DATA.STAT.840 Statistical Methods for Text Data Analysis

Exercises for Lecture 1: Introduction

The answers to the exercises should be returned as a single PDF file with all the answers. Additionally, for any exercises that require coding, return the code files as separate files.

Exercise 1.1: Types of text data. List five types of text data that are **not already covered** by the list on slides 18-20 of the lecture. For each of the data types, suggest what a question that could be analyzed based on a collection of such data.

Exercise 1.2: Chatterbots. Perform a conversation with an online chatterbot, such as Mitsuku or another chatterbot, even up to current large language models. Try to make the bot reveal that it is a bot *without directly asking it (many current bots are directly programmed to reveal themselves if asked)*: for example, try to make the bot reply in a way that shows limits in its understanding. Report the name and web address of the chatterbot and your conversation with the bot in your answer.

Exercise 1.3: Chain rule of probabilities. Consider a sequence of N words $w_1,\ldots,w_N$, where the vocabulary index of each word is a random variable. Denote the words to the left of word i by $\mathbf{Left}_i=[w_1,\ldots,w_{i-1}]$ and the words to the right by $\mathbf{Right}_i=[w_{i+1},\ldots,w_N]$. The conditional probability of the word at position i, given the words to the left, is then $p(w_i|\mathbf{Left}_i)$. The conditional probability of the word, given the words to the right, is $p(w_i|\mathbf{Right}_i)$. If there are no words to the left/right these simply reduce to $p(w_i)$.

Use the chain rule of probabilities to prove mathematically that $\prod_{i=1}^{N}\frac{p(w_i|\mathbf{Left}_i)}{p(w_i|\mathbf{Right}_i)}=1$.

Exercise 1.4: Python basics, part 1.

(a) Install Python (for example the Anaconda installation).

(b) Write in Python a function that computes and prints the probability density function of a multivariate Gaussian distribution, at a set of multiple (one or more) desired evaluation locations. The function should take in the parameters of the Gaussian and the set of evaluation locations as arguments, in a suitable format of your choosing. You may use Python libraries such as math, numpy, and scipy, but do not use a ready-made function for the multivariate Gaussian probability density - implement it yourself.

(c) Using the function that you wrote, evaluate the probability density of a 3-dimensional multivariate Gaussian whose mean is $\boldsymbol{\mu}=[1,3,5]^T$ and covariance matrix is $\boldsymbol{\Sigma}=\begin{bmatrix}4&2&1\\2&5&2\\1&2&3\end{bmatrix}$, evaluated at the locations $\boldsymbol{x}_1=[2,2,2]^T$, $\boldsymbol{x}_2=[1,4,3]^T$, and $\boldsymbol{x}_3=[1,1,5]^T$. For example, the three features could describe the number of adjectives, werbs, and nouns in a particular document, or the number of responses, likes, and retweets of a Twitter post.

Report the code you wrote and the output of the function. If you do not wish to use Python, you can instead perform this exercise in another language of your choosing, such as R or Matlab.