IISER Bhopal

INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH BHOPAL

Course instructor: **Dr. Jasabanta Patro**
Assignment number: **1**
Course: **DSE 407: NLP**
Date: **March 11, 2025**
Marks: **40**
Date of submission: **March 18, 2025**

Regulations:

- Each student is required to submit solutions based on the specified task.
- Multiple submissions are not allowed.
- Plagiarism: Strictly prohibited. All work should be original. The code will be checked for plague (as well as AI detector) and appropriate action will taken if found guilty of copying.

Submission Guidelines:

- **Deliverables:** public URL of **(i) code** (Code-Mixed Text Classification) and **(ii) code** (Sequence Labeling Task) and **(iii) attached pdf** (Report of both Code-Mixed Text Classification and Sequence Labeling Task).
- File naming convention:

  rollno_name_nlpassignment1.ipynb

  rollno_name_nlpassignment1.pdf

- Students need to submit only the URL of the Colab notebook (with public access) with clear instructions for running the code. The runtime of the code should not be more than 10 minutes.
- Students are required to write a brief report, with a maximum length of 200 words, explaining why their method works with final F1 score. The report must be submitted in PDF format only.
- Deadline: All assignments must be submitted by the deadline. Late submissions will be penalized.

Marking:

- Marking will be done based on three criterias, (i) **code**, (ii) **model performance**, and (iii) **report**.
- The performance of each submission will be evaluated using F1-score of positive class based on the predicted labels and the gold ones.
- All submitted code should be reproducible with public access. If the results cannot be reproduced, the submission will be considered incomplete and it will not be marked.

Code-Mixed Text Classification:

In this assignment, you will work with three datasets: Hate, Humor, and Sarcasm. Each of these dataset contain two files: Train and Validation. Your task is to develop a classification model that can categorize each sample within these datasets into one of two categories:

- Hate dataset: Classify each sample as Hate or Non-Hate.
- Humor dataset: Classify each sample as Humor or Non-Humor.
- Sarcasm dataset: Classify each sample as Sarcasm or Non-Sarcasm.

Steps

1. **Dataset Processing:** Load the dataset containing labeled code-mixed text samples.
2. **Feature Extraction:** Use **N-gram language models** (unigrams, bigrams, trigrams) as features.
3. **Model Training:**
    - Implement **ML models** like Naive Bayes or SVM classifier.
    - Train the model on the extracted N-gram features.
4. **Model Evaluation:**
    - Test the model on the validation set only.
    - Use **F1 score of positive class** as evaluation metrics.
    - Provide the classification report as well.
5. **Analysis:**
    - Discuss the challenges in code-mixed classification and suggest improvements.

Files Provided

1. **Dataset:** https://github.com/islnlp/Assignment_1_2025

Deliverables

1. Python notebook implementing the classification task.
2. Detailed report including results and analysis.

Sequence Labeling Task:

Objective:
Your task is to implement the Viterbi algorithm to perform Part-of-Speech (POS) tagging using the given corpus. You will also handle noisy data by exploring multiple decoding paths and dynamically adjusting the emission probabilities. Use this colab notebook to get started: https://colab.research.google.com/drive/1z_5wShL-A2YhUrFK_mhA06r9s7676KgI?usp=sharing.

Steps:

1. **HMM Setup:** Use the given data provided to derive hidden states (POS tags) and observable states (words).
2. **Viterbi Implementation:** Implement the Viterbi algorithm to decode the most probable POS tags for each sentence.
3. **Noise Handling:** Implement strategies to handle noise in the noisy test data by exploring multiple decoding paths.
4. **Performance Evaluation:** Compare the accuracy of the baseline Viterbi algorithm vs the noise-handled version on the provided datasets.

Files provided:

1. **train_data.txt:** 2000 sentences with tagged words for training.
2. **test_data.txt:** 400 sentences with correct tags.
3. **noisy_test_data.txt:** 400 sentences with some noise introduced.

Deliverables:

1. Python code implementing the Viterbi algorithm.
2. Analysis should be mentioned in the report comparing the performance of the baseline vs noise-handling versions.

Constraints:
You can only use numpy and collections library.