CS60057

Language Modelling-Assignment I

-Arkajyoti Pal, 15CS30003.

- For calculating **perplexity**, the right-end padding </s> is also considered in the number of words of a sentence as suggested by *Jurafsky* et. al.

1. Language Modelling without Smoothing

1.1 Preprocessing

• Numerals and special characters were first removed from the first 40k sentences of the Brown corpus using regex matching(re library).

• Remaining words were then converted to lower case.

1.2 Verification of Zipf's Law

Zipf's Law: frequency of an n-gram $f \propto \frac{1}{r}$ where r is the position of the n-gram in a non-increasing sorted list of frequencies. To verify Zipf's Law on the n-gram models(with $n \in [1, 3]$), we plot $\log(f)$ vs. $\log(rank)$ which should ideally follow a straight line with negative slope.

1.2.1 Unigram Language Model

As Fig. 1.1 shows the small frequencies towards the tail-end seem to deviate from a Zipfian distribution which is further evident if we plot only 90% of the list as shown in Fig. 1.2.

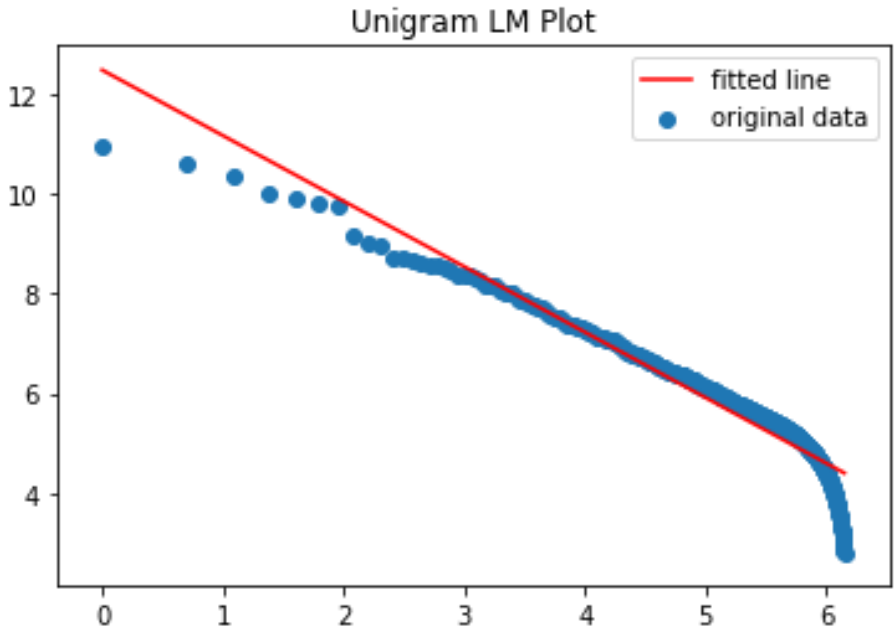

Fig. 1.1: Plot of log(f) vs log(r) for Unigram model. Slope of best-fit line: -1.32, Intercept: 12.49

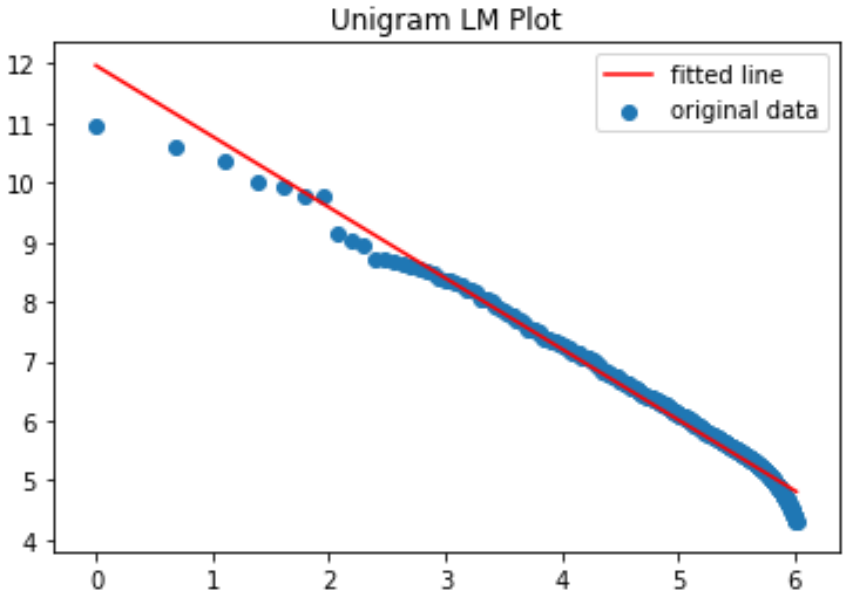

Fig. 1.2: Plot of log(f) vs log(r) for Unigram model. Slope of best-fit line: -1.19 ,Intercept: 11.97.

1.2.2 Bigram Language Model

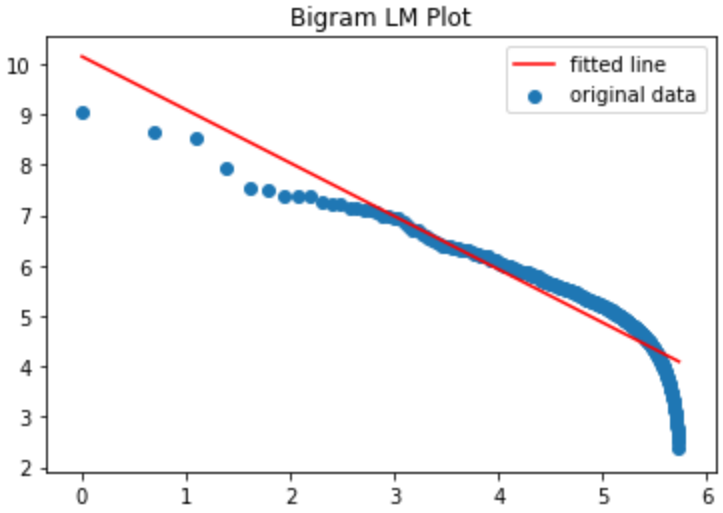

Fig. 2.1: Plot of log(f) vs log(r) for Bigram model. Slope of best-fit line: -1.05 , Intercept: 10.15.

As noted previously, the smaller frequencies towards the tail-end of the plot seem to deviated from a Zipfian distribution(Fig. 2.1) which is further evident by plotting the first 90% of the frequencies in Fig. 2.2.

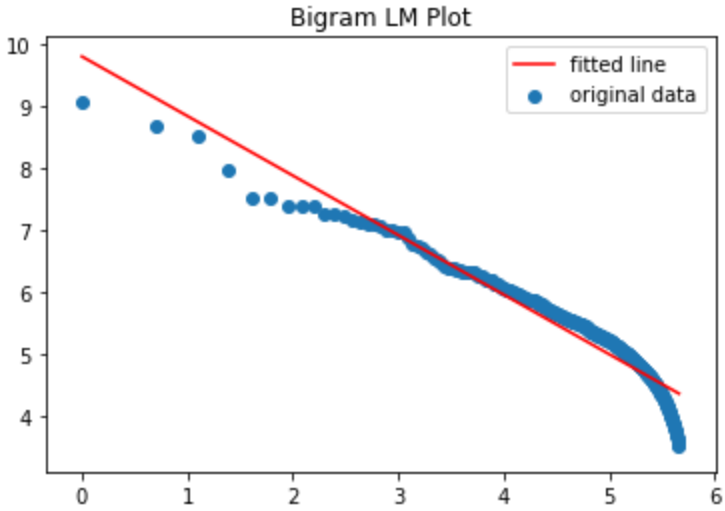

Fig. 2.2: Plot of log(f) vs log(r) for Bigram model. Slope of best-fit line: -0.96, Intercept: 9.79.

1.2.3 Trigram Language Model

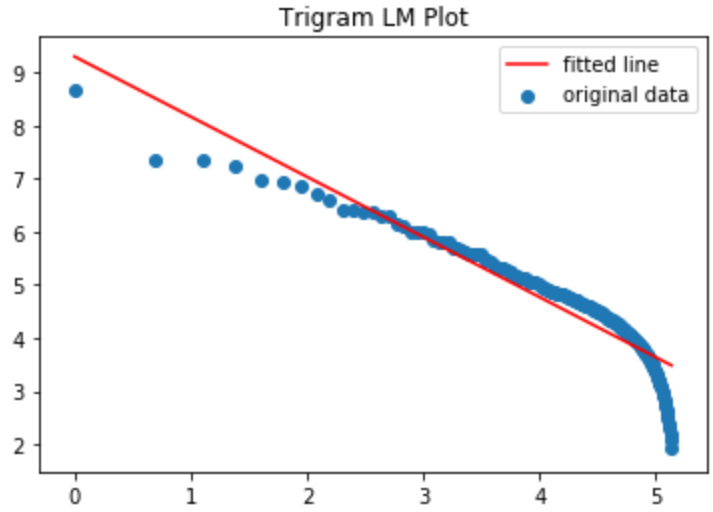

Fig. 3.1: Plot of log(f) vs log(r) for Trigram model. Slope of best-fit line: -1.13, Intercept: 9.3.

As noted previously, the smaller frequencies towards the tail-end of the plot seem to deviated from a Zipfian distribution(Fig. 3.1) which is further evident by plotting the first 90% of the frequencies in Fig. 3.2.

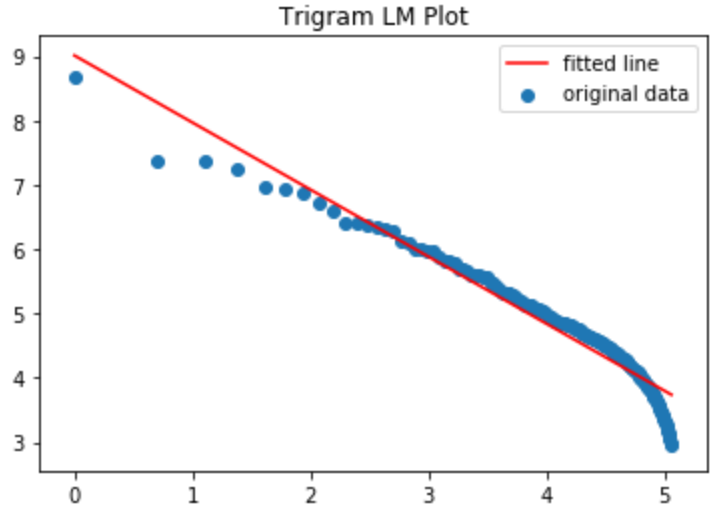

Fig. 3.2: Plot of log(f) vs log(r) for Trigram model. Slope of best-fit line: -1.04, Intercept: 9.01.

1.3 Top 10 n-grams(number in the tuple denotes count)

1.3.1 Top ten unigrams:

('the', 56474)
('of', 31329)
('and', 22154)
('to', 20413)
('a', 17918)
('in', 17766)
('is', 9474)
('that', 8306)
('for', 7797)
('it', 6198)
('was', 5967)

1.3.2 Top ten Bigrams

(('of', 'the'), 8512),
(('in', 'the'), 4986),
(('to', 'the'), 2820),
(('and', 'the'), 1848),
(('on', 'the'), 1829),
(('for', 'the'), 1592),
(('it', 'is'), 1390),
(('to', 'be'), 1377),
(('with', 'the'), 1261),
(('that', 'the'), 1243)

1.3.2 Top ten Trigrams

(('one', 'of', 'the'), 337),
(('the', 'united', 'states'), 336),
(('as', 'well', 'as'), 225),
(('some', 'of', 'the'), 156),
(('the', 'fact', 'that'), 154),
(('part', 'of', 'the'), 131),
(('of', 'the', 'united'), 127),
(('the', 'u', 's'), 126),
(('it', 'is', 'not'), 126),
(('a', 'number', 'of'), 118)

1.4 Log-likelihood and Perplexity scores without Smoothing

1.4.1 Unigram Language Model

The sequence "he lived a good life " has unigram log-likelihood of -32.62 and perplexity 229.54

The sequence "the man was happy " has unigram log-likelihood of -23.9 and perplexity 119.04

The sequence "the person was good " has unigram log-likelihood of -23.33 and perplexity 106.2

The sequence "the girl was sad " has unigram log-likelihood of -27.22 and perplexity 231.47

The sequence "he won the war " has unigram log-likelihood of -24.11 and perplexity 124.14

1.4.2 Bigram Language Model

The sequence "he lived a good life " has bigram log-likelihood of -26.89 and perplexity 88.33

The sequence "the man was happy " has bigram log-likelihood of -22.06 and perplexity 82.46

The sequence "the person was good " has bigram log-likelihood of -24.84 and perplexity 143.74

The sequence "the girl was sad " has bigram log-likelihood of -inf and perplexity inf

The sequence "he won the war" has bigram log-likelihood of -20.08 and perplexity 55.47

1.4.2 Trigram Language Model

The sequence "he lived a good life " has trigram log-likelihood of -inf and perplexity inf

The sequence "the man was happy " has trigram log-likelihood of -inf and perplexity inf

The sequence "the person was good " has trigram log-likelihood of -inf and perplexity inf

The sequence "the girl was sad " has trigram log-likelihood of -inf and perplexity inf

The sequence "he won the war" has trigram log-likelihood of -16.52 and perplexity 27.22

2. Language Modelling with Smoothing

$$P_{additive}(w_i|w_{i-n+1}..w_{i-1}) = \frac{k + count(w_{i-n+1}..w_i)}{k * |V| + count(w_{i-n+1}..w_{i-1})}$$

where |V| is equal to the number of unique unigrams in the corpus.

3.1 Laplacian/ Additive Smoothing

3.1.1 Unigram Model with Additive Smoothing

Unigram test log-likelihood with Laplacian/ Additive Smoothing **with k as 0.0001** :

The sequence "he lived a good life " has log-likelihood of -32.62 and perplexity 229.54
The sequence "the man was happy " has log-likelihood of -23.9 and perplexity 119.04
The sequence "the person was good " has log-likelihood of -23.33 and perplexity 106.2
The sequence "the girl was sad " has log-likelihood of -27.22 and perplexity 231.47
The sequence "he won the war" has log-likelihood of -24.11 and perplexity 124.14

Unigram test log-likelihood with Laplacian/ Additive Smoothing **with k as 0.001** :

The sequence "he lived a good life " has log-likelihood of -32.62 and perplexity 229.54
The sequence "the man was happy " has log-likelihood of -23.9 and perplexity 119.04
The sequence "the person was good " has log-likelihood of -23.33 and perplexity 106.21
The sequence "the girl was sad " has log-likelihood of -27.22 and perplexity 231.48
The sequence "he won the war" has log-likelihood of -24.11 and perplexity 124.14

Unigram test log-likelihood with Laplacian/ Additive Smoothing **with k as 0.01** :

The sequence "he lived a good life " has log-likelihood of -32.62 and perplexity 229.62
The sequence "the man was happy " has log-likelihood of -23.9 and perplexity 119.08
The sequence "the person was good " has log-likelihood of -23.33 and perplexity 106.24
The sequence "the girl was sad " has log-likelihood of -27.22 and perplexity 231.53
The sequence "he won the war" has log-likelihood of -24.11 and perplexity 124.18

Unigram test log-likelihood with Laplacian/ Additive Smoothing **with k as 0.1** :

The sequence "he lived a good life " has log-likelihood of -32.64 and perplexity 230.35
The sequence "the man was happy " has log-likelihood of -23.91 and perplexity 119.44
The sequence "the person was good " has log-likelihood of -23.34 and perplexity 106.58
The sequence "the girl was sad " has log-likelihood of -27.23 and perplexity 232.02
The sequence "he won the war" has log-likelihood of -24.12 and perplexity 124.57

Unigram test log-likelihood with Laplacian/ Additive Smoothing **with k as 1** :

The sequence "he lived a good life " has log-likelihood of -32.82 and perplexity 237.64
The sequence "the man was happy " has log-likelihood of -24.06 and perplexity 122.99
The sequence "the person was good " has log-likelihood of -23.5 and perplexity 109.94
The sequence "the girl was sad " has log-likelihood of -27.34 and perplexity 236.85
The sequence "he won the war" has log-likelihood of -24.28 and perplexity 128.42

3.1.2 Bigram Model with Additive Smoothing

Bigram test log-likelihood with Laplacian/ Additive Smoothing with **k as 0.0001 :**
The sequence "he lived a good life " has bigram log-likelihood of -26.96 and perplexity 89.36
The sequence "the man was happy " has bigram log-likelihood of -22.13 and perplexity 83.56
The sequence "the person was good " has bigram log-likelihood of -24.87 and perplexity 144.68
The sequence "the girl was sad " has bigram log-likelihood of -34.43 and perplexity 978.81
The sequence "he won the war" has bigram log-likelihood of -20.12 and perplexity 55.97

Bigram test log-likelihood with Laplacian/ Additive Smoothing with **k as 0.001 :**
The sequence "he lived a good life " has bigram log-likelihood of -27.47 and perplexity 97.38
The sequence "the man was happy " has bigram log-likelihood of -22.6 and perplexity 91.87
The sequence "the person was good " has bigram log-likelihood of -25.14 and perplexity 152.63
The sequence "the girl was sad " has bigram log-likelihood of -33.36 and perplexity 790.43
The sequence "he won the war" has bigram log-likelihood of -20.47 and perplexity 60.01

Bigram test log-likelihood with Laplacian/ Additive Smoothing with **k as 0.01 :**
The sequence "he lived a good life " has bigram log-likelihood of -29.84 and perplexity 144.48
The sequence "the man was happy " has bigram log-likelihood of -24.49 and perplexity 134.01
The sequence "the person was good " has bigram log-likelihood of -26.67 and perplexity 207.15
The sequence "the girl was sad " has bigram log-likelihood of -34.31 and perplexity 954.8
The sequence "he won the war" has bigram log-likelihood of -22.29 and perplexity 86.27

Bigram test log-likelihood with Laplacian/ Additive Smoothing with **k as 0.1 :**
The sequence "he lived a good life " has bigram log-likelihood of -35.5 and perplexity 371.12
The sequence "the man was happy " has bigram log-likelihood of -28.49 and perplexity 298.47
The sequence "the person was good " has bigram log-likelihood of -30.6 and perplexity 455.03
The sequence "the girl was sad " has bigram log-likelihood of -36.81 and perplexity 1575.99
The sequence "he won the war" has bigram log-likelihood of -26.59 and perplexity 203.78

Bigram test log-likelihood with Laplacian/ Additive Smoothing with **k as 1 :**
The sequence "he lived a good life " has bigram log-likelihood of -43.82 and perplexity 1485.99
The sequence "the man was happy " has bigram log-likelihood of -34.95 and perplexity 1084.75
The sequence "the person was good " has bigram log-likelihood of -36.66 and perplexity 1528.53
The sequence "the girl was sad " has bigram log-likelihood of -40.8 and perplexity 3497.22
The sequence "he won the war" has bigram log-likelihood of -33.43 and perplexity 801.45

3.1.3 Trigram Model with Additive Smoothing

Trigram test log-likelihoods with Laplacian/ Additive Smoothing with **k as 0.0001 :**
The sequence "he lived a good life " has log-likelihood of -38.41 and perplexity 602.76
The sequence "the man was happy " has log-likelihood of -25.16 and perplexity 153.19
The sequence "the person was good " has log-likelihood of -25.45 and perplexity 162.36
The sequence "the girl was sad " has log-likelihood of -26.66 and perplexity 207.05
The sequence "he won the war" has log-likelihood of -7.92 and perplexity 4.88

Trigram test log-likelihoods with Laplacian/ Additive Smoothing with **k as 0.001 :**
The sequence "he lived a good life " has log-likelihood of -39.9 and perplexity 772.27
The sequence "the man was happy " has log-likelihood of -27.39 and perplexity 239.14
The sequence "the person was good " has log-likelihood of -28.34 and perplexity 289.62
The sequence "the girl was sad " has log-likelihood of -31.56 and perplexity 551.35
The sequence "he won the war" has log-likelihood of -13.31 and perplexity 14.32

Trigram test log-likelihoods with Laplacian/ Additive Smoothing with **k as 0.01 :**
The sequence "he lived a good life " has log-likelihood of -44.29 and perplexity 1605.95
The sequence "the man was happy " has log-likelihood of -32.76 and perplexity 700.94
The sequence "the person was good " has log-likelihood of -34.16 and perplexity 926.99
The sequence "the girl was sad " has log-likelihood of -38.12 and perplexity 2047.03
The sequence "he won the war" has log-likelihood of -22.63 and perplexity 92.37

Trigram test log-likelihoods with Laplacian/ Additive Smoothing with **k as 0.1 :**
The sequence "he lived a good life " has log-likelihood of -51.52 and perplexity 5360.54
The sequence "the man was happy " has log-likelihood of -39.79 and perplexity 2859.31
The sequence "the person was good " has log-likelihood of -41.23 and perplexity 3809.45
The sequence "the girl was sad " has log-likelihood of -45.2 and perplexity 8432.04
The sequence "he won the war" has log-likelihood of -34.57 and perplexity 1005.91

Trigram test log-likelihoods with Laplacian/ Additive Smoothing with **k as 1 :**
The sequence "he lived a good life " has log-likelihood of -59.51 and perplexity 20302.7
The sequence "the man was happy " has log-likelihood of -47.83 and perplexity 14282.46
The sequence "the person was good " has log-likelihood of -48.74 and perplexity 17132.82
The sequence "the girl was sad " has log-likelihood of -52.05 and perplexity 33193.74
The sequence "he won the war" has log-likelihood of -46.42 and perplexity 10758.13

3.2 Good Turing Smoothing

Katz(1987) showed that Good Turing for large counts is not reliable . Based on his work, smoothing in practice is not applied to large r's. Hence a threshold of k=5 as recommended by him is proposed here such that:

r* = r for r > k.

The probability mass is then normalized proportionately.

3.2.1 Bigram Language Model with Good Turing

The sequence "he lived a good life " has bigram log-likelihood of **-20.25** and perplexity 29.22
The sequence "the man was happy " has bigram log-likelihood of **-15.42** and perplexity 21.86
The sequence "the person was good " has bigram log-likelihood of **-19.3** and perplexity 47.45
The sequence "the girl was sad " has bigram log-likelihood of **-27.58** and perplexity 248.87
The sequence "he won the war" has bigram log-likelihood of **-13.07** and perplexity 13.66

3.2.2 Trigram Language Model with Good Turing

Trigram test log-likelihood:
The sequence "he lived a good life " has trigram log-likelihood of -51.85 and perplexity 5665.0
The sequence "the man was happy " has trigram log-likelihood of -32.73 and perplexity 697.1
The sequence "the person was good " has trigram log-likelihood of -34.15 and perplexity 925.72
The sequence "the girl was sad " has trigram log-likelihood of -inf and perplexity inf
The sequence "he won the war" has trigram log-likelihood of -2.65 and perplexity 1.7

3.3.3 Why Good Turing Smoothing can not be applied to Unigram Model

In Bigram model, we take $n_0 = |V| * |V| - n_{bigrams\ that\ occured\ in\ the\ corpus}$ to account for bigrams that never occured.

And we do similarly for trigram model. However, in case of unigram model we don't know the size of the vocabulary |V| and hence we can't account for words that never appeared in the corpus and hence Good Turing smoothing can't be applied to unigram model.

3.3 Interpolation Method

Log-likelihood and Perplexity scores:
Bigram test log-likelihood with Interpolation with **lambda as 0.2 :**
The sequence "he lived a good life " has log-likelihood of -21.22 and perplexity 34.36
The sequence "the man was happy " has log-likelihood of -16.1 and perplexity 25.02
The sequence "the person was good " has log-likelihood of -16.22 and perplexity 25.63
The sequence "the girl was sad " has log-likelihood of -19.52 and perplexity 49.59
The sequence "he won the war" has log-likelihood of -15.81 and perplexity 23.6

Bigram test log likelihoods with Interpolation with **lambda as 0.5 :**
The sequence "he lived a good life " has log-likelihood of -21.71 and perplexity 37.26
The sequence "the man was happy " has log-likelihood of -16.67 and perplexity 28.03
The sequence "the person was good " has log-likelihood of -16.99 and perplexity 29.89
The sequence "the girl was sad " has log-likelihood of -20.27 and perplexity 57.59
The sequence "he won the war" has log-likelihood of -16.15 and perplexity 25.29

Bigram test log likelihoods with Interpolation with **lambda as 0.8 :**
The sequence "he lived a good life " has log-likelihood of -23.47 and perplexity 49.97
The sequence "the man was happy " has log-likelihood of -18.64 and perplexity 41.61
The sequence "the person was good " has log-likelihood of -19.49 and perplexity 49.34
The sequence "the girl was sad " has log-likelihood of -22.73 and perplexity 94.25
The sequence "he won the war" has log-likelihood of -17.65 and perplexity 34.15

Conclusion

The importance of smoothing can be seen from the fact that the log-likelihood scores of models without smoothing many a times returned a log-likelihood of -infinity suggesting a particular n-gram was not seen during training. Also, a rich, varied corpus can increase the chances of seeing a diverse set of n-grams reducing the chances of seeing unseen n-grams during testing.

Read Me

The ipython notebook has also been uploaded to ensure a better readability of the code as it is diveded into sections with markdowns.