NLP-Assignment1

Sravya Kurra
sxk180049

Q1.1
I used dictionaries for all the 3 counters and initialized to empty values. Set can be used for vocabulary also but I chose dictionary. We need dictionary because we need the count of each n-gram and context.

Q1.2
The formula for word probability has count of the context in denominator. If we have an unseen context, its count will be zero and word probability has zero in denominator and the word will have infinite probability. But in reality, if we have an unseen context, each word should have equal probability. So we will use 1/V, giving equal chance for every word in the vocabulary to occur

Q1.3
sentence Probability for statement1 is: -28.982533492033653. The second statement end's up in math domain error because we have the all the contexts but we don't have the n-gram making the numerator of word_prob zero. Hence word_prob is zero, taking a log which results in math domain error.

Q2.1
we weren't able to fix the error for second statement. Though we replace the unseen words with <unk>, n gram count is still zero.

Q2.2
prob = (float)(w_ngram + delta) / (w_con + delta * len(self.context_counts.keys()))
it works fine without any modifications

Q2.3
Delta=0.1
sentence prob for statement 1: -122.77918495348746
sentence prob for statement 2: -72.45359836527285

Delta=0.5
sentence prob for statement 1: -140.86755644928084
sentence prob for statement 2: -77.09974592526252

Delta=0.9
sentence prob for statement 1: -146.6438531931505
sentence prob for statement 2: -79.06683542032542

It can be seen that with increased values of delta, the sentence probability of 1st statement also increases, indicating that Laplace smoothing increases probability for the seen words as well. This is not desired. It is supposed to shift some probability from seen words to unseen to maintain a balance. From the above values, it is evident that that's not the case. So Laplace doesn't work great for n-gram language models

Sravya Kurra
sxk180049

Q2.4

With out Smoothening=>delta=0

NGramLM
sentence prob for statement 1: -36.35470829179692
second statement ends in error

NGramInterpolator

Interpolation probability for statement 1 -50.50925041952093
Interpolation probability for statement 2 -55.255979407880154

With smoothening
Delta=0.1
NGramLM
sentence prob for statement 1: -122.77918495348746
sentence prob for statement 2: -72.45359836527285
NGramInterpolator
Interpolation probability for statement 1 -128.65510696592608
Interpolation probability for statement 2 -69.30453919801988

Q3.1 Shakespeare

Delta=0
Math Domain Error
Delta=0.5
total_text_prob -11.943091275758256
PP is 153751.24385629082

We get math domain error when delta is zero because of unseen n-grams either due to words or contexts

Q3.2
Delta=0.5
Shakespeare
total_text_prob -11.943091275758256
PP is 153751.24385629082

Warpeace
total_text_prob -11.714722793527901
PP is 122359.99942038821

Lower PP is better, making the model trained on warpeace better with masking the Out-of-Vocabulary word. Usually we have to get lower probability for Shakespeare. But we mask the words that occurred once in Shakespeare and in sonnets and also perform smoothening resulting in increased perplexity than that of warpeace
Warpeace **without masking the test data**
total_text_prob -12.66076592618665
PP is 315138.0028603456

Warpeace **after masking the test data**

vocab count before mask 41566
vocab count after masking: 19406
total_text_prob -11.714722793527901
PP is 122359.99942038821

Shakespeare **without masking the test data**

total_text_prob -12.544009794264799
PP is 280410.475973963

Shakespeare **after masking the test data**

vocab count before mask 62983
vocab count after masking: 29125
total_text_prob -11.943091275758256
PP is 153751.24385629082

Q3.3
Perplexity works fine to an extent to identify the authorship of the text when the test data isn't masked(replace with <unk>)
It's mostly likely that an author uses similar words in similar contexts which he/she has encountered earlier and the author is more likely to use few words than other.
We don't need to bother much about out of vocabulary words in the test data as that can be taken care by smoothening. It's a simplest way one can use

Lexical features considering word length, sentence length and how frequent the word has been used helps to identify the author

NLP-Assignment1 Sravya Kurra
sxk180049

Q4.1

As to the beginning. </s>
John. Ah, poor heart! </s>
What is this? </s>
It shall be well used: exclaim no more voice </s>
Caesar, I bring consuming sorrow to my clamours! let us

They doesn’t make much sense. They are not meaningful and they don’t keep up the context and I observed that we have end tokens in them

Q4.2

Bigram: And I am a <unk> </s>
Trigram: And I will not be <unk> </s>
4-gram: And I will take thee too. </s>
5-gram: And I will take up that with 'Give the devil

These sentence starts of almost similar and they end with words than end of line token
These sentences ate little meaningful compared to random generation of text

Q5.1
It took a little more than a week. I faced major problem because there is an ambiguous statement in question 2 on word_prob.

Q5.2
I discussed with few of my classmates including Aishwarya, Manas and a group on groupme