional(self.lstm_layer, merge_mode='concat')