Intro to NLP

Devesh Marwah (2020115005)

February 2023

Pages: 3

Scores	Train-Pride	Test Pride	Train-Joyce	Test Joyce
Kneser	2.71938689890157	28.58409434429238	2.6006152656444623	33.78463180698573
Witten	2.764665042359865	5.775167828583368	3.75685919473356	8.137131351483555
Neural	247.42149048118213	305.74316668282313	74.86649334127229	124.45064419574636

1 Analysis

1.1 Language Models

Expectation: The test dataset should give higher perplexity than the train dataset. It is also observed in the given results. The train dataset for Joyce is much bigger and poetic as we go through it rather than prejudice, giving it a vast vocabulary and hence the values are a bit higher on that only. The idea of smoothing is to handle words which it might have never seen before, the same needs to be done. This is done by <UNK> token which is very low frequency value(here 1) clubbed together to act as if they are unknown. The idea is similar to good turing estimate and witten bell only builds on that.

Formula used:

$$p_{\text{KN}}(w_i|w_{i-n+1}^{i-1}) = \frac{\max\{c(w_{i-n+1}^{i}) - D, 0\}}{\sum_{w_i} c(w_{i-n+1}^{i})} + \frac{D}{\sum_{w_i} c(w_{i-n+1}^{i})} N_{1+}(w_{i-n+1}^{i-1}\bullet)\, p_{\text{KN}}(w_i|w_{i-n+2}^{i-1})$$

with the base case:

$$p_{\text{KN}}(w_i) = \frac{N_{1+}(\bullet w_i)}{N_{1+}(\bullet\bullet)}$$

where

$$N_{1+}(\bullet w_i) = |\{w_{i-1} : c(w_{i-1}w_i) > 0\}|$$

is the number of different words w_{i-1} that precede w_i in the training data and where

$$N_{1+}(\bullet\bullet) = \sum_{w_{i-1}} N_{1+}(w_{i-1}\bullet) = |\{(w_{i-1}, w_i) : c(w_{i-1}w_i) > 0\}| = \sum_{w_i} N_{1+}(\bullet w_i)$$

Figure 1: Kneser Ney

The perplexity hence on Joyce data in ngrams is expected to be higher and it shows in the values.

Witten-Bell Smoothing

Forumla used:

$$p_{\mathrm{WB}}(w_i|w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}}\, p_{\mathrm{ML}}(w_i|w_{i-n+1}^{i-1}) + (1-\lambda_{w_{i-n+1}^{i-1}})\, p_{\mathrm{WB}}(w_i|w_{i-n+2}^{i-1})$$

$$1-\lambda_{w_{i-n+1}^{i-1}} = \frac{N_{1+}(w_{i-n+1}^{i-1}\bullet)}{N_{1+}(w_{i-n+1}^{i-1}\bullet) + \sum_{w_i} c(w_{i-n+1}^{i})}$$

where:

$$N_{1+}(w_{i-n+1}^{i-1}\bullet) = |\{w_i : c(w_{i-n+1}^{i-1}w_i) > 0\}|$$

Figure 2: Witten bell

1.2 Neural model

The neural model is a bit tricky and has been made with keras. The dev split, however not shown has been used to fine tune the hyper parameters. It was mainly used to parameterise the 'reccurrent_dropout' parameter instead of regular dropout. Regular dropout is applied on the inputs and/or the outputs, meaning the vertical arrows from x_t and to h_t Recurrent dropout masks (or "drops") the connections between the recurrent units; that would be the horizontal arrows in the picture.

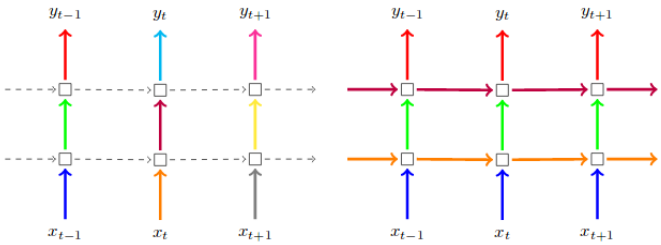

Figure 3: Recurrent Dropout

I started with the value of 0.3 and clipped it down to 0.2 but finally decided to set it on 0.5 for decent results.
The results of neural model are not so good because of the datafeeding. The data has been first converted to a vocabulary dictionary and assigned a number so that it can be feeded in it. It has been then specifically fixed to the upper length of 500 to fit it for training. The lower frequencies val has been handled by the <UNK> token and put in that to kinda account for unseen values. Other than that it has been One hot encoded by the 'np.eye' function to put it in the lstm. It has been then embedded using the keras layer only instead of glove embeddings. The intuition is that text especially in the Joyce textbook is quite different from normal language and would be better to use a local embedding layer only than Glove weights.

The maximum sentence length for pride and prejudice is 677 and for Joyce is 3133 in the corpus. Note that the joyce corpus has been modified for sentence length a bit and few punctuations were added to reduce the length. Even more than that, the vocablury in that is HUGE and hence PCA was applied to the vectors to reduce the layer. Then it was fit on ada only and can be run as instructed in the README.

2 Conclusion

The results are okayish and expected because a vanilla LSTM is used for the same and some better model architecture is needed for better results. The ngram model is currently performing better, but we know with better architectures, we can easily beat these basic models.