Assignment 1 - NLP

TA: Shivani Kumar (shivaniku@iiitd.ac.in)
Maximum Marks: 70
Due Date: 11:59 PM 11th-April-2021

Instructions:

- Please don't copy from the internet or any other student. Plagiarism will thoroughly be checked on all the submitted files. Refer: Academic Dishonesty Policy
- Submit the python file along with a report in one zip file. The report should contain:
  - The methodology followed
  - The assumptions made (if any)
  - The results obtained
  - The observations (if any)
- If any python code file is found to be in text/pdf file format, the assignment will not be graded or the graded assignment will be allotted 0.
- Allowed Programming language: Python
- Use classroom discussion for any doubt.
- The maximum points for this assignment are 70.
- Please find the dataset for the assignment here.
- The format of the zip file should be: Assignment1_rollNo.zip

Task 1: Preprocessing

[20 points]

Given the text file (grail.txt) from the Web Text Corpus of NLTK, perform the following tasks:

1. Report the number of sentences and tokens contained in the file given as input.
2. Convert the whole text to lower case and report the number of unique tokens present before and after lower casing in the input file.
3. Report the number of stopwords in the file. Report the number of tokens left after stopword removal.
4. Perform stemming after removing stopwords and report the number of unique tokens left in the text.
5. Report the number of words starting with a consonant and the number of words starting with a vowel in the file given after performing steps 1,2,3, and 4.
6. Given a word and a file as input, return the number of sentences starting with that word in the input file after performing steps 1,2,3, and 4.

7. Given a word and a file as input, return the number of sentences ending with that word in the input file after performing steps 1,2,3, and 4.
8. Given a word and a file as input, return the count of that word in the input file after performing steps 1,2,3, and 4.

Task 2: Hidden Markov Model

[50 points]

Implement an HMM-based approach to POS tagging. Specifically, you have to implement the Viterbi algorithm using a bigram tag/state model. For training, a POS-tagged section of the BERP corpus has been attached (Training set_HMM.txt). Your systems will be evaluated against an unseen test set drawn from the same corpus.

Training: The training data consists of around 15,000 POS-tagged sentences from the BERP corpus. The sentences are arranged as one word-tag pair per line with a blank line between sentences, words and tags are tab-separated. Contractions are split out into separate tokens with separate tags. An example is shown here:

I PRP
'd MD
like VB
french JJ
food NN

Assume that the tags and words that appear in the training data constitute all the tags that exist (no new tags or words will appear in testing).

Decoding: For decoding, your system will read in sentences from a file with the same format minus the Tags, ie, one word per line ending with a period and a blank line before the next sentence. As output, you should emit an appropriate tag for each word in the same format as the training data.