Stony Brook University
Department Of Computer Science

**CSE 538 Natural Language Processing - Fall 2019**
Instructor: Niranjan Balasubramanian

# Assignment 4

Yatna Verma (**112673790**)

# GRU Implementation

## init()

Create a sequential model and add a GRU layer with a Bidirectional layer wrapper around it to the model.
Let the Bidirectional layer concat the forward and backward RNN result by keeping merge_mode to default.

## call()

Create a token_mask to mask out inputs which are zero.
Pass the word embeddings and token_mask to the model defined in init and get the BiGRU layer output.
Pass the output from the BiGRU layer to the atttention layer by calling the attn() function
Finally pass the output of attention layer to the final Dense layer before taking softmaxto get final output class.
We also add L2 regularization to prevent weights from assuming very high values and thus preventing overfitting.

## attn()

Create M by taking tanh of sentence embedding.
Reshape omega to [1, 1, 2*hidden_size] so that it can be multiplied (scalar) by M
Multiply M and reshaped_omega and sum along the last dimension (reduce_sum along axis 2)
To make the score of each word between 0 and 1 (alpha), take softmax along axis 1
Finally get the attention layer output by multiplying calculated attention(alpha) with the input.
Lastly, pass through a non-linearity (tanh) and return output.

# EXPERIMENTS

| **Experiment** | **Val F1 Score (after 5 epochs)** |
|---|---|
| Only Word Embed | 0.5744 |
| Word Embed + POS | 0.5127 |
| Word Embed + Dep Structure | **0.5787** |

## Experiment 1 (Using pre-trained glove embeddings)

Using random initialized word embeddings gave a F1 score of 0.5305, whereas using pre-trained glove embeddings gave an F1 score of .5744

This increase is because pre-trained word embeddings already carry some information about the meaning and usage of the word. This helps the model to learn a better representation of the sentence.

### Experiment 2 ( Word Embed + POS)

Using POS labels along with word embedding reduces the F1 score by ~0.06 to 0.5127
POS tags are useful for tasks like dependency parsing but here they did not turn out to be much useful. This is because to add the POS information for each word we had to double the embedding size (from 100 to 200). This means we added additional parameters for the model to learn which increases the computational cost as well as adds additional errors. POS tags could have been much more useful if it were a CNN model, however RNN models themselves are capable of capturing long range dependencies and POS tags did not add much useful information. Thus the useful information added by POS(if any) was overshadowed by the errors added and thus the F1 score reduced.

### Experiment 3 ( Word Embed + Dependency Structure)

Using dependency structure along with glove embeddings increases the F1 score to 57.87
Dependency structure holds the semantic meaning of a sentence. This semantic structure cannot be captured by any biRNN which only captures sequential information. Therefore adding dependency structure add semantic meaning to our sentence embeddings.

Also the biGRU model captures the relation between entities in both directions. However, the entity relation is present only in 1 direction. Shortest dependency path helps us recognize this direction.

# Advanced Model Implementation

## Model - Max Pooling with Multilayer BiRNN

### Justification -

**RNN over CNN** - I chose to stick with RNN rather than going with CNN because RNN-based model can deliver better performance on relation classification, and they are particularly capable of learning long-distance relation patterns.

**2 layer BiRNN** - I stacked 2 layers of BiRNN (tried both GRU and LSTM). The idea was to capture more advanced features as compared to a single layer.

**Max Pooling layer** - With the RNN structure, since the semantic meaning of a sentence is learned word by word, the segment-level feature vector produced at the end of the sentence in theory, represents the entire sentence. In practice, however the accumulation approach is not very suitable for relation learning because there are many long-distance patterns in the training data. Accumulation by recurrent connections tends to forget long-term information quickly, and supervision at the end of the sentence is hard to be propagated to early steps in model training, due to the annoying problem of gradient vanishing. Thus we resort to max-pooling. The argument is that the segment-level features, although not very strong in representing the entire sentence, can represent local patterns well. The semantic meaning of a sentence can be achieved by merging representations of the local patterns.

Max-pooling was chosen over mean-pooling as only several key words (trigger) and the associated patterns are important for relation classification, and so max-pooling is more appropriate to promote the most informative patterns.

**Diagram -**