CS4248
AY 2013/14 Semester 1
Assignment 2

Due Date

Friday 25 October 2013 3pm. Late assignments will not be accepted and will receive ZERO mark.

Objective

In this assignment, you are to write a program (in Java, C++, or C) to perform context-sensitive spelling correction. This task is to detect spelling errors that result in valid words. An example of such a spelling error is the word *raise* in the following sentence:

The price will *raise* to $100.

The correct word is *rise*. In this assignment, this task will be formulated as disambiguation between two words in a confusion set. Examples of such confusion sets include { *adapt*, *adopt* }, { *formally*, *formerly* }, { *raise*, *rise* }, etc.

Specifically, you will implement a program to determine whether a word *w* in a given confusion set { w_1, w_2 } should be disambiguated as w_1 or w_2. The task is formulated as a supervised learning task from labeled training sentences. You are to implement a Bayesian classifier for context-sensitive spelling correction, making use of the naïve Bayes assumption. The features used include:

(a) Surrounding words: Each word that appears in the sentence containing the confusable word *w* is a feature. All surrounding words are converted to lowercase, and stop words and punctuation symbols are removed.

(b) Collocations: A collocation $C_{i,j}$ is an ordered sequence of words in the local, narrow context of the confusable word *w*. Offsets *i* and *j* denote the starting and ending positions (relative to *w*) of the sequence, where a negative (positive) offset refers to a word to its left (right). Each collocation string spanning positions *i* to *j* is a feature. You get to decide on the list of collocations (i.e., what pairs of offsets) to use.

Training and Test Data

You are provided with a list of training sentences in which a confusable word *w* appears. Sample training sentences are shown below:

```
0011 And antitrust regulators could >> raise << barriers .

0077 The rate may not >> rise << more than five percentage
points over the life of the loan .
```

`0011` and `0077` are sentence ids. The confusable word is preceded by >> and followed by <<. Note that each sentence has been tokenized (punctuation symbols are separated from words).

The command for training the Bayesian classifier is:

```
java sctrain word1 word2 train_file model_file
```

The file `train_file` is a file containing the training sentences. The file `model_file` contains the statistics gathered from the training process. `word1` and `word2` are the confusable words (which are `raise` and `rise` in the above example).

The format of the test file is the same as the training file, except that the confusable word is not given. A sample test sentence is shown below:

```
1048 Bond funds , hammered by the >> << in interest rates this
year , continue to have outflows across the board , fund groups
say .
```

The command to test on this test file and generate an output file is:

```
java sctest word1 word2 test_file model_file answer_file
```

For each test sentence in `test_file`, `answer_file` contains one line indicating the test sentence id and the disambiguated confusable word as determined by the Bayesian classifier:

```
1048 rise
```

Your program will be evaluated by training it on sets of confusable words, and testing the program on a set of new, blind test sentences containing the confusable words to be disambiguated. Your program must be general such that it is able to disambiguate any pair of confusable words.

All program execution and testing will be done on sunfire.

Deliverables

You are to work on this programming assignment ***individually***. You must hand in the following:
1. Your source code (with documentation). Note that graphical user interface is not needed.
2. A short report (no more than 4 pages, Times New Roman 12 point font) describing details of the method used to build your program, your own testing conducted, and the test results obtained. Marks will be deducted for a report that is longer than 4 pages.

Email your program and report to cs4248@comp.nus.edu.sg by the due date. Your program and report should be emailed as one single attached zip file, with the email subject line clearly indicating your name (as it appears on your student ID card), and your matriculation number. An example email subject line is:

CHIONG CHUN HONG, KELVIN U061234A

In addition, submit a hardcopy of your report to the lecturer in class by the submission deadline.

Grading

60% for your report, which includes the method used to build your program and your test results obtained on the 3 given confusion sets ({ *adapt*, *adopt* }, { *formally*, *formerly* }, { *raise*, *rise* }).

40% for the accuracy of your program, as measured by the number of new, blind test instances correctly disambiguated by your program.