Assignment 03 - Exploring SRILM Toolkit

Anirudh Kalla

October 19, 2020

1 Question 01

Bigram Perplexity (01) : 2.16914
Bigram Perplexity (02) : 2.07033
Unigram Perplexity (01) : 5.58150
Unigram Perplexity (02) : 5.56193

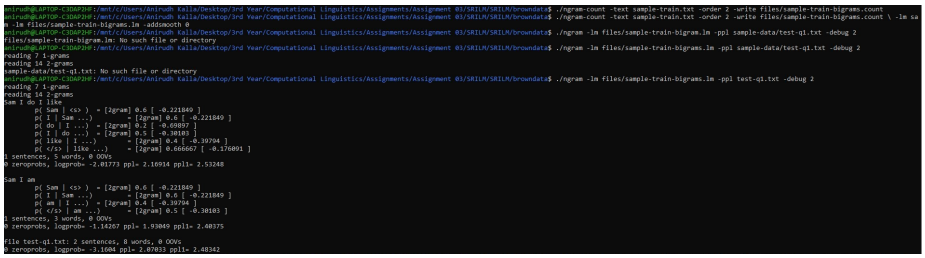

Figure 1: Bigram/Unigram Perplexity

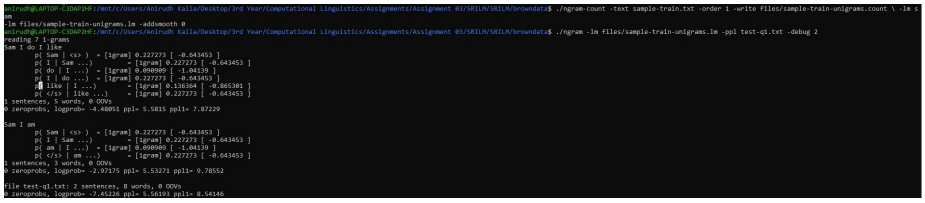

Figure 2: Bigram/Unigram Perplexity

2 Question 2.1

Bigram Perplexity (Q-5): 2.61475

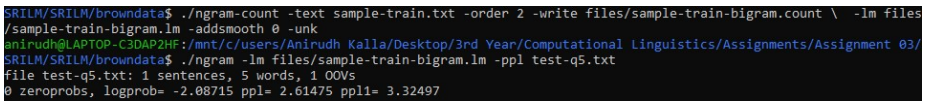

Figure 3: Bigram Perplexity Q-5

3 Question 2.2

Unigram Perplexity (Q-5): 5.85343

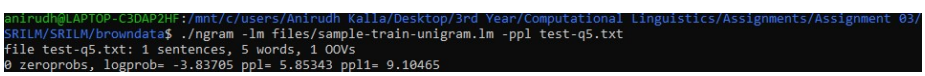

Figure 4: Unigram Perplexity Q-5

4 Question 3.1

Smoothed Bigram Perplexity (Q-5): 5.8464

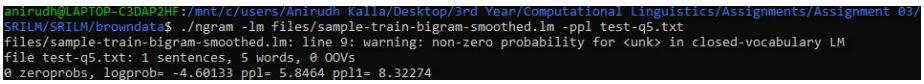

Figure 5: Bigram Perplexity Smoothed

5 Question 3.1

Smoothed Unigram Perplexity (Q-5): 7.82454

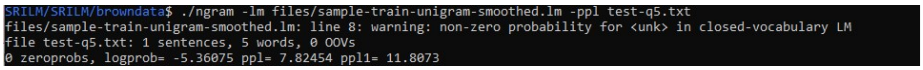

Figure 6: Unigram Perplexity Smoothed

6 Question 04

1. The Log Probability is infinite because of it's undefined nature.
   $10^x \neq 0 \forall x \in \{-\infty, \infty\}$

2. -unk tag helps identify unknown (OOV) words. This, in turn, helps in assigning them some probability mass through the most appropriate smoothing technique (as determined by SRILM Toolkit)

3. The SRILM Toolkit applies smoothing by default, therefore to explicitly prevent smoothing, we call -addsmooth 0

4. We can apply smoothing techniques to distribute probability mass from higher probability brackets to low/zero probability brackets

7 Question 05

Good-Turing Smoothing was applied to the bigram model and the following result was obtained:-
Perplexity Score: 3.16342

8 Question 07

Trigram Model Details:-

1. Perplexity: 5.67142

```
SRILM/SRILM/browndata$ ./ngram-count -text sample-train.txt -order 2 -write files/sample-train-bigram-gt.count \  -lm fi
les/sample-train-bigram-gt.lm -gt2min -unk
Warning: option "-gt2min" got non-floating-point argument "-unk".  Using default: 1.
warning: no singleton counts
GT discounting disabled
warning: count of count 8 is zero -- lowering maxcount
warning: count of count 7 is zero -- lowering maxcount
warning: count of count 6 is zero -- lowering maxcount
warning: count of count 5 is zero -- lowering maxcount
warning: count of count 4 is zero -- lowering maxcount
warning: discount coeff 2 is out of range: 0
anirudh@LAPTOP-C3DAP2HF:/mnt/c/users/Anirudh Kalla/Desktop/3rd Year/Computational Linguistics/Assignments/Assignment 03/
SRILM/SRILM/browndata$ ./ngram -lm files/sample-train-bigram-gt.lm -ppl test-q5.txt -debug 2
reading 7 1-grams
reading 14 2-grams
Sam I do like linguistics
        p( Sam | <s> )  = [2gram] 0.5 [ -0.30103 ]
        p( I | Sam ...)         = [2gram] 0.5 [ -0.30103 ]
        p( do | I ...)  = [2gram] 0.166667 [ -0.778151 ]
        p( like | do ...)       = [2gram] 0.333333 [ -0.477121 ]
        p( <unk> | like ...)    = [OOV] 0 [ -inf ]
        p( </s> | <unk> ...)    = [1gram] 0.227273 [ -0.643453 ]
1 sentences, 5 words, 1 OOVs
0 zeroprobs, logprob= -2.50079 ppl= 3.16342 ppl1= 4.21887

file test-q5.txt: 1 sentences, 5 words, 1 OOVs
0 zeroprobs, logprob= -2.50079 ppl= 3.16342 ppl1= 4.21887
```

Figure 7: GT Smoothing

2. Log Probability: -4.52215

```
anirudh@LAPTOP-C3DAP2HF:/mnt/c/users/Anirudh Kalla/Desktop/3rd Year/Computational Linguistics/Assignments/Assignment 03/
SRILM/SRILM/browndata$ ./ngram -lm files/sample-train-trigram.lm -ppl test-q5.txt -debug 2
reading 8 1-grams
files/sample-train-trigram.lm: line 10: warning: non-zero probability for <unk> in closed-vocabulary LM
reading 14 2-grams
reading 1 3-grams
Sam I do like linguistics
        p( Sam | <s> )  = [2gram] 0.333333 [ -0.477121 ]
        p( I | Sam ...)         = [3gram] 0.4 [ -0.39794 ]
        p( do | I ...)  = [2gram] 0.166667 [ -0.778151 ]
        p( like | do ...)       = [2gram] 0.222222 [ -0.653212 ]
        p( <unk> | like ...)    = [1gram] 0.0294118 [ -1.53148 ]
        p( </s> | <unk> ...)    = [1gram] 0.206897 [ -0.684247 ]
1 sentences, 5 words, 0 OOVs
0 zeroprobs, logprob= -4.52215 ppl= 5.67142 ppl1= 8.02472

file test-q5.txt: 1 sentences, 5 words, 0 OOVs
0 zeroprobs, logprob= -4.52215 ppl= 5.67142 ppl1= 8.02472
```

Figure 8: Trigram Model Details

9 Question 08

1. If we have a task of text classification or sentiment analysis then we should remove stop words as they do not provide any information to our model (due to their relative abundance). But if we have the task of language translation then stopwords are useful, as they have to be translated along with other words.

2. Removing punctuations is a good practice since space separated puntuations may often be treated as separate words. Therefore to prevent error in Perplexity and Log-Prob calculations.

10 Question 09 (On Brown Corpus)

Optimum Values of :-

1. $\lambda = 0.3$
2. $\lambda_1 = 0.7$
3. Smoothing: Good-Turing 2 for Bigrams and Good-Turing 3 for Trigrams

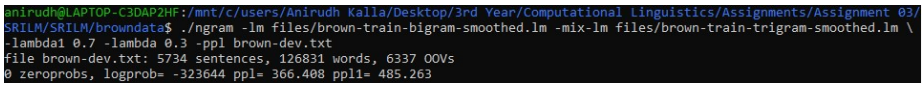

Figure 9: Dev Data

Figure 10: Test Data