SNLP exercise set 03: character n-grams

Ç. Çöltekin — SfS / University of Tübingen

May 19, 2017

In this exercise set, you will work with character n-grams. The data set is the same as the data sets used in the previous two exercises.[1] You may want to filter CHAT-specific special characters within words better than earlier exercises.

Exercise 1. (**) Using the file `Adam/adam55.cha`, calculate the necessary frequencies (counts) for estimating two bigram models over letters (characters). You will use two models, one for the words uttered by the child, the other for the words uttered by the mother. Since we want to have a proper probability distribution over words, do not forget to include an 'end of word' symbol, and for convenience, also include a 'beginning of word' symbol.

Do not calculate probabilities yet. Count the necessary objects, and store them along with their counts in an appropriate data structure.

Exercise 2. Print out your bigram counts you collected in Exercise 1 as a table whose rows correspond to the first letters in the bigram, and the columns correspond to the second letter. For example, according to example output in Table 1, the bigram `aa` occurs once, the bigram `ab` occurs twice, and the bigram `ba` occurs 8 times.

Based on your output, what is the most frequent fist and last letters of the mother?

Table 1: Example output for Exercise 2.

	a	b	c	d	…
a	1	2	3	4	…
b	8	7	6	5	…
c	10	11	12	13	…
d	0	16	15	14	…
⋮	⋮	⋮	⋮	⋮	⋱

Exercise 3. Based on both the child's and the mother's bigram models, calculate the relative frequency estimates (MLE) of the probabilities of the following words: *i*, *you*, *my*, *your*, *mommy*, *yes*, *no*, *good*, *bad*, *wug*, *nlp*.

Are the probabilities of words 'yes' and 'no' comparable (for the same speaker)?

Exercise 4. Calculate the probabilities for the words in Exercise 3, but this time estimate the probabilities using Laplace smoothing.

Exercise 5. Calculate the most likely two-letter word that may be uttered by the mother based on the MLE bigram model.

How many comparisons do you need to make if you want to find the most likely 10-letter word?

For this question consider a brute-force approach here, we'll discuss (much) better approaches later.

Exercise 6. (***) Sample 100 words from the child's bigram model. Tips: Remember that n-gram provide parameters of a multinomial distribution. What you want to do here is roughly

1. set `ch` to 'beginning-of-word' character
2. get a single sample the distribution $P(w_i|w_{i-1} = \texttt{ch})$
3. output or store the sampled character, set `ch` to the sampled character
4. go to step 2 until you reach the 'end-of-word' character

You need `numpy.random.multinomial()`. `numpy.flatnonzero()` may also be helpful.

[1] See exercise set 1 for the details on how to obtain the data set.

5. repeat all of the above as many times as the number of words to be generated

Exercise 7. (****, optional) Repeat Exercise 6 using a trigram model.