CS6320, Fall 2015
Dr. Mithun Balakrishna
Homework 1
Due October 4th, 2015 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. Regular Expressions (25 points)

Write regular expressions for the following. You may use either Perl/Python notation, but make sure to say which one you are using. By "word", I mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth.

1. the set of all alphabetic strings
2. the set of all lower case alphabetic strings ending in a b
3. the set of all strings with two consecutive repeated words (e.g., "Humbert Humbert" and "the the" but not "the bug" or "the big bug")
4. the set of all strings from the alphabet {"a", "b"} such that each "a" is immediately preceded by and immediately followed by a "b"
5. all strings that start at the beginning of the line with an integer and that end at the end of the line with a word

2. Money! (25 points)

Complete the FSA for English money expressions (Slide 91 or Fig. 2.15 in text book). You should handle amounts up to $100,000, and make sure that “cent” and “dollar” have the proper plural endings when appropriate. Formulate the problem precisely, making only those distinctions necessary to ensure a valid solution. Draw a diagram of the complete state space.

3. Unsmoothed Unigram/Bigram Probabilities (25 points):

Write a program to compute unsmoothed unigrams and bigrams. Your program’s correctness will be evaluated using an unseen corpus.

Your program should:

1. Accept as input: a plain text file containing tokenized text (one token/word per line). Please treat the word or words in each line as a single token for computing unigrams and bigrams. There is NO need to convert the words in any manner i.e. lower-casing, removal of punctuations, lemmatizing, stemming, etc.

   Sample input file:

   ```
   I
   like
   big
   big
   ice-creams.
   Big
   big
   ice-creams
   are
   the
   best
   thing.
   It’s
   ………
   ```

2. Produce as output the following two (2) files:

   i. text file containing the unigram probabilities (sorted alphabetically)

   Sample unigram probabilities file (the unigram probability values below are for demonstration purposes ONLY and are NOT real values):

are	0.12
best	0.19
big	0.11
Big	0.21
I	0.0001
ice-creams.	0.001
ice-creams	0.005
It's	0.10
like	0.002
the	0.3
thing.	0.20

………

   ii. text file containing the bigram probabilities (sorted alphabetically)

   Sample bigram probabilities file (the bigram probability values below are for demonstration purposes ONLY and are NOT real values):

are	the	0.11
best	thing.	0.31
big	big	0.019
Big	big	0.019
big	ice-creams.	0.010
big	ice-creams	0.012
ice-creams	are	0.146
ice-creams.	Big	0.015
I	like	0.022
It's	....	0.16
like	big	0.0167

the	best	0.14
thing.	It's	0.16
………		

4. **Smoothed Unigram/Bigram Probabilities (25 points):**
   Add an option to your program from Question 3 to do Good-Turing discounting. Your program’s correctness will be evaluated using the same unseen corpus as Question 3.