Tutorial 2: Information Theory

Advanced String Substitution

When using `sed` or `perl` to substitute one character string with another, it can sometimes be useful to store the value that was matched, and re-use it in the replacement string. To do this, you enclose the part of the first pattern which you want to store between parentheses, and you re-use it on the right by writing `$1` (or `$2` for the second group in parentheses, and so on). For example, to match every character and output it followed by a `#` sign, you could use:

```
$ perl -pe 's/(.)/$1#/g' FILE | less
```

(Note: `sed` can do this, too, but the syntax is a little more cumbersome.)

Exercises

For each question, give **your calculations, the commands you used** (where relevant)**, and the actual answer**. You may hand in the tree for question 2 on paper in the lecture.

1. Suppose you have a corpus of size 74, which has 10 word types, each with the following frequencies. Based on the unigram probability distribution, estimate the per-word entropy. You can use R to calculate this.

John	Think	Said	Mary	Saw	Bill	Hit	Tom	Likes	Period
7	6	4	8	9	1	13	4	6	16

2. Devise the best binary code you can for the above language, and show the code tree. What is the average number of bits required to send a message, using your code? Why is it different from the theoretical lower bound you computed in Question 1?
3. Calculate the **per character** entropy for English and German, using the two corpora from the earlier tutorials (german.txt and english.txt). To simplify your answer, do the following: Convert all letters to lower case, convert the space to an underscore (_), and ignore all other (non-letter) characters. Use shell scripting to count letters etc, and then do the calculations using R.
4. For question 3, you determined the probability mass function for $P_{german}(X)$ and $P_{english}(X)$ for the random variable X={a,..,z, _}. (Again, use R.)
   - Now compute the relative entropy: $D(P_{german} \| P_{english})$
   - Comment briefly (2-3 sentences) on what this number tells us.
   - Compute $D(P_{english} \| P_{german})$. Is D symmetric or non-symmetric?
5. Assume $P_{english}$ provides a true model of the random variable X, the per-character probability mass function for English. However you have to use a model, q, based on letter frequencies observed in the very small 'example' corpus (example.txt). Compute the cross entropy of your model: H(X,q). Use R.
   Important hint: If any letters do not occur in the 'example' corpus, assume they have a frequency of 1 (one). You can do this with "join", using the options -e, -a and -o in addition to the options -1 and -2, which you already know.
   The -o option lets you specify which column of which file you want to place in which order, for example -o '1.2 1.1 2.1' will print the second column of the first file as the first column, followed by the first column of the first file, followed by the first column of the second file. The -e option can only be used in combination with the -o option.