CS1671 Human Language Technology
Assignment 3

1 Part 1

Part 1 covers these concepts/methods:

- Part of Speech (POS) Tagging
- Training and Testing Hidden Markov Models
- Parameter estimatation from a corpus, including smoothing
- The Viterbi algorithm

This question takes you through the overall process of Hidden Markov Model POS tagging using the Viterbi algorithm. Throughout this question, assume all the POS tags are Penn Treebank tags. Please make sure your answers are understandable.

Here is a sample of the training data. Note that the data has been tokenized, and there is one sentence per line.

Until/IN Congress/NP acts/VBZ ,/, the/DT government/NN has/VBZ n't/RB any/DT authority/NN to/TO issue/VB new/JJ debt/NN obligations/NNS of/IN any/DT kind/NN ,/, the/DT Treasury/NP said/VBD ./.

- Pick out two words in this sentence that are ambiguous. For each, give another possible tag.
- Sometimes, the test data also have POS tags included. When would we have POS tags in the test data? When would we not have them in the test data?
- To train your model:

  Exactly what counts must be gathered as you read in the training data? (Hint: there are four types of counts needed.) Please give your answer (and your answers below) in the form of dictionaries/hashtables, and give one example of each type of count. For example, in Python I might write countType1[('NN','VB')] (but use a better name, and state what the count type is).
- After you have read in the training data, you need a couple of loops to set some entries in two of your hash tables to 0. Write pseudo-code to do that.
- Give pseudo-code for calculating the probabilities needed to define the model. Use LaPlace ("add one") smoothing.

- Suppose the input sentence is "bear is with" Suppose we have only three POS tags, "NN", "VB", and "Other", and three words, "bear", "is", "with". Draw the Hidden Markov Model (in the same format as we did in class). Make up some "reasonable" probabilities - use your judgement, but make sure that the numbers that should sum to 1 do so. Recall:

$$\sum_a P(A = a|B = b) = 1$$

  Note that "bear" can be a verb as well, as in, "bear with me."

- Now, fill up the scores table with the appropriate values (either by hand or via typing). Please do not multiply them out. The entries in the table should be, for example, 0.2 * 0.25 * 0.9 Put the numbers in the same order as they are in the algorithm in the slides (to help with grading).

- Finally, show the sequence of most probable tags Viterbi produces.

2 Part 2

This section gives you practice with syntax and context-free grammars.

On the schedule, there are a program that generates random sentences according to a grammar (generate.py), and an example input grammar (generateInput.txt).

Your task is to modify the grammar so that it handles everything it currently does, but also handles 3 of the 5 things listed below. Your grammar should not generate any non-grammatical sentences with respect to the 3 things you fix.

Since you are creating a context-free grammar, the only way to handle dependencies is to multiply the POS tags and non-terminals (e.g., create additional subtypes).

Run generate.py with your grammar enough times that the capabilities of your grammar are illustrated. Please add a key (manually typed in) at the top of the file listing and briefly describing the new tags you create. Manually mark output sentences that show the capabilities of your grammar. Use the codes indicated below (e.g., N: for number agreement in noun phrases). Move the sentences around in the file so that the sentences illustrating, e.g., number agreement ("N:") are next to each other. The TA should be able to easily check off the capabilities of your grammar. If you mark too few, she will not be convinced and she will need to hunt for additional examples; if you mark too many, then it will be excessively time-consuming to go through your output.

The TA will also run generate.py on your grammar.

What to hand in: Your grammar (the file name should be your last name + ".grammar.txt"), and your annotated output (the file name should be your last name + ".output.txt").

Your vocabulary should include at least the following:

- Verbs: go, be, have, sleep, play, think, throw, read, persuade

- Pronouns: I, me, you, they, them, it, he, she, him, her
- Prepositions: in, on, to, at, with, by
- Nouns: store, game, stone, boy, dog, person, stick, house, book
- Proper nouns: Mary, Juan, Qiu (male name), Kalyani (female name)
- Adjectives: nice, smart, funny, quick, fun
- Determiners: the, a, some, that

Your grammar should be able to handle 3 out of the following. As stated above, it should not generate any ungrammatical sentences with respect to the three things you fix (though meaningless sentences are fine).

- N: Number agreement within noun phrases.

  E.g. Good: the boy, the boys, some boys, boys
  E.g. Bad: a boys, boy

- A: Proper use of adjectives:

  E.g. Good: The tall boy, The boy is funny

- PP: PP adverbial modifiers:

  E.g. Good: She slept by the garage. He thought that Mary was sweet in the garage (with attachment of "in the garage" to the verb, i.e., he had that thought when he was in the garage).

- G: Gerunds:

  E.g. Good: Playing games is fun; Thinking that dogs are fun is funny

- Pro: Proper use of pronouns:

  number: singular (sg), plural (pl)
  person: first (1st), second (2nd), third (3rd)
  case: nominative (subject), accusative (object)
  gender: masculine (he), feminine (she), neuter (it)

- Pass: Passive constructions:

  E.g. Good: The book was read by the boy; The book was thrown by the dog; The stick was thrown.

  E.g. Bad: The book was slept. She was gone by the boy.