

A3: Translate your own language

In this assignment, we will explore the domain of neural machine translation. The focus will be on translating between your native language and English. We will experiment with different types of attention mechanisms, including general attention, multiplicative attention, and additive attention, to evaluate their effectiveness in the translation process.

Note: You are ENCOURAGED to work with your friends, but DISCOURAGED to blindly copy other's work. Both parties will be given 0.

Note: Comments should be provided sufficiently so we know you understand. Failure to do so can raise suspicion of possible copying/plagiarism.

Note: You will be graded upon (1) documentation, (2) experiment, (3) implementation.

Note: This is a one-weeks assignment, but start early.

Deliverables: The GitHub link containing the jupyter notebook, a README.md of the github, and the folder of your web application called 'app'.

Task 1. Get Language Pair - Based on MT + Transformer.ipynb, modify the dataset as follows:

- 1) Find a dataset suitable for translation between your native language and English. Ensure to source this dataset from reputable public databases or repositories. It is imperative to give proper credit to the dataset source in your documentation. (1 points)
- 2) Describe in detail the process of preparing the dataset for use in your translation model. This includes steps like text normalization, tokenization, and word segmentation, particularly focusing on your native language's specific requirements. Specify the libraries or tools you will use for these tasks and give appropriate credit to the developers or organizations behind these tools. If your native language requires special handling in tokenization (e.g., for languages like Chinese, Thai, or Japanese), mention the libraries (like Jieba, PyThaiNLP, or Mecab) and the procedures used for word segmentation. (1 points)

Note: proper attribution for both the dataset and the tools used in its processing is essential for maintaining academic integrity.

Task 2. Experiment with Attention Mechanisms - Implement a sequence-to-sequence neural network for the translation task. Your implementation should include the following attention mechanisms, with their respective equations:

- 1) General Attention: (0.5 points)

$$e_i = s^T h_i \in \mathbb{R} \quad \text{where} \quad d_1 = d_2$$

- 2) Multiplicative Attention: (0.5 points)

$$e_i = s^T W h_i \in \mathbb{R} \quad \text{where} \quad \mathbf{W} \in \mathbb{R}^{d_2 \times d_1}$$

- 3) Additive Attention: (0.5 points)

$$e_i = v^t \tanh(\mathbf{W}_1 h_i + \mathbf{W}_2 s) \in \mathbb{R}$$

Note: For an in-depth exploration of attention mechanisms, you can refer to this paper¹.

¹An Attentive Survey of Attention Models <https://arxiv.org/pdf/1904.02874.pdf>

Task 3. Evaluation and Verification - Choose the most effective attention mechanism based on your experiments in Task 2. For the final evaluation and verification, perform the following:

- 1) Compare the performance of these attention mechanisms in terms of translation accuracy, computational efficiency, and other relevant metrics. (1 points)
- 2) Provide performance plots showing training and validation loss for each type of attention mechanism (General, Multiplicative, and Additive). These plots will help in visualizing and comparing the learning curves of different attention models. (0.5 points)
- 3) Display the attention maps generated by your model. Attention maps are crucial for understanding how the model focuses on different parts of the input sequence while generating the translation. This visualization will offer insights into the interpretability of your model. (1 points)
- 4) Analyze the results and discuss the effectiveness of the selected attention mechanism in translating between your native language and English. (1 points)

Task 4. Machine Translation - Web Application Development - Develop a simple web application that showcases the capabilities of your language model in machine translation. (3 points)

- 1) The application should feature an input box where users can enter a sentence or phrase in a source language.
- 2) Based on the input, the model should generate and display the translated version in a target language. For example, if the input is "Hello, how are you?" in English, the model might generate "Hola, ¿cómo estás?" in Spanish.
- 3) Provide documentation on how the web application interfaces with the language model for machine translation.

As always, the example Dash Project in the GitHub repository contains an example that you can follow (if you use the Dash framework).

Best of luck in developing your machine translation!