NLP Problem Set 1

SUBMISSION INSTRUCTIONS:

Due-date: The due-date for this homework assignment is as listed on Canvas (see "Due" within the Assignment page). The completed assignment needs to be submitted into by that due-date.

What to submit:

1. Self-grading text file named **selfgradesNLP_1.txt**. See instructions below. This must be **a separate file (i.e., must not be part of another file you submit within the assignment)**.
2. Solutions to written problems. This **must** be a single file and this file **must** be in **pdf** format. Scans of handwritten solutions are allowed, provided that the handwriting is clear, and provided that the file containing the scanned pages is a **pdf** file.

   Show all relevant steps leading to the solutions.

Do **not** zip, tar, or compress in any other way the files you submit. Any files you submit must be uncompressed. Any files you submit must be either in pdf (i.e., ".pdf') or text (i.e., ".txt") formats. Any file formats other than ".pdf' or ".txt" are **not** allowed.

GRADING:

This problem set is self-graded. We will randomly choose a subset of submissions to verify the grades.

How to self-grade problems: The maximal number of points you are allowed to claim for any particular problem is listed in the problem section-headings. You may claim partial credit for any of the written problems.

Reminder: Copying solutions from **any** source – e.g., online, solutions manuals, assignment solutions from prior semesters, or solutions from any other sources – is prohibited.

Self-grading file filling and submission instructions:

To receive any credit for this assignment, a separate self-grading text file **must** be included as part of this assignment submission. Use the self-grading text file that is provided with this assignment. You need to edit that file by filling in your self-assigned grades, and submit the edited file with your self-grades filled in. The submitted self-grading file must be **exactly** in format and according to the following specifications.

Self-grading file must contain **one line only** (i.e., single line inside the file). This line must contain text fields separated by be **commas**, i.e., comma-separated format. The first three comma-separated fields must contain (in this order) your last name, your first name, and your Northeastern University student email address, respectively. The remaining fields must contain your self-assigned grades for **all** gradable problems (in the order in which those problems appear in the assignment).

Gradable problem is any problem in this assignment for which a point-value (maximum number of points you can obtain for that particular problem) has been listed in the heading of that problem's respective section or subsection (look for square parentheses, e.g., [X points]).

The last field of the self-grading file line must contain your TOTAL self-grade for this homework assignment. Obviously, this total must be equal to the sum of all problem self-grades.

Do the following to complete your self-grading file:

1. Get the "starter" self-grading file, provided within this assignment on Canvas. This file contains the following single line:

   Lastname,Firstname,Email,

2. Replace last-name, first-name, and email-address fields with your data. You are reminded that the email address must be your Northeastern University email address.

3. Continuing this same single line, enter your self-grades for all gradable problems, in the order of gradable problems and including any zero-grades you may have self-assigned for any of the problems.

   Each self-assigned grade must be followed by a **single comma** (i.e., self-grade fields are comma-separated fields).

   Make sure that the number of these comma-separated self-grade fields equals the number of gradable problems. Do not skip self-grades of any gradable problems. If you have not attempted a particular problem, enter self-grade of (numerical zero) for that problem. Do **not** use any blanks.

4. Put a comma after the last self-grade field and then continue in the same line by entering the total self-grade for this homework. Do not add any characters after the total self-grade field. Other than filling the fields in the single line according to the specifications provided here, do **not** alter anything in the file format. Do **not** add any extra lines.

   Example:

   - Johnny Doe is listed in Northeastern University student records as: LAST=Doe, FIRST=Charles, MIDDLE=John.
   - Suppose a hypothetical homework assignment consisting of six gradable problems (six here is just an example) with their maximum point-values of 15, 15, 10, 20, 10, 30, respectively, and that Johnny self-graded them as 15, 0, 10, 20, 0, 30.
   - Johnny needs to change the line in the self-grade file, putting his data for last-name, first-name and email-address, appending (in the same line) comma-separated fields containing all self-grades (non-zeros as well as zeros, as applicable), and ending the line with the field containing the total self-grade for this homework. Therefore, Johnny changes the single line in the self-grading file to the following:

     Doe,Charles,doe_ch@northeastern.edu,15,0,10,20,0,30,75.

5. Prior to submitting your homework, check the self-grading file and make sure that it is completed **exactly** according to **all** of the instructions provided herein. Also, check that you did not overlook any gradable problems.

6. Be sure to submit your completed self-grading file together with your homework.

Please note that a credit of zero will be given for this homework if the self-grading file, filled in according to the instructions provided herein, is not submitted as part of your homework.

Written Problems

1 Regular Expressions [15 points]

Assume that "word" is defined as an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth.

Write regular expressions for the following:

a) 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").

b) The set of 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;

c) The set of all strings that have both the word grotto and the word raven in them (but not, e.g., words like grottos that merely contain the word grotto);

Note: Read sec. 2.1.5 and 2.1.6 of Jurafsky & Martin textbook.

2 Bag-of-words classifier [35 points]

Recall the following in context of Bag-of-Words classifier:

$$\hat{y} = \text{argmax } \psi\left(x, y \cdot \theta\right), \quad y \in Y$$

Show in detail how $\hat{y}$ can be represented using feature function $f(x, y)$.

3 BOW probabilities [35 points]

Let x be a bag-of-words vector such that $\sum_{j=1}^{V} x_j = 1$.
Verify that the multinomial probability

$$p_{mult}(\text{x};\phi) = B(x) \prod_{j=1}^{V} {\phi_j}^{x_j}, \text{ where } B(x) = \frac{\left(\sum_{j=1}^{V} x_j\right)!}{\prod_{j=1}^{V}(x_j)!},$$

is identical to the probability of the same document under a categorical distribution, $p_{cat}(\text{w};\phi)$.

4 Bigram model [15 points]

Consider the following corpus:
☐ *Concrete Mathematics: A Foundation for Computer Science, is first and foremost a most approachable book on mathematics: mathematics combinatorial and algorithmic, mathematics discrete and continuous* ■

☐ *There is a chapter on discrete probability, which is presented with an eye for the beginning computer science major* ■

☐ *The viewpoint taken is that the essence of concrete mathematics resides in the solution of useful problems* ■

□ *In about eleven pages, many elementary results on these numbers are succinctly and elegantly laid out, including the beautifully symmetric Cassini identity between any three successive Fibonacci numbers*■

Using on the above corpus, calculate the probability of the word sequence shown below, assuming the bigram (2-gram) language model. Do not include stop-words like (a, an, the, is, in, and) and punctuation (“,” , “:” , “.”) in your calculations.

The major essence of concrete mathematics resides in the interplay between the discrete and continuous and in the cross fertilization between combinatorial and algorithmic. Just like we humans learn various skills using different methods, machines learn various skills through different types of learning algorithms.