Fall Semester 2016
Iowa State University

ENGL 520 - Computational Analysis of English

Assignment 3
Submission Deadline: 15 OCT 2016, end of the day
Note: This assignment carries **20** marks.

Instructions: This assignment consists of three questions. First question carries 10 marks and the other two carry 5 marks each (Please note: This assignment is for 20 marks, unlike others which are for 15 marks!). First two questions require you to write a program. You can use any programming language you want. If any of the programs does not run and throws errors, you cannot get a credit for that. Third question does not involve programming, but involves analysis of existing POS tagging software. Late submissions will not get full credit. Upload your assignment as a zip file.

Question 1

Using the training data file uploaded with this assignment, write a program that creates trigram probability estimates for POS tags. Now, write a second program that accepts a string as input from user, and predicts the POS tags for words in the string based on those simple trigram estimates. Your tagger need not be 100% accurate. But, it should run and output some tags for words in a sentence, and should be able to handle cases where there are new words not seen in training data (hint: use some smoothing). Add comments in your code appropriately.

Question 2

1. Question 2a - for Python programmers: Use one of the taggers in nltk, and write a python program that accepts input sentences from users, and does part of speech tagging for these sentences using the tagger. Refer to Chapter 5 in NLTK book to know what taggers are available and how to use them.

2. Question 2b - Install one of these perl modules: Lingua::EN::Tagger or Lingua::TreeTagger, and write a perl program that accepts input sentences from users, and does part of speech tagging for these sentences using the tagger. Follow the documentation for these two modules to know how to work with these taggers.

3. Note for both programs: the programs should take input from user interactively until the user types "quit".

Question 3

Here are the links to some online demos for POS taggers developed by NLP researchers:

- `http://cogcomp.cs.illinois.edu/page/demo\_view/pos`
- `http://nlp.stanford.edu:8080/parser/` - look at only the tagger output in this demo.
- `http://morphadorner.northwestern.edu/postagger/example/`
- `http://ucrel.lancs.ac.uk/claws/trial.html`

Pick any two taggers from these, and do the following:

1. Create a list of 10 sentences written by English learners at an Elementary proficiency (You can use any sentences from corpus resources you already have. Or pick something from `http://cblle.tufs.ac.jp/llc/icci/search.php?menulang=en`. Or contact me for any other learner corpus resources).

2. Run the taggers on these 10 sentences, and write what you think about the robustness of taggers to such ungrammatical input (and misspelt words). You should write about about both the taggers, and compare their performance. Make sure you have two types of sentences in your chosen list: a) sentences that are incorrect (mis-spelt or ungrammatical), yet, do not affect POS tagging. b) sentences that are incorrect and affect POS tagging.

Note: The taggers may or may not follow the same tagset. Some taggers by nature give fine grained tags, and some do not. Your task is to just notice how accurate they are with learner text and compare the taggers in terms of accuracy, not on the differences between the granularity of tags. Your commentary can be up to 2 pages long. List the sentences you tried with at the end of this document, as an appendix in an additional page.