Assignment II: Next-word Probability, Trainset Likelihood

Natural Language Models and Interfaces 2012

11 April 2012

You are supposed to do this assignment during the lab session on the 11th of April and finish it in your own time before the session of April 17th.

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 $START/STOP$ 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 OVIS TRAIN corpus).

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.

2. The program should accept an additional file as input. This file should contain sequences of n words $w_1 \ldots w_n$ (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 $w_1 \ldots w_n$ in the file, the program should report $P(w_n|w_1, \ldots, w_{n-1})$ (the probability that the sequence $w_1, \ldots, w_{n-1}$ 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.

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 $w_1 \ldots w_m$ 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.

4. Example:

`corpus.txt`	`ngrams.txt`	`sentences.txt`

`A has number 1`
 	
`K with number 1 is better than A`
 	
`B has number 2`
 	
`L with number 2 is better than B`
 	
`C has number 3`
 	
`M with number 3 is better than C`
 	
`D has number 4`
 	
`N with number 4 is better than D`
 	
`E has number 5`
 	
`O with number 5 is better than E`
 	
`F has number 6`
 	
`P with number 6 is better than F`
	`is better`
	`number 1`
		`A has number 1`
		`K with number 1 is better than A`

```
./lab2 2 corpus.txt ngrams.txt sentences.txt
= 10 most frequent 2-grams =
is better 6
has number 6
better than 6
with number 6
number 1 2
number 3 2
number 2 2
number 5 2
number 4 2
number 6 2
= conditional probabilities =
1.0
0.16666666666666666
= sentence probabilities =
0.003472222222222222
0.0005787037037037037
```

5. Evaluate your model on the OVIS corpus. What are the most frequent bigrams and trigrams? What are the (log) likelihoods of each of the sentences under a bigram or a trigram language model? What is the most likely word to follow “ik wil van” and “ik wil niet van” under a bigram and trigram model? Generate candidate ngrams using `grep -o` and a regular expression like `ik wil van [a-z]\+`.