# Natural Language Processing

## Assignment 4

Viraj Kamat
112818603

## Bi-GRU Model

The Bidirectional Gated Recurrent Neural network model was implemented with the help of Keras bidirectional layer. The return_sequences parameter is set to True that captures the hidden_state of each cell in the GRU neural network at each time step.

The reason why we chose a GRU for this relation extraction problem mainly had to do with the vanishing gradient model, which failed to capture long distance dependencies in a sentence – this adversely affected the performance of the GRU model. We use a bidirectional approach since the hidden to hidden connections between each cell can utilize information from both past and future contexts – this is again essential for relation extraction[1] .

We then used an attention function on our model that took the output of the neural network and worked as follows :

1. Apply a tan function on the output
2. A trainable parameter w is multiplied with the output of step 1
3. The output of step 2 known as alpha is multiplied with the output of the RNN upon which a softmax is computed.
4. We then again compute a tan of the output of step 4.

We also compute a l2 regularization of the trainable parameters which as stated in the paper by Zhou et al helps prevent co-adaptation of hidden units by randomly omitting features of the hidden network during forward propagation.

Observation of different experiments on the BiGRU model :

1. Observation with Word-embeddings only : The F1 score for using word embeddings only when using our Neural model was a modest .46. While word embeddings do capture semantic information of the word involved and a n-dimensional space in which word-embeddings are drawn represents show word-vectors close to each other relate, the word vectors alone are not sufficient to capture dependency/relation between words in a sentence.

2. Observation with Word-embeddings and POS embeddings : The F1 score with word-embeddings and Part of speech tags scored a poor F1 score of .33 . While the presence of part of speech tags should have helped the neural model learn, providing additional embeddings in the form of Part-of-speech tags only skewed the learning process of the model. Meaningful information of the POS tags should help the model learn. The baseline model which included Part-of-speech tags along with the dependency structure adequately helped the neural model learn since it provided more info on what the POS tags meant regarding words in relation to one another.

3. Observation with Word-embeddings and Dependency structure : The F1 score of the neural model with the word-embeddings scored a healthy .58 . This makes sense since relation extraction aims to capture the relation between two words in a sentence and a dependency tree structure is built upon a sentence based on the relation of its words, thus the presence of the dependency tree tree structure vastly benefited the neural model in the learning process.

| Model Parameters | F1-Score | Loss at last epoch |
|---|---|---|
| Word embeddings | 0.4612 | 1.9288 |
| Word embeddings & POS-tags | 0.3388 | 2.0358 |
| Word embeddings & Dependency structure | 0.5829 | 1.7423 |
| Word embeddings, POS-tags & Dependency structure | 0.5997 | 1.8061 |

## Advanced Model – Convolutional Neural Network

### Justification :

The chosen method to implement an advanced model was a Convolutional Neural network which scored above the Bi-GRU with an F1-score of .64.

Convolutional neural networks have been used in the past for feature extraction i.e by referring to the word-embeddings, locally sensitive information of the words could be extracted and obtain high level features which could then be applied for relation extraction. Additional information is then included in the training process such as POS tags to remove the corpus richness limitation of the single-word vector training model.

Since sentence level attention is crucial in relation extraction and CNN's are purpose built for this task, it makes a lot more sense to use one, for eg. Take the following sentence

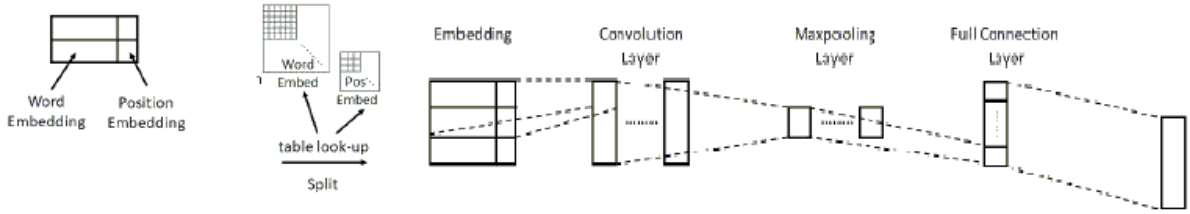
"Applebees is the best restaurant in Virgnia."

The word "best" plays an important role in the sentence which conveys the overall sentiment. An attention based neural model should then be able to identify those portions of a sentence that conveys important information and focus less on those components that contribute less. CNN's employ a technique where a matrix adequately scans over each set of words in a sentence to understand its meaning/sentiment or the task at hand.

Our model then works as follows :

1. We input a sentence which consists of word embeddings
2. Employs a convolutional neural model that learns local features from the different parts of the sentence
3. It finally concatenates the output such that into a single vector as the sentence global feature and uses a softmax layer to obtain the conditional probability matrix which describes the probability of different relation.
4. Our network thus consist of a convolution layer, a max-pooling layer and a fully connected layer as shown in the figure below

### Figure :



| Model Parameters | F1-Score | Loss at last epoch |
|---|---|---|
| dropout : 0.1<br>epochs : 5<br>num_filters : 32 | 0.5972 | 2.792 |
| dropout : 0.6<br>epochs : 10<br>num_filters : 32 | 0.5752 | 2.7824 |
| dropout : 0.6<br>epochs : 5<br>num_filters : 64 | 0.6405 | 2.7871 |

## References :


[1] Attention-based bidirectional long short-term memory networks for relation classification (Section 3.2)
[2] https://towardsdatascience.com/tensorflow-2-0-create-and-train-a-vanilla-cnn-on-google-colab-c7a0ac86d61b
[3] https://ieeexplore.ieee.org/abstract/document/8114660