NTMI 2016: Project Exercises
Part A - Step 2

This project has four steps. The steps are gradual and allow you to build the project in blocks that you can re-use in subsequent steps.

STEP 2: Markov language models using ngram statistics
Due 9th Feb (Tue) 2016 before midnight

For this exercise, assume that the beginnings and ends of paragraphs correspond to the beginnings and ends of sentences (that is, each paragraph is one sentence). Therefore, add a $START/END$ symbol at the beginning and end of each paragraph.

Write a program that performs the following tasks on an input corpus (test this on the AUSTEN TRAIN corpus `http://www-nlp.stanford.edu/fsnlp/statest/austen.txt`).

1. Given a natural number n and a corpus, the program constructs two tables: (1) $(n-1)$-grams and (2) n-grams from the corpus, together with their frequencies.

   Report the 10 most frequent bigrams in the corpus together with their frequencies.

   `NOTE: You may use the program from STEP 1 for this, but notice that by adding the START/STOP symbols, the frequencies may have changed.`

   Command line:
   `./a1-step2 -corpus [path] -n [value]`

2. The program should accept an additional file as input. This file should contains sequences of n words $w_1 \dots w_n$ (one sequence per line, words

separated by white space and ordered from left to right). Your program may ignore lines containing sequences of length other than n.

For each sequence $w_1 \dots w_n$ in the file, the program should report $P(w_n|w_1, \dots, w_{n-1})$ (the probability that the sequence $w_1, \dots, w_{n-1}$ is followed by w_n). The program should calculate this probability based on the $(n-1)$-gram and n-gram tables constructed from the corpus.

Command line:
`./a1-step2 -corpus [path] -n [value] -conditional-prob-file [path]`

3. The program should also accept a file with a list of sentences as input. Every line (=sentence) in this file is a list of words (separated by white space) of arbitrary length. For each sentence $w_1 \dots w_m$ in the file, the program should report the probability of the sentence based on the n-gram model estimated from the corpus. For this, the program should use the following formula:

$$P(w_1, \dots, w_m) = \prod_{i=1}^{i=m+1} P(w_i|w_{i-n+1}, \dots, w_{i-1})$$

where $w_j = START$ for $j \leq 0$ and $w_{m+1} = STOP$.

Remarks:

(a) If $n > 2$, this formula adds more than one $START$ symbol at the beginning of the sentence. This is the same as using shorter sequences for the first words of the sentence (for example, if $n = 3$, we can use $START$, w_1 instead of $START$, $START$, w_1 for the first word in the sentence).

(b) Because sentences are long, probabilities may be very small. Make sure small positive probabilities are not rounded to zero by the built-in floating point operations.

Command line:
`./a1-step2 -corpus [path] -n [value] -sequence-prob-file [path]`

4. Given are the following two sets of words:

$$A = \{know, I, opinion, do, be, your, not, may, what\}$$

$$B = \{I, do, not, know\}$$

(a) Write a procedure which reads in a set of words and generates all their permutations.

(b) For each such permutation, calculate the probability of that permutation as a sentence (that is, add start and stop symbols). For this calculation, use a first order Markov model ($n = 2$ of parts 1 - 3 above). For each set of words, report the two permutations with the highest probability.

(c) Report the output of this procedure on set A and set B given above.

Command line:

```
./a1-step2 -corpus [path] -scored-permutations
```