

# Text Generation with RNN

## Objectives:

Implement a simple RNN for text generation to deepen your understanding of how recurrent neural networks can be used to model sequences and generate text based on learned patterns.

- RNN Model Implementation
  - Implement a basic Recurrent Neural Network model from scratch using PyTorch or TensorFlow. Your model should include an embedding layer, at least one RNN layer, and a fully connected layer for output. Refer to the "Recurrent Neural Networks (RNN)" section of the lectures for guidance on the architecture.
- Dataset Preparation
  - Select a small text dataset for training your model. This could be a collection of poems, song lyrics, or any text of your choice. Preprocess the data by tokenizing the text into sequences and converting them into numerical format suitable for training your RNN. Use web scrapping technique to get a raw text from <https://www.gutenberg.org/cache/epub/84/pg84-images.html> web page. Define **window\_size** variable (e.g 100).

### Data Sequences (Input):

- Length: 99 tokens (`window_size - 1`)
- Format: List of lists, where each inner list contains 99 numbers

Example format:

```
data = [  
    [token1, token2, ..., token99],  
    [token2, token3, ..., token100],  
    [token3, token4, ..., token101],  
    ...]
```

### Target Values:

- Length: 1 token per sequence
- Format: List of single numbers

Example format:

```
targets = [  
    token100,  
    token101,  
    token102,  
    ...]
```

### Sequence Requirements:

- Each input sequence should have exactly 99 tokens
- Each target should be a single token
- Input sequence and target should be consecutive in the original text

- Training
  - Train your RNN model on the prepared dataset. Aim to optimize the model to predict the next word in a sequence based on the given context. Adjust hyperparameters such as learning rate, number of epochs, and hidden layer dimensions to improve performance.
- Text Generation
  - Once trained, use your model to generate text. Start with a seed sentence or word, then predict the next word using your model. Append the predicted word to your text and use the updated sequence as the new input to generate the next word. Repeat this process to generate a text of at least 100 words.
- Analysis
  - Analyze the generated text. Discuss how well your model captures the style and coherence of the chosen dataset.