INLP Assignment-1
Report

Jai Bhatnagar (2021111029)

February 2024

1 Perplexity Scores

LM1

Smoothing : Good-Turing
Corpus : Pride and Prejudice

Average Test set perplexity = 8761.66
Average Train set perplexity = 9063.86

LM2

Smoothing : Linear Interpolation
Corpus : Pride and Prejudice

Average Test set perplexity = 577.72
Average Train set perplexity = 11.40

LM3

Smoothing : Good-Turing
Corpus : Ulysses

Average Test set perplexity = 6447.80
Average Train set perplexity = 9798.02

LM4

Smoothing : Linear Interpolation
Corpus : Ulysses

Average Test set perplexity = 1385.09
Average Train set perplexity = 11.61

As we can see, in linear interpolation models the train set perplexity is very good and test set perplexity is higher because of unseen n-grams and words.

However here good-turing models perform worse than linear interpolation models since they assign a high probability to unseen n-grams. And since a high probability is assigned to unseen n-grams it may perform better on test set than on train set.

2 Generation (Given starting token '$< s >$')

No Smoothing model on "Pride and Prejudice" corpus

Text Generated (upto 100 tokens):

$< s >$ That will do extremely well , unless you comply with the cold and ceremonious politeness of her being yet more agitated voice , " he replied , " said he , " - - you never dropt a word about fishing , as to command a full refund of the room . $< /s >$ $< s >$ Younge was of great houses ; after honestly telling her what he wanted to know every thing else she is saving enough . $< /s >$ $< s >$ Bennet could have foreseen . $< /s >$ $< s >$ Darcy should have no reason , for the sake of finding her otherwise than

No Smoothing model on "Ulysses" corpus

Text Generated (upto 100 tokens):

$< s >$ Martin Cunningham added . $< /s >$ $< s >$ Full fathom five thy father lies . $< /s >$ $< s >$ He let go of the south . $< /s >$ $< s >$ And among the warm dark stairs and called them and find it in the lute alone sat : Goulding , told Mr Bloom could easily have picked up the flabby gush of porter that was her age and beef to the not particularly redolent sea on the occasion (¡NUM¿ / - Ill tell him I loved rousing that dog in the north city diningrooms in Marlborough street from Miss Kate Morkan in the Buckingham Palace

LM2

Text Generated (upto 100 tokens):

$< s >$ and Mrs seek , during were A deeper shade . $< /s >$ $< s >$ Elizabeth ' s condescension the to open the whole interesting to Elizabeth will because distribution good none of them from it . $< /s >$ $< s >$, extremely Ignorant - so advantageous to them , capable of _some_ , considering his civil enquiries after Mr $< s >$. him and herself with the him civility , which he was to _she_ the , your resentment than surprise . $< /s >$ $< s >$ On entering her de Bourgh Gutenberg Literary very well - proportioned room wishing , at all those things . Mrs was

LM4

Text Generated (upto 100 tokens):

$< s >$ black cheers that them all . . $< /s >$ $< s >$) _ KITTY city told Freeman ! $< /s >$, exponent of the slicked his days with settedst little by the light , scavengers etc . $< /s >$ $< s >$ soul different $< s >$ Yet , Stephen said , now at the this chaffering How (Dringdring ! $< /s >$ $< s >$ BLOOM : m afraid , father whose hair Rows of seriously sure $< /s >$ $< s >$ Wait $< s >$ final old the . $< /s >$ Wheatenmeal with honey is another _pot_ to a place aforesaid , the constable . $< /s >$ smugglers boarhound . $< /s >$ $< s >$ I know overjoyed as

3 Theory

Good-Turing Smoothing

We assume a frequency distribution according to Zipf's law i.e. most words will have low frequency, least words will have most frequency, and the plot between frequencies (r) and number of words having those frequencies (N_r) will be almost linear in log-scale.

First we change the N_r values to preserve this trend even at higher values of r or lower values of N_r.

$$N_r = \frac{2N_r}{(t-q)}$$

for all non-zero N_r, where t is the next higher frequency at which N_r is non-zero and q is the previous lower frequency at which N_r is non-zero.

Then we train a linear regressor to smooth out the N_r values

$$log(N_r) = a + b\, log(r)$$

Finally we estimate the adjusted frequencies as,

$$r^* = (r+1)\frac{S(N_{r+1})}{S(N_r)}$$

where r^* is the new adjusted frequency, $S(N_r)$ is the predicted value of N_r from the linear regression model. $(S(N_0) = 1)$

After this, these adjusted frequencies are used everywhere for probability calculations.

Linear Interpolation

In linear interpolation we use the probabilities of n-grams, (n-1)-grams ... unigrams. Each probability is assigned a weight which are learned using the following algorithm (for trigrams).

```
set λ1 = λ2 = λ3 = 0
foreach trigram t1,t2,t3 with f(t1,t2,t3) > 0
    depending on the maximum of the following three values:
        case (f(t1,t2,t3)−1)/(f(t1,t2)−1):  increment λ3 by f(t1,t2,t3)
        case (f(t2,t3)−1)/(f(t2)−1):  increment λ2 by f(t1,t2,t3)
        case (f(t3)−1)/(N−1):  increment λ1 by f(t1,t2,t3)
    end
end
normalize λ1,λ2,λ3
```

Then probability is calculated as (for trigrams),

$$P_L(w3|w1w2) = \lambda_1 P(w3|w1w2) + \lambda_2 P(w3|w2) + \lambda_3 P(w3)$$