

CS-681: Assignment #3

Due date: April 22, 2025 (11:59)

Objective:

The aim of this assignment is to give you hands-on experience in building recurrent neural network models for Natural Language Processing (NLP) tasks using different types of techniques and optimizing hyperparameters for improved model performance.

Instructions:

In this assignment, you will implement and evaluate three types of recurrent neural network architectures—Standard RNNs, LSTMs, and Bidirectional RNNs—on two important NLP tasks: machine translation and text generation. Your objective is to understand how these models differ in handling sequential data and to evaluate their relative performance on each task.

You will work with real-world datasets, apply appropriate preprocessing, design and train models, and compare their performance using standard metrics such as BLEU score, perplexity, and sample quality.

Tasks Overview:

You are required to:

1. **Preprocess and prepare datasets** for each task:
 - For machine translation, [use a parallel corpus](#) (e.g., English-Arabic, English- Urd, English- Indonesian).
 - For text generation, use a [Project Gutenberg: The Mysterious](#) (or any book from the [Project Gutenberg](#) website).
2. **Implement three models** for each task:
 - A standard RNN model
 - An LSTM-based model
 - A Bidirectional RNN/LSTM model
 - Integrate an **attention mechanism** into your LSTM model for machine translation.
 -
3. **Train all models** using consistent hyperparameters and training procedures. For text generation experiments with different temperature values.
4. **Evaluate the models:**
 - For translation: Use metrics such as **BLEU score** and **accuracy**.
 - For text generation: Evaluate with **perplexity** and include **sample outputs** for qualitative analysis.
5. **Compare and report:**
 - Analyze training and evaluation results.
 - Highlight the differences in performance between models.
 - Discuss the benefits and limitations of each architecture in different tasks.

Submission Requirement:

- A Jupyter Notebook or code script with clear documentation and comments.
- A report summarizing your approach evaluation result

Grading Breakdown (Total: 100 points)

1. Data Preprocessing (10 points)
 - Cleaning, tokenizing, and preparing input/output sequences for both tasks (5 pts)
 - Handling vocabulary, padding, and batching appropriately (5 pts)
2. Model Implementation (30 points)
 - Correct implementation of Standard RNN for both tasks (10 pts)
 - Correct implementation of LSTM for both tasks (10 pts)
 - Correct implementation of Bidirectional RNN or BiLSTM for both tasks (10 pts)
3. Model Training (15 points)
 - Training setup and consistent hyperparameters across models (5 pts)
 - Use of teacher forcing for translation and correct setup for text generation (5 pts)
 - Use of training/validation splits and monitoring loss (5 pts)
4. Evaluation (20 points)
 - Use of BLEU score for translation and perplexity for text generation (10 pts)
 - Quality and variety of generated text samples (5 pts)
 - Proper evaluation on unseen test/validation data (5 pts)
5. Report & Analysis (20 points)
 - Clear documentation of models, hyperparameters, and experiments (5 pts)
 - Comparative analysis of models with appropriate graphs/tables (10 pts)
 - Insightful conclusions based on quantitative and qualitative results (5 pts)
6. Code Organization & Reproducibility (5 points)
 - Clean, modular, and well-commented code
 - Ability to rerun experiments with minimal setup