Author of Solutions: Manzoor Ali
Email: manzoorali29@gmail.com

Natural Language Processing
Chapter 03:
N-GRAM LANGUAGE MODELS

—

Book: Speech and language processing Book

(Authors of book: Daniel Jurafsky and James H.Martin**)**

---

1. Write out the equation for trigram probability estimation (modifying Eq. 3.11). Now write out all the non-zero trigram probabilities for the I am Sam corpus on page 33.

   References:

   Eq. 3.11:

$$P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

   Page 33 Corpus:

```
<s> I am Sam </s>
<s> Sam I am </s>
<s> I do not like green eggs and ham </s>
```

Solution:

a. The equation for trigram probability is:

$$P(w_n|w_{n-2}w_{n-1}) = \frac{C(w_{n-2}w_{n-1}w_n)}{C(w_{n-2}w_{n-1})}$$

b. Non-zero trigram probabilities:

P(am|<s> I) = ½ = 0.5

P(Sam| I am) = ½ = 0.5

P(</s>| am Sam) = 1/1 = 1

P(I | <S> Sam) = 1/1 = 1

….. , and so forth.

2. Calculate the probability of the sentence "i want chinese food". Give two probabilities, one using Fig. 3.2 and the 'useful probabilities' just below it on page 35, and another using the add-1 smoothed table in Fig. 3.6. Assume the additional add-1 smoothed probabilities P(i|<s>) = 0.19 and P(</s>|food) =0.40.

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Figure 3.2 Bigram probabilities for eight words in the Berkeley Restaurant Project corpus of 9332 sentences. Zero probabilities are in gray.

Solution:

P(<s> I want chinese food </s>) = P(I|<s>) P(want|I) P(chinese|want) P(food|chinese) P(</s>|food) = 0.19 * 0.33 * 0.0065 * 0.52 * 0.40 = 8.47704×10^{-5}

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

Figure 3.6 Add-one smoothed bigram probabilities for eight of the words (out of $V = 1446$) in the BeRP corpus of 9332 sentences. Previously-zero probabilities are in gray.

P(<s> I want chinese food </s>) = P(I|<s>) P(want|I) P(chinese|want) P(food|chinese) P(</s>|food) = 0.19 * 0.21 * 0.0029 * 0.052 * 0.40 = 2.406768×10^{-6}

3. Which of the two probabilities you computed in the previous exercise is higher, unsmoothed or smoothed? Explain why.

Solution:

Without smoothing, bigrams have high probabilities; therefore, the unsmoothed solution in exercise 3 has higher probability than the smoothed.

4. We are given the following corpus, modified from the one in the chapter:

```
<s> I am Sam </s>
<s> Sam I am </s>
<s> I am Sam </s>
<s> I do not like green eggs and ham </s>
```

Using a bigram language model with add-one smoothing, what is P(Sam |am)? Include <s> and </s> in your counts just like any other token.

Solution:

Step 1: Create the vocabulary:

['i', 'am', 'sam', '</s>', '<s>', 'do', 'not', 'like', 'green', 'eggs', 'and']

Step 2: find the length 'V' of the vocabulary:

V = 11

Step 3: create a count matrix for all bigrams

	i	am	sam	</s>	<s>	do	not	like	green	eggs	and
(<s>,)	3.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
(i,)	0.0	3.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
(am,)	0.0	0.0	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
(sam,)	1.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
(</s>,)	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0
(do,)	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
(not,)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
(like,)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
(green,)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
(eggs,)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
(and,)	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Step 4: Calculate the sum of each row:

	i	am	sam	</s>	<s>	do	not	like	green	eggs	and	
(<s>,)	3.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0
(i,)	0.0	3.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	4.0
(am,)	0.0	0.0	2.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
(sam,)	1.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0
(</s>,)	0.0	0.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
(do,)	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
(not,)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0
(like,)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0
(green,)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0
(eggs,)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
(and,)	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Step 5: Add one to all the element:

	i	am	sam	</s>	<s>	do	not	like	green	eggs	and
(<s>,)	4.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
(i,)	1.0	4.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0
(am,)	1.0	1.0	3.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
(sam,)	2.0	1.0	1.0	4.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
(</s>,)	1.0	1.0	1.0	1.0	4.0	1.0	1.0	1.0	1.0	1.0	1.0
(do,)	1.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0	1.0
(not,)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0
(like,)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0
(green,)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0
(eggs,)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0
(and,)	1.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Step 6: Add vocabulary length to the sum of each row:

	i	am	sam	</s>	<s>	do	not	like	green	eggs	and	
(<s>,)	4.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	15.0
(i,)	1.0	4.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	15.0
(am,)	1.0	1.0	3.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	14.0
(sam,)	2.0	1.0	1.0	4.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	15.0
(</s>,)	1.0	1.0	1.0	1.0	4.0	1.0	1.0	1.0	1.0	1.0	1.0	14.0
(do,)	1.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0	1.0	12.0
(not,)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0	12.0
(like,)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0	12.0
(green,)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	1.0	12.0
(eggs,)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	2.0	12.0
(and,)	1.0	1.0	2.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	12.0

Step 7: Divide each element by the output of step 6 to create a probability matrix:

	i	am	sam	...	green	eggs	and
(<s>,)	0.266667	0.066667	0.133333	...	0.066667	0.066667	0.066667
(i,)	0.066667	0.266667	0.066667	...	0.066667	0.066667	0.066667
(am,)	0.071429	0.071429	0.214286	...	0.071429	0.071429	0.071429
(sam,)	0.133333	0.066667	0.066667	...	0.066667	0.066667	0.066667
(</s>,)	0.071429	0.071429	0.071429	...	0.071429	0.071429	0.071429
(do,)	0.083333	0.083333	0.083333	...	0.083333	0.083333	0.083333
(not,)	0.083333	0.083333	0.083333	...	0.083333	0.083333	0.083333
(like,)	0.083333	0.083333	0.083333	...	0.166667	0.083333	0.083333
(green,)	0.083333	0.083333	0.083333	...	0.083333	0.166667	0.083333
(eggs,)	0.083333	0.083333	0.083333	...	0.083333	0.083333	0.166667
(and,)	0.083333	0.083333	0.166667	...	0.083333	0.083333	0.083333

Step 8: Now find the value for (sam|am) which is 0.214286

5. Suppose we didn't use the end-symbol </s>. Train an unsmoothed bigram grammar on the following training corpus without using the end-symbol </s>:

```
<s> a b
<s> b b
<s> b a
<s> b b
```

Demonstrate that your bigram model does not assign a single probability distribution across all sentence lengths by showing that the sum of the probability of the four possible 2 word sentences over the alphabet {a,b} is 1.0, and the sum of the probability of all possible 3 word sentences over the alphabet {a,b} is also 1.0.

Solution:

Applying all the steps in 4 above without using smoothing and removing </s> token produce the following table:

	a	b	<s>
(<s>,)	0.500000	0.500000	0.000000
(a,)	0.333333	0.333333	0.333333
(b,)	0.250000	0.250000	0.500000

The user can now demonstrate it.

6. Suppose we train a trigram language model with add-one smoothing on a given corpus. The corpus contains V word types. Express a formula for estimating P(w3|w1,w2), where w3 is a word which follows the bigram (w1,w2), in terms of various N-gram counts and V. Use the notation c(w1,w2,w3) to denote the number of times that trigram (w1,w2,w3) occurs in the corpus, and so on for bigrams and unigrams.

Solution:

$$P(w_3 | w_1, w_2) = \frac{c(w_1,w_2,w_3)+1}{c(w_1,w_2)+V}$$

7. We are given the following corpus, modified from the one in the chapter: If we use linear interpolation smoothing between a maximum-likelihood bigram model and a maximum-likelihood unigram model with λ1 = 1/2 and λ2 =1/2, what is P(Sam|am)? Include <s> and </s> in your counts just like any other token.

```
<s> I am Sam </s>
<s> Sam I am </s>
```

```
<s> I am Sam </s>
<s> I do not like green eggs and Sam </s>
```

Solution:

$$P(w_2 \mid w_1) = \lambda_1 * P(w_2 \mid w_1) + \lambda_2 * P(W_2)$$

According to the equation:

$$P(w_2 \mid w_1) = (0.5 * ⅔) + (0.5 * 4/25) = 0.4133$$

8. Write a program to compute unsmoothed unigrams and bigrams.

Solution:

The python code is available on the GitHub repo with the name Chapter03_08.py

9. Run your n-gram program on two different small corpora of your choice (you might use email text or newsgroups). Now compare the statistics of the two corpora. What are the differences in the most common unigrams between the two? How about interesting differences in bigrams?

Solution:

I recommend the user to run the code and observe the changes

10. Add an option to your program to generate random sentences.

Solution:

The function to generate the random sentences is available on the GitHub repo:

Chapter03_10.py

11. Add an option to your program to compute the perplexity of a test set.

Solution:

The function to compute the perplexity is available on the GitHub repo:

Chapter03_11.py

12 Given a training set of 100 numbers consists of 91 zeros and 1 each of the other digits 1-9. Now we see the following test set: 0 0 0 0 0 3 0 0 0 0. What is the unigram perplexity?

Solution:

$$PP(W) = \sqrt[m]{\prod_{i=1}^{m} \frac{1}{P(w_i|w_{i-1})}}$$

Test Set = 0 0 0 0 0 3 0 0 0 0

m = 10

P(0) = 91/100 = 0.91 * 9 =

P(3) = 1/ 100 = 0.01

PP(W) = $\sqrt[100]{(0.91 * 9) * 0.01} = 0.081^{-1/10} = 1.285735094$