CS4248
AY 2017/18 Semester 1
Assignment 2

Due Date

Friday 13 October 2017 at 3pm. Late assignments will not be accepted and will receive ZERO mark.

Objective

In this assignment, you are to write a program (in Java, Python, C++, or C) to perform part-of-speech (POS) tagging. Given a sentence, the task is to assign a POS tag to each word in the sentence. We will adopt the Penn Treebank tag set in this assignment.

For example, given the following input sentence:

```
He also is a consensus manager .
```

The POS tagger should output the following:

```
He/PRP also/RB is/VBZ a/DT consensus/NN manager/NN ./.
```

Note that each input sentence has been tokenized (punctuation symbols are separated from words).

You are to implement a POS bigram tagger based on a hidden Markov model, in which the probability of the POS tag of the current word is conditioned on the POS tag of the previous word. The Viterbi algorithm will be used to find the optimal sequence of most probable POS tags.

Training and Testing

You are provided with a training set of POS-tagged sentences (`sents.train`). You will train your POS tagger using this training set. A separate development set of sentences is also provided (`sents.devt`) for tuning the POS tagger.

The command for training (and if necessary tuning) the POS tagger is:

```
java build_tagger sents.train sents.devt model_file
```

The file `model_file` contains the statistics gathered from the training (and if necessary tuning) process, which include the POS tag transition probabilities and the word emission probabilities (and other tuned parameters if necessary).

The test file consists of a list of sentences (without POS tags), one sentence per line. A sample test file is provided (`sents.test`).

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

```
java run_tagger sents.test model_file sents.out
```

The output file has the same format as the POS-tagged training file. A sample output file is also provided (`sents.out`).

The commands `build_tagger` and `run_tagger` as shown above will be executed to evaluate your POS tagger, except that testing will be done on a set of new, blind test sentences. 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 your implemented method.

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:

LIM YONG SHENG A0103123H

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

Grading

60% for your report, which includes details of how you build your POS tagger (how unknown words are handled, the smoothing method used, accuracy of 10-fold cross validation on the training set, etc).
40% for the accuracy of your POS tagger, as measured on the blind test set of sentences.