**CS 5316 NLP**

**Assignment 3**

# Deadline: 16 March at 11.55 PM

**1. Language Modeling Using** nltk: Use Natural Language Toolkit (nltk) to learn languge models for two corpora. The first one is Brown corpus and for the second choose any appropriate text corpora (e.g., movies review, reuters, etc).

1. Create unigram probability distribution for both corpora. Use an appropriate smoothing technique (e.g., add-one smoothing, Kneser-Ney).
2. Compare the distributions for both corpora and identify 20 words with marked differences in the two corpora.
3. Create bigram probability distribution of words for your corpus. What are the 50 bigrams with the most *pointwise mutual information* compared to the unigrams in your corpus?
4. Write code that will create probability distributions for each Markov state of the previous *n* words. Use it to make a function called *generate text* that generates text from your corpus. Show examples for n-grams with *n*=1 through 5.

For part (d) a source code is in Q1(d) text file is available which will help you starting with Markov text generator. It’s basically a mapping from markov states to probability distribution of words that can occur next.

## 2. Sentiment Analysis

Download dataset from <https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences>this link. It has sentences from three domains: i) imdb ii) amazon iii) yelp

Apply preprocessing e.g., removing stop words, lemmatization, stemming. State overall vocabulary size before and after preprocessing.

Split data set into training, validation and testing in order to complete modeling. Present your results using confusion matrix and F1 score. Use naïve Bayes and Logistic Regression learning algorithm for this task.  **Use the multinomial model for document representation.**

## 3. Text generation with RNN (LSTM)

In this task you will be creating LSTM model for a given text file named data\_lstm.txt. Use keras to train the model. You can find all the keras package details from this <https://keras.io/>link. You can use *max length, step size, batch size, epochs* and *diversity* values freely. Save the model and weights in .yaml and .h5 format file respectively.

After creating and saving the model. You will write a method called *text\_generation()* which will randomly select the sentence of length *max length* from the data\_lstm.txt and generate a sentence of ‘n’ words using model you just learned. For example, if n=10 then model should generate a sentence of length 10 words, given a randomly selected sentence.

Helpful links regarding RNN (LSTM) are as follows:

* <https://blog.paperspace.com/recurrent-neural-networks-part-1-2>
* <https://colah.github.io/posts/2015-08-Understanding-LSTMs>

**\*Important Instructions:**

Kindly create 3 subfolders in the root folder and root folder name should be your roll number. Following is the example:

* ‘*roll\_number Assignment 3*’ (root folder) o Q1
  + - (a) and (b) folder
    - (c) folder
    - (d) folder
  + Q2
    - Place your coding files and results in this folder
  + Q3
    - Place your coding files, .yaml, .h5 files and results in this folder

You will submit results and findings in word or pdf file for each question and its parts (if there is any) along with coding files. For all the questions, state your findings as much as you can by performing experiments with different list of parameters for each question.