Assignment 02 - Language Modelling

Anirudh Kalla

October 14, 2020

1 Question 01

Vocabulary=100236

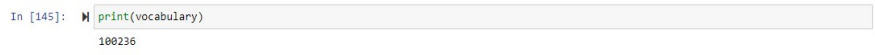

Figure 1: Vocabulary

2 Question 02

Most Probable Unigrams:

1. jury
2. of

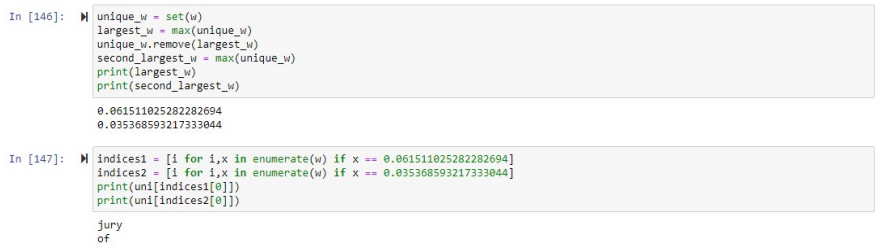

Figure 2: Most Probable Unigrams

3 Question 03

The Most Frequent Bigram:-
('of', 'the')

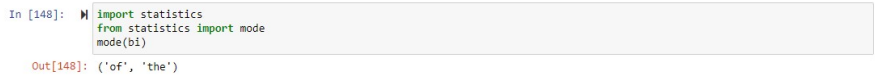

Figure 3: Most Probable Bigrams

4 Question 04

Probability of an unseen Bigram :-
$9.976455564866915X10^{-6}$

```
In [149]:
unseen_bigram_prob=1/vocabulary
print(unseen_bigram_prob)

9.976455564866915e-06
```

Figure 4: P(Unseen Bigram)

5 Question 06.a

Probability of given sentence using the Vanilla MLE Bigram Model is:-
0.0 This is because none of these bigrams occur in the brown-corpus.

Therefore according to Maximum Likelihood Estimation, this sentence has NULL probability.

```
In [150]:
lst_test = []
unigrams_test = {}
bigrams_test = {}
uni_test=[]
bi_test=[]
text_test="<s> peter piper picked a pickled pepper </s>"
lst_test = text_test.strip().split()
print('Read ', len(lst_test), ' tokens')
del text_test
for l in lst_test:
    if not l in unigrams_test:
        unigrams_test[l] = 1
    else:
        unigrams_test[l] += 1
        uni_test.append(l)
vocabulary=len(unigrams_test)
print('Generated ', len(unigrams_test), ' unigrams')

for i in range(len(lst_test) - 1):
    temp = (lst_test[i], lst_test[i+1])
    if not temp in bigrams_test:
        bigrams_test[temp] = 1
    else:
        bigrams_test[temp] += 1
        bi_test.append(temp)
print(unigrams_test)
test_prob=1
print('Generated ', len(bigrams_test), ' bigrams')
print(bigrams_test)
u_test=list(unigrams.keys())
b_test=list(bigrams.keys())
for i in range(len(b_test)):
    if b_test[i] in bigrams.keys():
        test_prob=test_prob*bi_prob
print(test_prob)

Read  8  tokens
Generated  8  unigrams
{'<s>': 1, 'peter': 1, 'piper': 1, 'picked': 1, 'a': 1, 'pickled': 1, 'pepper': 1, '</s>': 1}
Generated  7  bigrams
{('<s>', 'peter'): 1, ('peter', 'piper'): 1, ('piper', 'picked'): 1, ('picked', 'a'): 1, ('a', 'pickled'): 1, ('pickled', 'p
epper'): 1, ('pepper', '</s>'): 1}
0.0
```

Figure 5: P(Given Sentence: Vanilla MLE)

6 Question 06.b

Probability of given sentence using Laplace Smoothing Bigram Model is:-
0.00000000000000000000000000000000001795733085778058

This is because, even though none of these bigrams occur in the brown-corpus each of these UNK bigrams are assigned a (smoothed) probability. Giving the P(Given Sentence) a non-zero value.

```
text_test="<s> peter piper picked a pickled pepper </s>"
lst_test = text_test.strip().split()
print('Read ', len(lst_test), ' tokens')
del text_test
for l in lst_test:
    if not l in unigrams_test:
        unigrams_test[l] = 1
    else:
        unigrams_test[l] += 1
        uni_test.append(l)
vocabulary=len(unigrams_test)
print('Generated ', len(unigrams_test), ' unigrams')

for i in range(len(lst_test) - 1):
    temp = (lst_test[i], lst_test[i+1])
    if not temp in bigrams_test:
        bigrams_test[temp] = 1
    else:
        bigrams_test[temp] += 1
        bi_test.append(temp)
print(unigrams_test)
test_prob=1
print('Generated ', len(bigrams_test), ' bigrams')
print(bigrams_test)
u_test=list(unigrams.keys())
b_test=list(bigrams_test.keys())
for i in range(len(b_test)):
    if b_test[i] in bigrams.keys():
        test_prob=test_prob*bi_prob_laplace
    else:
        test_prob=test_prob*unseen_bigram_prob
        print('N')
print("{:.50f}".format(test_prob))
```

```
Read  8  tokens
Generated  8  unigrams
{'<s>': 1, 'peter': 1, 'piper': 1, 'picked': 1, 'a': 1, 'pickled': 1, 'pepper': 1, '</s>': 1}
Generated  7  bigrams
{('<s>', 'peter'): 1, ('peter', 'piper'): 1, ('piper', 'picked'): 1, ('picked', 'a'): 1, ('a', 'pickled'): 1, ('pickled', 'p
epper'): 1, ('pepper', '</s>'): 1}
N
N
N
N
N
N
0.00000000000000000000000000000000001795733085778058
```

Figure 6: P(Given Sentence: Laplace smoothed)

7 Question 07

MLE table of Bigrams (Vanilla)

```
w=[]
x=[]
y=[]
data=[]
data1=[]
data2=[]
data3=[]
data4=[]
for k,v in bigrams.items():
    first_word = k[0]
    first_word_count = unigrams[first_word]
    bi_prob = bigrams[k] / unigrams[first_word]
    bi_prob_laplace= (bigrams[k]+1) / (unigrams[first_word]+vocabulary)
    x.append(bi_prob)
    y.append(bi_prob_laplace)
    uni_prob = unigrams[first_word] / total_corpus
    w.append(uni_prob)
    final_prob = bi_prob + uni_prob
    #print(k[0] +'\t'+ k[1] +'\t'+'\t' + str(v) + '\t'+ '\t'+ str(first_word_count) + '\t'+ '\t'+'\t'+ str(bi_prob))
    data1.append(k[0]+' '+k[1])
    data2.append(str(v))
    data3.append(str(first_word_count))
    data4.append(str(bi_prob))
data = list(zip(data1, data2, data3, data4))
df = pd.DataFrame(data, columns = ['Bigrams', 'Bigrams Count', 'Unigram Count', 'Bigram Probability'])
df.head()
```

```
Read  1014940  tokens
Generated  100236  unigrams
Generated  539927  bigrams
```

	Bigrams	Bigrams Count	Unigram Count	Bigram Probability
0	The Fulton	1	6452	0.00015499070055796651
1	Fulton County	5	16	0.3125
2	County Grand	1	58	0.017241379310344827
3	Grand Jury	2	17	0.11764705882352941
4	Jury said	1	2	0.5

Figure 7: Bigram MLE Table

8 Question 08

MLE table of Bigrams (Laplace Smoothed)

```
w=[]
x=[]
y=[]
data=[]
data1=[]
data2=[]
data3=[]
data4=[]
for k,v in bigrams.items():
    first_word = k[0]
    first_word_count = unigrams[first_word]
    bi_prob = bigrams[k] / unigrams[first_word]
    bi_prob_laplace= (bigrams[k]+1) / (unigrams[first_word]+vocabulary)
    x.append(bi_prob)
    y.append(bi_prob_laplace)
    uni_prob = unigrams[first_word] / total_corpus
    w.append(uni_prob)
    final_prob = bi_prob + uni_prob
    #print(k[0] +'\t'+ k[1] +'\t'+'\t' + str(v) + '\t'+ '\t'+ str(first_word_count) + '\t'+ '\t'+'\t'+ str(bi_prob))
    data1.append(k[0]+' '+k[1])
    data2.append(str(v))
    data3.append(str(first_word_count))
    data4.append(str(bi_prob))
data = list(zip(data1, data2, data3, data4))
df = pd.DataFrame(data, columns = ['Bigrams', 'Bigrams Count', 'Unigram Count', 'Bigram Probability'])
df.head()
```

```
Read  1014940  tokens
Generated  100236  unigrams
Generated  539927  bigrams
```

	Bigrams	Bigrams Count	Unigram Count	Bigram Probability
0	The Fulton	1	6452	0.00015499070055796651
1	Fulton County	5	16	0.3125
2	County Grand	1	58	0.017241379310344827
3	Grand Jury	2	17	0.11764705882352941
4	Jury said	1	2	0.5

Figure 8: Bigram MLE (Laplace Smoothed) Table