IISER Bhopal

INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH BHOPAL

Course instructor: **Dr. Jasabanta Patro** Assignment number: 2
Course: **DSE 318/607: NLP** Date: **March 27, 2025**
Marks: **40** Date of submission: **April 4, 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** (Text Classification Using FFNN). and **(ii) code** (Text Classification with RNN, LSTM, Bi-LSTM, Transformer and BERT)and **(iii) report** (Submit the classification reports for all the architectures you have implemented, including RNN, LSTM, Transformer, and BERT).
  - The Colab notebook should only contain the inference part of the model and load the pre-trained weights. The training part should be commented out.
  - The model should be able to load weights from your public GitHub repository (create a repo with the trained model and download weights from it).
- File naming convention:

  rollno_name_nlpassignment2.ipynb
- 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.
- Deadline: All assignments must be submitted by the deadline. Late submissions will be penalized.

Marking:

- Marking will be done based on two criterias, (i) **code**, and (ii) **model performance** (more focus will be given to performance).
- The performance of each submission will be evaluated using average macro F1-score 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 the submission will not be marked.

Text Classification Using FFNN:

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.

Task overview:

1. **Embeddings:** Create a custom Word2Vec model from scratch. Train it on your dataset with an embedding size of 100 dimensions, and save the trained embeddings in your GitHub repository.
2. **Modeling:** Use Feedforward Neural Networks (FFNN) with a maximum size of 64 units.
3. **Training setup:**
    - Use Adam, AdamW or SGD optimizer.
4. **Note:** Comment out the training part (both embedding creation and model training) of the code (we can undo comments to check the training part also). The model should already be trained, and the deliverable will be focused on inference only.

Files Provided

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

Deliverables

1. Python notebook implementing the classification task.

Text Classification Using RNN, LSTM, Transformer and Pretrained Language Model:

Task overview:

1. **Base model:**
   - The training process is permitted to incorporate various neural network architectures. These include the use of word embeddings as well as use of Recurrent Neural Networks (RNN), Long Short-Term Memory networks (LSTM), Transformer and BERT are permitted to train.
   - A combination of these architectures can also be applied.
   - Usage of pre-processing techniques is optional.
2. **Note:** Comment out the training part of the code (we can undo comments to check the training part also). The model should already be trained, and the deliverable will be focused on inference only.

Deliverables

1. Python notebook implementing the classification task.
2. Submit the classification reports for all the architectures you have implemented, including RNN, LSTM, Transformer, and BERT. The reports should include relevant performance metrics such as Accuracy, Precision, Recall and macro F1-score.

References:

- Text classification using word embedding: NPTEL-tutorial.

Constraints	Value
BERT	google-bert/bert-base-uncased
Library	Pytorch

Table 1: Modeling constraints for part-I