CS 563: Natural Language Processing
Assignment-1: Part-of-Speech Tagging

Deadline: 15 February 2024

- Markings will be based on the correctness and soundness of the outputs.
- Marks will be deducted in case of plagiarism.
- Proper indentation and appropriate comments (if necessary) are mandatory.
- Use of frameworks like scikit-learn etc is not allowed.
- *All benchmarks(accuracy etc), answers to questions and supporting examples should be added in a separate file with the name 'report'*.
- *All code needs to be submitted in '.py' format.* Even if you code it in '.IPYNB' format, download it in '.py' format and then submit
- You should zip all the required files and name the zip file as:
    - <roll_no>_assignment_<#>.zip, eg. 1501cs11_assignment_01.zip.
- Upload your assignment (the zip file) in the following link:
    - CS563-NLP-2024-Assignments

Problem Statement:

- The assignment targets to implement Hidden Markov Model (HMM) to perform Part-of-Speech (PoS) tagging task

Implementation:

HMM based Model:

- HMM Parameter Estimation
    - Input: Annotated tagged dataset
    - Output: HMM parameters
    - Procedure:
        - Step1: Find states.
        - Step2: Calculate Start probability (π).
        - Step3: Calculate transition probability (A)
        - Step4: Calculate emission probability (B)
- Features for HMM:
    - Train two HMM models based on:
        - First order markov assumption (Bigram) where current word PoS tag is based on the previous and current words

        - Second order markov assumption (Trigram) where current word PoS tag is based on the previous two words along with the current word

Testing:

- After calculating all these parameters use these parameters to tag the test input sequence using the Viterbi algorithm

Dataset:

- Dataset consists of sentences and each word is tagged with its corresponding PoS tag
- Brown dataset: Brown_train.txt
- Format of dataset:
    - Each line contains <Word/Tag> (word followed by '/' and tag)
    - Sentences are separated by a new line

Documents to submit:

- Model code
- Perform 5 fold cross-validation on the Training dataset and report both average & individual fold results (Accuracy, Precision, Recall and F-Score).
- Create a confusion matrix using Python Library.
- Briefly discuss Bigram vs Trigram assumption while training HMMs.
- With some examples (good pairs and bad pairs) why the model is confused and when it is giving correct results. Analyze and Explain the reason behind it.
- Also, Implement a RNN based model for this task and compare the result of both RNN and HMM.
- Discuss which model is better? With some justification and analysis when RNN is better than HMM and when HMM is better than RNN and when both fail and why?
- Write a report (doc or pdf format) on how you are solving the problems as well as all the results including model architecture (if any).

For any queries regarding this assignment, contact:

Ramakrishna Appicharla (ramakrishnaappicharla@gmail.com),
Arpan Phukan (arpanphukan@gmail.com),
Sandeep Kumar (sandeep.kumar82945@gmail.com), and,
Aizan Zafar (aizanzafar@gmail.com)