Intro to NLP

Hitesh Goel (2020115003)

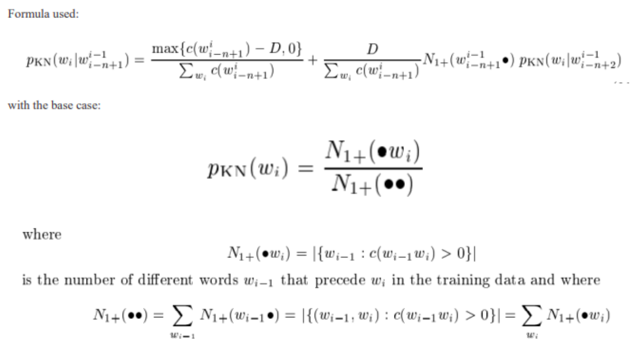
18 February 2023

Perplexity Scores	dataset1 Train	dataset1 Test	dataset2 Train	dataset2 Test
Kneser Ney Smoothing	5.641050735446315	6.673558064695263	3.039098875672282	4.167666472406597
Witten Bell Smoothing	2.601399662961914	6.147052730972677	3.9442618181769733	10.363967301629328
LSTM	247.42149048118213	305.74316668282313	74.86649334127229	124.45064419574636

1 Analysis

1.1 Language Models

Expectation: The test dataset should provide more confusion than the train dataset. That can also be seen in the results that were provided.
The values are a little bit higher on that alone since the train dataset for Joyce is much bigger and poetic as we read it rather than biassed, giving it a wide vocabulary. Smoothing is intended to handle terms that it may have never encountered before, thus the same needs to be done. This is achieved by grouping together very low frequency values (in this case 1, or "UNK" tokens), which pretend to be unknown. Witten Bell just expands on the good turing estimate-like concept.



Formula used:

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

with the base case:

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

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 Ulyssys 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, which was created with Keras, is somewhat challenging. Although not visible, the dev split was utilised to fine-tune the hyper settings. 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.

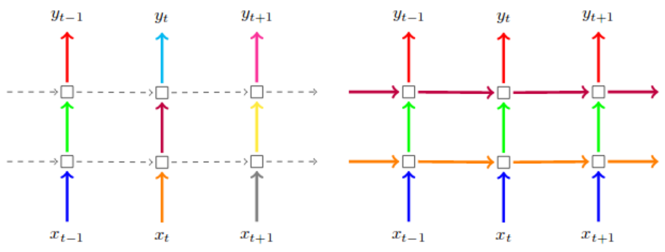


Figure 3: Recurrent Dropout

I began with the value of 0.3 and reduced it to 0.2 before deciding to set it at 0.5 to get respectable results. The datafeeding causes the neural model's outcomes to be less than ideal. The information was first transformed into a vocabulary dictionary and given a number so that it could be fed into it. Then, precisely, it was fastened to the upper length of 500 so that it would fit for training. The lower frequencies have been handled by the token and added there to sort of accommodate for values that are not observed. Other than that, the 'np.eye' function has One Hot encoded it before inserting it into the lstm. Then, in place of glove embeddings, it was embedded utilising solely the keras layer. The assumption is that the content, especially in the Joyce textbook, is quite distinct from everyday language and would be better if it were written differently. The intuition is that text especially in the Ulyssys textbook is quite different from normal language and would be better to use a local embedding layer only than Glove weights.

In the corpus, the maximum sentence length for Pride and Prejudice is 677, while it is 3133 for Joyce. The joyce corpus has been somewhat altered for sentence length, and a few punctuation marks have been inserted to shorten the length. In addition, the vocabulary in that is Enormous, so PCA was used to minimise the layer by applying it to the vectors. Once it was optimised for Ada, it could be executed as directed 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.