Natural Language Processing - Assignment 03

Deadline: 31/10/2021 11:59:59 PM **Maximum Marks: 50**

Instructions:

- The assignment is to be attempted in a group of max 2. You have to clearly mention the contribution of each member in README.pdf.
- Language allowed: Python
- You are allowed to use libraries such as NLTK for data preprocessing.
- For Plagiarism, institute policy will be followed.
- You need to submit README.pdf, Code files (it should include both .py files/.ipynb files), and Output.pdf.
- Mention methodology, preprocessing steps and assumptions you may have in README.pdf.
- Mention your sample outputs in the output.pdf.
- You are advised to prepare a well-documented code file.
- Submit code, readme and output files in ZIP format with the following name: A3_<roll_no1>_<roll_no2>.zip
- Use classroom discussion for any doubt.

Dataset: Download here

- Each sentence represents one sentence and is separated by a fullstop(.) at the end.
- Words in each sentence have the format **word / tag**.
  Note: In some cases, the word itself contains a slash (/), e.g., **a/b** / **tag**. In such cases, treat a/b as a single word unit.
- Sentences are already tokenized.

Task: Part-of-Speech Tagging

1. Design and implement Hidden Markov Model (HMM) based Part-of-Speech (POS) tagger implementing Viterbi algorithm with the following assumptions. **[20]**
   a. **Markov assumption length 1** - Probability of any state sk depends on its previous state only, i.e., $P(s_k \mid s_{k-1})$
   b. **[Optional] Markov assumption length 2** - Probability of any state sk depends on its previous two states only, i.e., $P(s_k \mid s_{k-2}\, s_{k-1})$
   Note: Only if your system permits. It is expected that it will increase the performance of the tagger.

2. Design and implement a MLP-based Part-of-Speech (POS) tagger implementing the following. **[15]**
   a. Create Word2Vec & Glove representations of each word. Compare their performances.
   b. Use the representation of each to perform the task of POS tagging using multi-layer perceptrons(MLP).
   Note: You need to think of a suitable architecture.

Experiments:

1. Perform 3-fold cross validation on the dataset performing both Task 1 and 2 and show a comparative study (tabular) on following basis: **[5]**
   a. Precision, recall and F1-score.
   b. Tag-wise precision, recall and F1-score
   c. Confusion matrix (Each element Aij of matrix A denotes the number of times tag i classified as tag j).
   d. Statistics of tag set.

2. Which word types are most frequently tagged incorrectly by HMM and MLP? Please provide a most likely justification for the same. **[10]**