Assignment 1, CS683 (NLP), Spring Semester 2020-21

[Total Points: 10 + 2 optional bonus points as a stretch]

[Turnaround Date: 17th February 2021, 11:59:00 PM]

Model creation: Take the following 10 webpages (the main text body only, including all the titles, section headers etc.), up till but excluding the "See also" section and all that follows till the end of the page, as well as excluding the "Contents" section.

(a) https://en.wikipedia.org/wiki/Sachin_Tendulkar
(b) https://en.wikipedia.org/wiki/Virat_Kohli
(c) https://en.wikipedia.org/wiki/Sourav_Ganguly
(d) https://en.wikipedia.org/wiki/Sunil_Gavaskar
(e) https://en.wikipedia.org/wiki/Kapil_Dev
(f) https://en.wikipedia.org/wiki/Anil_Kumble
(g) https://en.wikipedia.org/wiki/Dilip_Vengsarkar
(h) https://en.wikipedia.org/wiki/Mohammad_Azharuddin
(i) https://en.wikipedia.org/wiki/MS_Dhoni
(j) https://en.wikipedia.org/wiki/Rohit_Sharma

Create and store bigram and trigram models.

Reporting: Report the following. (a) Raw word-count. (b) Count of (i) bigrams and (ii) trigrams. (c) Choose a bigram. Report the probability values of each trigram that would be generated for the set of trigrams actually present in the above corpus, such that the first two words of the trigram is the chosen bigram. Repeat this experiment with one more bigram.

Execution testing: A script/command line call that would provide a bigram as an input and produce a trigram at output, generated from the given bigram with a probabilistic generation. Execute this command using a randomly chosen bigram a large number of times and report the distribution of the output trigrams. Repeat this with another randomly chosen bigram. Repeat this with any two randomly chosen unigrams (preferably with a randomly chosen adjective/adverb). The course instructor might execute a few custom queries too, by providing an input gram to this program. Also, a command to retrieve the entire list of bigrams and trigrams, and the conditional probabilities of generating each possible bi/tri gram from the corresponding uni/bi gram.

Stretch goal (2 bonus points): Consider scenarios where OOV words will be provided in the execution testing phase. Rebuild the model separately to smoothly accommodate such occurrences within the n-gram models. Repeat the experiments above.