Experiemnt – 5

Objective:- The goal of this assignment is to implement a sequence-to-sequence (seq2seq) model for machine

translation from English to Spanish. You will explore two architectures:

1. LSTM Encoder-Decoder without Attention

2. LSTM Encoder-Decoder with Attention

    • Bahdanau (Additive) Attention

    • Luong (Multiplicative) Attention

Dataset:- Dataset consist of sets of English - Spanish pairs. Link

Hello. Hola.

How are you? ¿C´omo est´as?

I am fine. Estoy bien.

Theory:- LSTM Encoder-Decoder without Attention

Overview

The LSTM Encoder-Decoder architecture is commonly used for sequence-to-sequence tasks, such as language translation. It comprises two main components:

- **Encoder:** Reads and compresses the input sequence into a fixed-size context vector (also called the hidden state).
- **Decoder:** Uses the context vector to generate the target sequence, step-by-step.

Working

- The encoder processes the input sequence using LSTM units and returns the final hidden and cell states.
- These final states are passed to the decoder as the initial states.
- The decoder predicts each word/token in the output sequence based on its previous hidden state and the previously generated token.

Limitations

- Compressing all input information into a single vector can lead to performance degradation for long sequences.
- The model may forget earlier parts of the sequence, making it harder to translate or map longer inputs accurately.

LSTM Encoder-Decoder with Attention

Attention mechanisms address the limitation of using a fixed context vector. Instead of relying only on the last encoder state, the decoder **attends** to different parts of the input sequence at each step of output generation.

Types of Attention Mechanisms

A. Bahdanau (Additive) Attention

Introduced by Bahdanau et al. (2014), also called "Additive Attention".

Key Idea

At each decoding time step, the decoder can **learn to align** and focus on different parts of the input sequence.

B. Luong (Multiplicative) Attention

Proposed by Luong et al. (2015), this version is known as "Multiplicative Attention" and is more computationally efficient.

Comparison:-

Feature	LSTM without Attention	Bahdanau Attention	Luong Attention
Context Vector	Fixed (last encoder state)	Dynamic at each step	Dynamic at each step
Alignment Score Type	None	Additive (MLP)	Multiplicative (Dot/General)
Computation Complexity	Low	Higher	Lower
Sequence Length Handling	Poor for long sequences	Better	Better