Exam 2

CS 322: Natural Language Processing

Fall 2015

Name: _____

Read all of the following information before starting the exam:

- Show all work, clearly and in order, if you want to get full credit. I reserve the right to take off points if I cannot see how you arrived at your answer (even if your final answer is correct).
- Circle or otherwise indicate your final answers.
- Please keep your written answers brief; be clear and to the point. I reserve the right to take points off for rambling and for incorrect or irrelevant statements.
- This exam has 4 problems and is worth 65 points.
- *This exam is open book and open web, but closed people.* Remember to cite all sources.
- Good luck!

1	2	3	4	Σ
15	20	10	20	65

1. Signal Analysis (*15 points*) For the following functions, suppose that we looked at a windowed version of the function from x = 0.5 to x = 1.0. If we pass the samples in that window to our Fourier Transform function, which bins (in the lower half of the transform output) will have the highest root-mean-squared magnitude? (Give bin numbers.)

a. (*5 pts*) $a(x) = cos(8x\pi) + sin(4x\pi)$

b. (*5 pts*) $b(x) = \frac{3}{4}sin(8x\pi - \frac{6\pi}{7})$

c. (*5 pts*) $c(x) = sin(8x\pi - \frac{3\pi}{4}) - cos(16x\pi + \frac{3\pi}{4})$

2. Parsing (*20 points*) Consider the following grammar:

```
S -> NP VP
NP -> Det Noun
NP -> Noun
NP -> NP PP
NP -> NP and NP
VP -> VP PP
VP -> VP NP PP
VP -> VP NP
VP -> Verb
# VP -> Aux VP PP     THIS LINE WAS IN ERROR
# THE FOLLOWING LINE IS ITS CORRECTION, SORRY
VP -> Aux VP NP
PP -> Prep NP

Det -> the | a | an
Prep -> in | like | with
Verb -> like | can | fool | catch
Aux -> can | may | will
Noun -> i | cats | dogs | can
```

a. (*5 pts*) Show your transformation to a legal grammar that would be needed for the CKY or Earley parsing algorithm (use the one that you did not write for HW8.)

b. (*10 pts*) Given the sentence:

```
i can like cats and dogs
```

Show the status of the parser's internal variables (i.e. the Chart for Early or the Grid for CKY) after running the algorithm. Again, show the algorithm that you did not write for HW8. You need not mark the backpointers that would be needed to reconstruct your tree.

c. (*5 pts*) What are all legal parse trees for the sentence, under your transformed grammar?

3. Long Answer (*10 points*)

You are building a language model, and you decide to integrate a probabilistic parser in with a bigram language model to help word prediction. Suppose that you have a large treebank data set to train on. Describe how you could build such a model, i.e. What is involved in the calculation of $P(w_i|w_1w_2...w_{i-1})$?

I expect a few paragraphs at least. Make sure to address training of the various models and how the bigram system will combine with the parser.

4. HMM (*20 points*) On Moodle, there is a file named `topics.zip`. Download it and unzip it, and you will find a text file `topicstrimmed.txt` that contains a stream of words, labeled by topic. Consider the words as a single stream of observed states. The data comes from Usenet groups corresponding to six topics (religion, cars, guns, baseball, medicine, windows.) The data has punctuation stripped and all text converted to lower case. The first word of each line marks its topic, the second word is from the text.
Your goal is to create an HMM that models changing topics, where the six hidden states correspond to the six topics. Write code to build an HMM for the data, using no smoothing. You may assume that the start state is chosen to be any of the six states with equal probability.
Generate your answers to this question using any code that you write from scratch. Do not reuse code for this problem. Submit code to Moodle as `exam2hmm.py` that, when run with `topicstrimmed.txt` as a command-line argument, outputs answers to the following questions:

a. (*3 pts*) What is the **transition** probability P(baseball | religion)?

b. (*3 pts*) What is the **transition** probability P(windows | windows)?

c. (*3 pts*) What is the **emission** probability P(god | medicine)?

d. (*3 pts*) What is the **emission** probability P(religion | baseball)?

e. (*8 pts*) Suppose that the two-word sequence "com re" is observerd. For each of the 36 hidden state sequences, report the probability that the observation is generated by that sequence. Report your results in descending order.

You code output should be neat and readable!