INTRODUCTION TO NLP

Roll number - 2019121001

Assignment-2 Report

Implementation of the Model

The implementation of the current model requires three steps. They are as follows.

- Tokenization
- Preparation Of Dataset
- Model

Tokenization - By having a manual overview of the corpus, it was observed that the corpus was divided into different paragraphs, and each graph started with # and also ended with the same. So firstly, each paragraph was separated based on the regular expression to detect the heading. Then further, each paragraph was broken into sentences by splitting based on stop mark. With each Line, start and stop tag are appended to identify each sentence’s starting and ending point easily. Then further, the punctuations and non-alphanumeric characters, and other noises are removed and then tokenized.

Preparation Of Dataset - Since the neural networks can identify only the numerical features, each word in the whole dataset’s vocabulary is assigned a unique id. Each word(token) is converted to its numerical encoded form and then is passed further to create ngrams. After completing them, all the vectors are to be made of the same length, and hence smaller ngrams are padded with the ‘PAD’ tag encoded as 0. Due to limited RAM availability, the vocabulary cannot accommodate all words. Hence, the top 500 frequently occurring words are chosen as the vocabulary, and the rest are replaced with the <OOV> tag.

Model - The current model consists of the three-layer, Embedding layer, Bidirectional LSTM, and Dense Layer. The model uses Stochastic gradient descend for optimizing the problem. Categorical cross-entropy is chosen as the loss function.

Calculation of Perplexity Scores

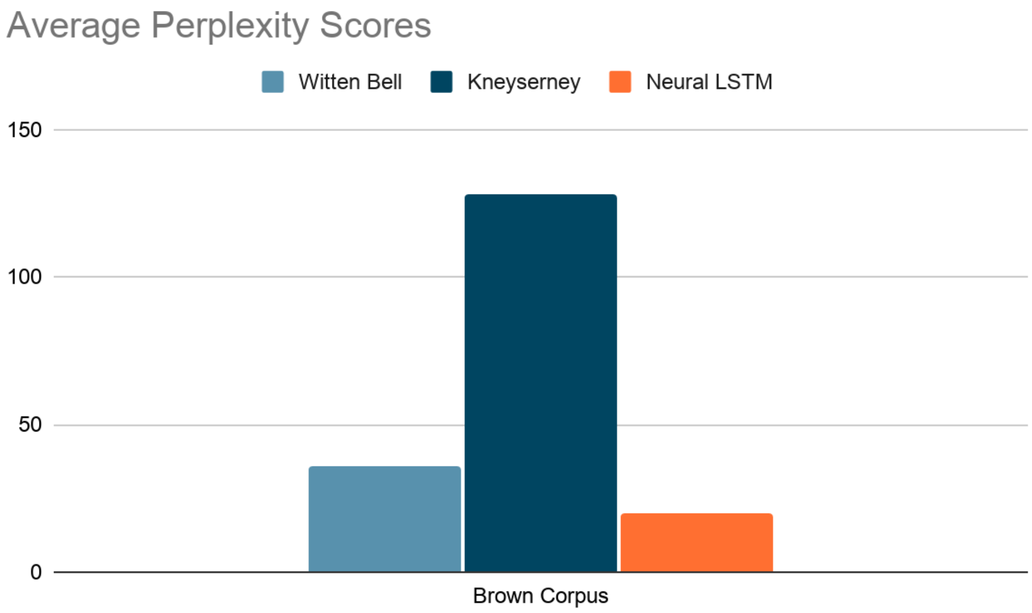
Since the output layer gives the probability of each word for the ngram given, the log of each token’s probability is added and then divided by the total number of tokens. Then, to get the perplexity score, the value is made the power of the exponentiation.

Evaluations and Results

Due to the large size of the train and validation data set, the model seems to fit only in 10 epochs. The accuracy observed on the training dataset was approximately 40% (39.7%), and the validation accuracy was nearly the same 40%. This implies that the model that learns from the data is able to replicate similarly on the unseen data.

The accuracy seemed relatively low. To test the model, firstly model was trained on ten sentences, which formed nearly 450 entries in training data, and approximately 90% training accuracy was observed, which implies that it is probably because of the heavy variation in data, which makes it unable to learn all the instances well.

Comparison with statistical models



From the above chart, we can see the neural LM outperforms all the two other statistical models. The neural lstms can handle larger data more accurately, if the data had been better, higher accuracy would have been achieved, and the average perplexity scores would have been relatively lower.