INLP Assignment-1 Report

Perplexity Scores

1. On “Pride and Prejudice” corpus:
    1. LM 1: Tokenization + 3-gram LM + Good-Turing Smoothin
        1. Train– 4.300661406461312
        2. Test - 2.0904711622577774
    2. LM 2: Tokenization + 3-gram LM + Linear Interpolation
        1. Train – 6.670651959858675
        2. Test - 127.93983394745626
2. On “Ulysses” corpus:
    1. i. LM 3: Tokenization + 3-gram LM + Good-Turing Smoothing
        1. Train - 5.021805077744325
        2. Test – 2.4248484817284806
    2. ii. LM 4: Tokenization + 3-gram LM + Linear Interpolation
        1. Train - 6.7261515713523155
        2. Test – 123.73480183199926

Observations

1. The perplexity of test for good Turing smoothing is lower than train as the probabilities assigned to the unseen n-grams is very high as we used the formula p0 = N1/N which assigned a very high probability to unseen trigrams. As the test sentences may not have been seen before while training the probability of test sentences comes out to be high which brings down the average perplexity perplexity .
2. The perplexities for train and test in case of linear interpolation smoothing look valid.

Generation examples:

test sentence – how are, k=10

predictions without smoothing:
('you', 0.21818371120541644), ('the', 0.10909185560270822), ('things', 0.10909185560270822), ('all', 0.05454592780135411), ('grub', 6.30511676014699e-16), ('charmed', 6.30511676014699e-16), ('bluest', 6.30511676014699e-16), ('neat', 6.30511676014699e-16), ('pretended', 6.30511676014699e-16), ('clarke', 6.30511676014699e-16)

predictions using linear interpolation:
('you', 0.2675603573839283), ('the', 0.13686307364337272), ('things', 0.11061056912577123), ('all', 0.060260753130291896), ('<EOS>', 0.02612358593414 65), ('<SOS>', 0.016924287635176036), ('not', 0.015103659886370156), ('a', 0.014549737136501142), ('in', 0.008010128424855014), ('they', 0.007785786516151883)

predictions using good turing :
{'the': 0.18717869841877266, 'you': 0.6009622802954314, 'all': 0.024680322867023187, 'things': 0.18717869841877266, 'arator': 0.9240811986989467, 'remerciez': 0.9240811986989467, 'extinction': 0.9240811986989467, 'links': 0.9240811986989467, 'waggons': 0.9240811986989467, 'crystallised': 0.9240811986989467}

Observations
1. with no smoothing there are more rare combinations of the words as I have assigned a very low probability for unseen trigrams if it does not exists
2. Linear interpolation gives words whose bigrams and unigrams that are frequently seen throughtout the corpus.
3. using good turing the words which occured together with the given context have appeared along with random unkown words.