CS6320, Fall 2016
Dr. Mithun Balakrishna
Homework 2
Due October 9th, 2016 11:59pm

A. Submission Instructions:

- Submit your solutions via eLearning.
- Please submit a single zip file with the following files:
  - For programming questions:
    - Source code file(s) in C/C++, Java, or Python. For using any other programming language, please get prior approval from the TA.
    - A ReadMe file with instructions on how to compile/run the code.
  - For all other questions, a PDF/Doc/PS/Image file with the solutions.
- Late Submission Penalty:
  - up to 2 hours late — 10% deduction
  - 2 - 4 hours late — 20% deduction
  - 4 - 12 hours late — 35% deduction
  - 12 - 24 hours late — 50% deduction
  - 24 - 48 hours late — 75% deduction
  - more than 48 hours late — 100% deduction (zero credit)

B. Problems:

1. POS Tagging Errors (10 points)

Find one tagging error in each of the following sentences that are tagged with the Penn Treebank POS tagset (Figure 5.6):

1. I/PRP need/VBP a/DT flight/NN from/IN Atlanta/NN

   `Solution: Atlanta/NNP`

2. Does/VBZ this/DT flight/NN serve/VB dinner/NNS

   `Solution: dinner/NN`

3. I/PRP have/VB a/DT friend/NN living/VBG in/IN Denver/NNP

   `Solution: have/VBP`

4. Can/VBP you/PRP list/VB the/DT nonstop/JJ afternoon/NN flights/NNS

   `Solution: Can/MD`

2. Rule Based POS Tagging (30 points)

For this question, you have been given a POS-tagged training file, *HW2_F15_NLP6320_POSTaggedTrainingSet.txt*, that has been tagged with POS tags from the Penn Treebank POS tagset (Figure 5.6). Use this POS tagged file to compute for each word *w* the tag *t* that maximizes P(*t*|*w*). Retag the training file with POS tags that are most probable for a given word. Compute the error rate by comparing the retagged file against the original tagged file. Now perform error analysis to find the top-5 erroneously tagged words. Write at least five rules to do a better job of tagging these top-5 erroneously tagged words, and show the difference in error rates.

Solution: Programming question. Answers are implementation dependent.

3. HMM Decoding: Viterbi Algorithm (30 points):

Implement the Viterbi algorithm and run it with the HMM in Fig. 6.3 to compute the most likely weather sequences for each of the two observation sequences, *331122313* and *331123312*.

Solution: Programming question.

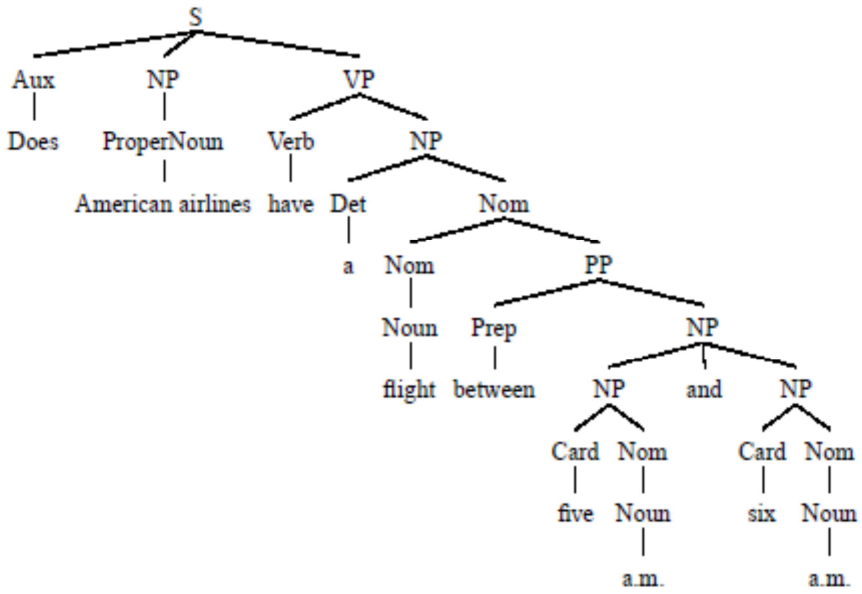
Weather sequence for *331122313*: HHCCHHHHH with probability 3.575714750873601e-005

Weather sequence for *331123312*: HHCCHHHHH with probability 3.9516275425280015e-005

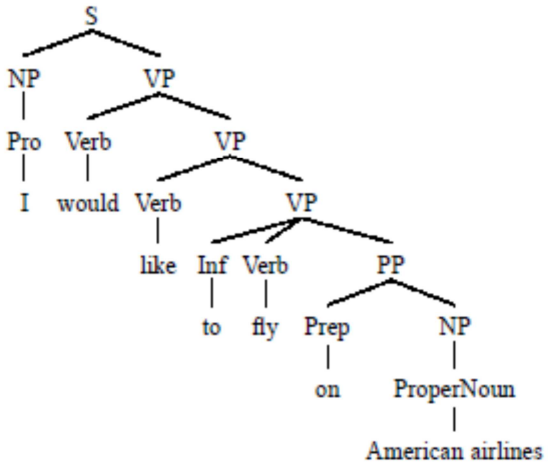
4. Parse Trees (30 points):

Draw tree structures for the following sentences (use the rules from the Formal Grammars lecture and Chapter 12):

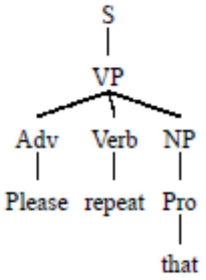
1. Does American Airlines have a flight between five a.m. and six a.m.?



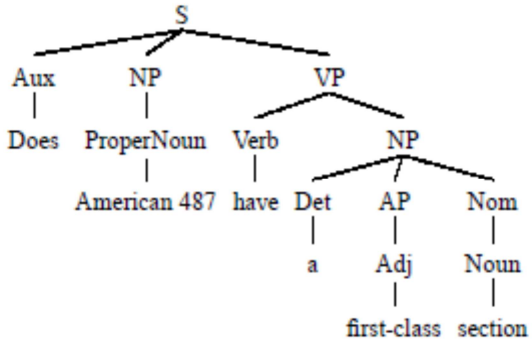
2. I would like to fly on American airlines.



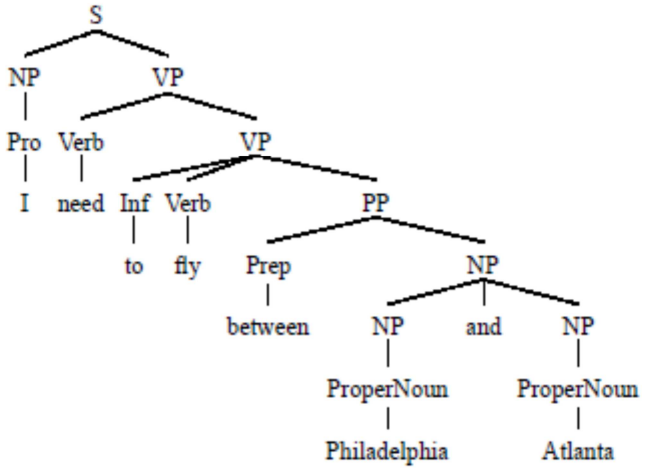
3. Please repeat that.



4. Does American 487 have a first-class section?



5. I need to fly between Philadelphia and Atlanta.



6. What is the fare from Atlanta to Denver?