IWS 10-11. STATISTICAL INFERENCE.

Exercise 6.1 [★★]

Explore figures for the percentage of unseen n-grams in test data (that differs from the training data). Explore varying some or all of: (i) the order of the model (i.e., n), (ii) the size of the training data, (iii) the genre of the training data, and (iv) how similar in genre, domain, and year the test data is to the training data.

Exercise 6.2 [★]

As a smaller example of the problems with Laplace's law, work out probability estimates using Laplace's law given that 100 samples have been seen from a potential vocabulary of 1000 items, and in that sample 9 items were seen 10 times, 2 items were seen 5 times and the remaining 989 items were unseen.

Exercise 6.3 [⋆]

Show that using ELE yields a probability function, in particular that

$$\sum_{w_1 \cdots w_n} P_{\text{ELE}}(w_1 \cdots w_n) = 1$$

Exercise 6.4 [⋆]

Using the word and bigram frequencies within the Austen test corpus given below, confirm the ELE estimate for the test clause *she was inferior to both sisters* given in section 6.2.2 (using the fact that the word before *she* in the corpus was *person*).

w	$C(w)$	$w_1 w_2$	$C(w_1 w_2)$
person	223	person she	2
she	6,917	she was	843
was	9,409	was inferior	0
inferior	33	inferior to	7
to	20,042	to both	9
both	317	both sisters	2

Exercise 6.5 [⋆]

Show that Good-Turing estimation is well-founded. I.e., you want to show:

$$\sum_{w_1 \cdots w_n} P_{\text{GT}}(w_1 \cdots w_n) = \frac{f_{\text{GT}}(w_1 \cdots w_n)}{N} = 1$$

Exercise 6.6 [⋆]

We calculated a Good-Turing probability estimate for *she was inferior to both sisters* using a bigram model with a uniform estimate of unseen bigrams. Make sure you can recreate these results, and then try doing the same thing using a trigram model. How well does it work?

Exercise 6.7 [⋆⋆]

Build language models for a corpus using the software pointed to on the website (or perhaps build your own). Experiment with what options give the best language model, as measured by cross-entropy.

Exercise 6.8 [⋆⋆]

Get two corpora drawn from different domains, and divide each into a training and a test set. Build language models based on the training data for each domain. Then calculate the cross-entropy figures for the test sets using both the language model trained on that domain, and the other language model. How much do the cross-entropy estimates differ?

Exercise 6.9 [⋆⋆]

Write a program that learns word n-gram models of some text (perhaps doing smoothing, but it is not really necessary for this exercise). Train separate models on articles from several Usenet newsgroups or other text from different genres and then generate some random text based on the models. How intelligible is the output for different values of n? Is the different character of the various newsgroups clearly preserved in the generated text?

Exercise 6.10 [⋆⋆]

Write a program that tries to identify the language in which a short segment of text is written, based on training itself on text written in known languages. For instance, each of the following lines is text in a different language:

doen is ondubbelzinnig uit
pretendre à un emploi
uscirono fuori solo alcune
look into any little problem

If you know a little about European languages, you can probably identify what language each sample is from. This is a classification task, in which you should usefully be able to use some of the language modeling techniques discussed in this chapter. (Hint: consider letter n-grams vs. word n-grams.) (This is a problem that has been investigated by others; see in particular (Dunning 1994). The website contains pointers to a number of existing language identification systems – including one that was originally done as a solution to this exercise!)