**Summer Semester 2020 SNLP Assignment 7**

**Name: Awantee Deshpande  
Id: 2581348  
Email:** [**s8awdesh@stud.uni-saarland.de**](mailto:s8awdesh@stud.uni-saarland.de)

**Name: Lakshmi Rajendra Bashyam  
Id: 2581455  
Email:** [**s8laraje@stud.uni-saarland.de**](mailto:s8awdesh@stud.uni-saarland.de)

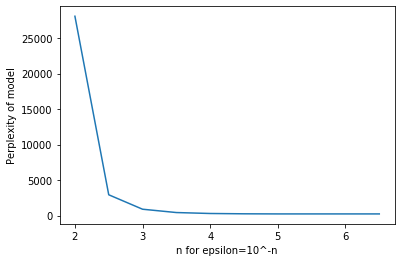
**Pruning Language Models**

1) Probability threshold pruning

The code for this can be found in SNLPAssg7\_1.py. We get the following results

|  |  |  |  |
| --- | --- | --- | --- |
| **Epsilon (n for 10-n)** | **#Unigrams** | **#Bigrams** | **Perplexity** |
| (no pruning) | 5765 | 45533 | 225.6136781 |
| 3 | 132 | 44465 | 887.5127072 |
| 4 | 1014 | 45479 | 285.62681502 |
| 5 | 5765 | 45533 | 225.6136781 |
| 6 | 5765 | 45533 | 225.6136781 |

Analysis: It is pretty intuitive that the more words we have in our model to train, the better the model is i.e. the lower the perplexity is. Hence, when we prune away a larger number of ngrams (corresponds to a small value of n) the model relies more and more on the back-off uniform probability values. We tried testing for n=0 and n=1 this makes 𝝐 = 1 and 0.1 resp. which prunes away all the ngrams leaving no model at all! Thereafter, as n increases, we prune away lesser and lesser number of ngrams, and the model perplexity decreases. Beyond n=5, we do not prune away a single n-gram and the perplexity is the same as that without pruning, which is ~225.   
(NOTE: At n=2 and 𝝐=0.01, the model has only 15 unigrams in it and has a huge perplexity of 28087.90931123331).



**Class-Based Language Models**

2) Incorporating POS-Tagging for Language Models

The code for this task can be found in SNLPAssg7\_2.py. We get the following results for a discounting factor of 0.9.

Perplexity of the POS Tag Class based model is 1010.6576210497838

Perplexity of Absolute Discounting based model is 1333.759847490673

Analysis: Some words are similar to other words in the vocabulary and can be grouped into classes (POS tags here). If we assign the words to such classes, it is reasonable to assume that unseen histories for words also belong to similar classes as the seen ones, and estimate the probabilities based on that. This provides more “information” to the model and reduces the uncertainty over the test set. Hence, we see that the perplexity is lower for the POS Tag based class model.  
(NOTE: To reduce bias, we shuffle the Penn Treebank data before splitting it into the train and test set. Therefore, every time the code is run, different perplexity values will be obtained for both models).