VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

Logo

Description automatically generated**FACULTY OF INFORMATION TECHNOLOGY**

**DEEP LEARNING   
FINAL PROJECT PART 2**

*Instructor:* **Assoc. Prof. PhD. Lê Anh Cường**

*Student:* **Trương Gia Bảo - 521H0201**

**Hồ Hữu An - 521H0489**

**Trần Nguyễn Duy Bảo - 521H0493**

*Class:* **21H50302**

**21H50301**

**HO CHI MINH CITY, 2024**

VIETNAM GENERAL CONFEDERATION OF LABOR

**TON DUC THANG UNIVERSITY**

Logo

Description automatically generated**FACULTY OF INFORMATION TECHNOLOGY**

**DEEP LEARNING   
FINAL PROJECT PART 2**

*Instructor:* **Assoc. Prof. PhD. Lê Anh Cường**

*Student:* **Trương Gia Bảo - 521H0201**

**Hồ Hữu An - 521H0489**

**Trần Nguyễn Duy Bảo - 521H0493**

*Class:* **21H50302**

**21H50301**

**HO CHI MINH CITY, 2024**

# ACKNOWLEDGEMENT

I would like to express my sincere thanks to the teachers and lecturers of Ton Duc Thang University and especially Mr. Le Anh Cuong - lecturer of Machine Learnig Group 05 for kindly helping and supporting as well as answering our questions throughout the process of making this report, so that I can get the best results.

**THE PROJECT WAS COMPLETED**

**AT TON DUC THANG UNIVERSITY**

I would like to assure you that this is my own project and guided by Le Anh Cuong. The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments, and evaluations collected by the author himself from different sources are clearly stated in the references section.

In addition, the project also uses some comments, reviews as well as figures of other authors and other organizations with quotes and annotations of origin.

If any fraud is detected, I would like to take full responsibility for the content of my project. Ton Duc Thang University is not involved in copyright or copyright violations caused by me in the process (if any).

*Ho Chi Minh City, 15th May 2024*

*Author*

|  |  |
| --- | --- |
| *Trương Gia Bảo* | *Hồ Hữu An* |
|  | *Trần Nguyễn Duy Bảo* |

**FACULTY ENDORSEMENTS AND REVIEWS**

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(ký và ghi họ tên)

# TABLE OF CONTENTS

[ACKNOWLEDGEMENT 1](#_Toc167040092)

[TABLE OF CONTENTS 1](#_Toc167040093)

[CHAPTER 1: INTRODUCTION 2](#_Toc167040094)

[CHAPTER 2: MODELS 3](#_Toc167040095)

[**2.1** Multihead Attention 3](#_Toc167040096)

[**2.2** Encoding base on Bert 3](#_Toc167040097)

[**2.3** Decoding base on PhoBert 4](#_Toc167040098)

[CHAPTER 3: EVALUATE 5](#_Toc167040099)

[REFERENCE 6](#_Toc167040100)

# CHAPTER 1: INTRODUCTION

**Machine translation model from English to Vietnamese**

A machine translation program that we have developed with the goal of converting documents from English to Vietnamese accurately and efficiently. The outstanding feature of this project is the use of an advanced Transformer model architecture, in which the encoding part is built based on BERT and the decoding part uses PhoBERT.



The model we have developed is an improved combination of BERT and PhoBERT, two famous models in the field of natural language processing. BERT (Bidirectional Encoder Representations from Transformers) is chosen as the encoding part to understand the context of words in English sentences. At the same time, PhoBERT, a refined version of BERT specifically for Vietnamese, is used as the decode part to create the corresponding Vietnamese translation sentence.

# CHAPTER 2: MODELS

### Multihead Attention

In multihead attention, the model uses multiple "heads" of attention to learn representations from many different "perspectives" of the input data. Each attention "head" consists of three phases: linear transformation to generate queries, key and value vectors for each head, attention mechanism to calculate attention weights, and linear combination of value values.



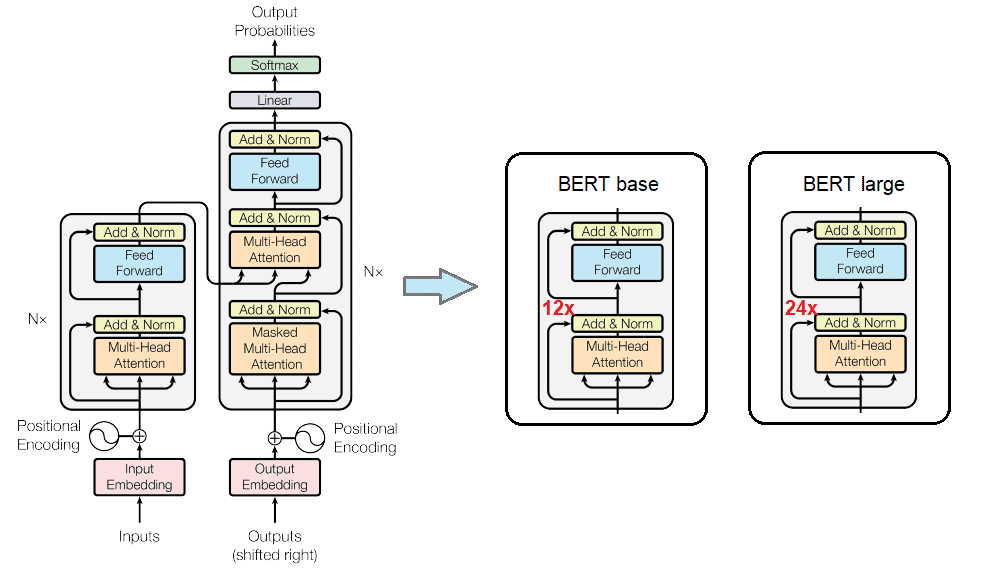
*WQi*​,*WKi*​,*WVi*​ are linear weight matrices that allow the model to learn transformations for the query, key, and value vectors in each attention head.

Each head of attention can focus on different parts of the input, allowing the model to interact with the data at multiple levels and create richer representation styles.

Multihead attention allows simulation of learning expressions from multiple "aspects" of input data, helping to improve the model's representation capabilities.

### Encoding base on Bert

BERT, stands for Bidirectional Encoder Representations from Transformers, is pre-trained using a combination of masked language modeling and next sentence prediction tasks on a large corpus that includes the Toronto Book Corpus and Wikipedia. Engineered to generate deep bidirectional representations from unlabeled text, it considers both left and right context across all layers.



**Architecture:**

***BERT****BASE***​: L**=12**, H=**768**, A**=12**,** Total Parameters=110M

***L****: the number of Transformer layers(blocks)*

***H****: size of the hidden layers*

***A****: the number of heads in the attention*

* **Pre-trained Masked Language Model (MLM)**: A random proportion of words in the input string are masked (replaced by token [MASK]), and the model tries to predict them based on the remaining words in the string .
* **Next Sentence Prediction (NSP)**: The model is given a pair of sentences and trained to predict whether the second sentence is a continuation of the first sentence.

### Tokenizer base on PhoBert

PhoBERT is a variant of BERT fine-tuned for the Vietnamese language. This model retains the structure and main features of BERT but is trained on a large amount of Vietnamese data to improve performance for many tasks in natural language processing.

**Architecture:**

***PhoBERT****BASE***​: L**=12**, H=**768**, A**=12**,** Total Parameters=110M

***L****: the number of Transformer layers(blocks)*

***H****: size of the hidden layers*

***A****: the number of heads in the attention*

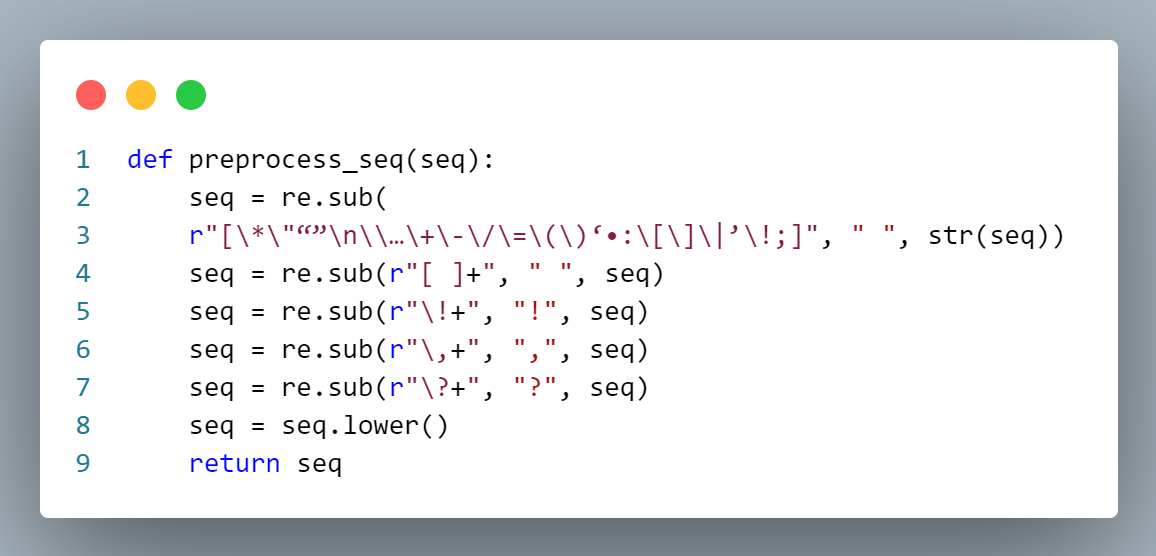
# CHAPTER 3: BUILDING DATASET

**The goal of building a dataset is to prepare data for the machine translation model. We need to convert the source and target text sentences into corresponding token and index strings to feed into the model.**

1. Data Format

Source and target data are stored in text files, each line containing a sentence. The data is read in and divided into two lists ***`source\_data`*** and ***`target\_data`***.

1. Data preprocessing



Before converting the data into a token string, we perform some preprocessing steps. In the code, we use the ***`preprocess\_seq`*** function to remove special characters and convert uppercase to lowercase

1. Convert data to token string

Use tokenizers from the Hugging Face library to convert sentences into token strings. In the code, using ***`AutoTokenizer`*** to create tokenizers from configuration parameters. We then use the ***`tokenizer.tokenize`*** method to split the sentence into a list of words and ***`tokenizer.convert\_tokens\_to\_ids`*** to convert the tokens into indices. The result is a list of token strings and a list of corresponding indices.

1. Build the data class

Defining the ***`TranslateDataset`*** class based on the ***`Dataset`*** class from the PyTorch library. In this class, we implement required methods like ***`\_\_len\_\_`*** and ***`\_\_getitem\_\_`*** to query the sample count and samples from the dataset. In the ***`\_\_getitem\_\_`*** method, we convert the data into tokens and prepare the input tensors for the model.

1. DataLoader and DataTrain

Finally, we use the ***`DataLoader`*** class to create batches from the dataset and use them to train the machine translation model. By batching, we can take advantage of parallel computing performance on GPUs.

# CHAPTER 4: MODEL TRAINING

1. Define the training function

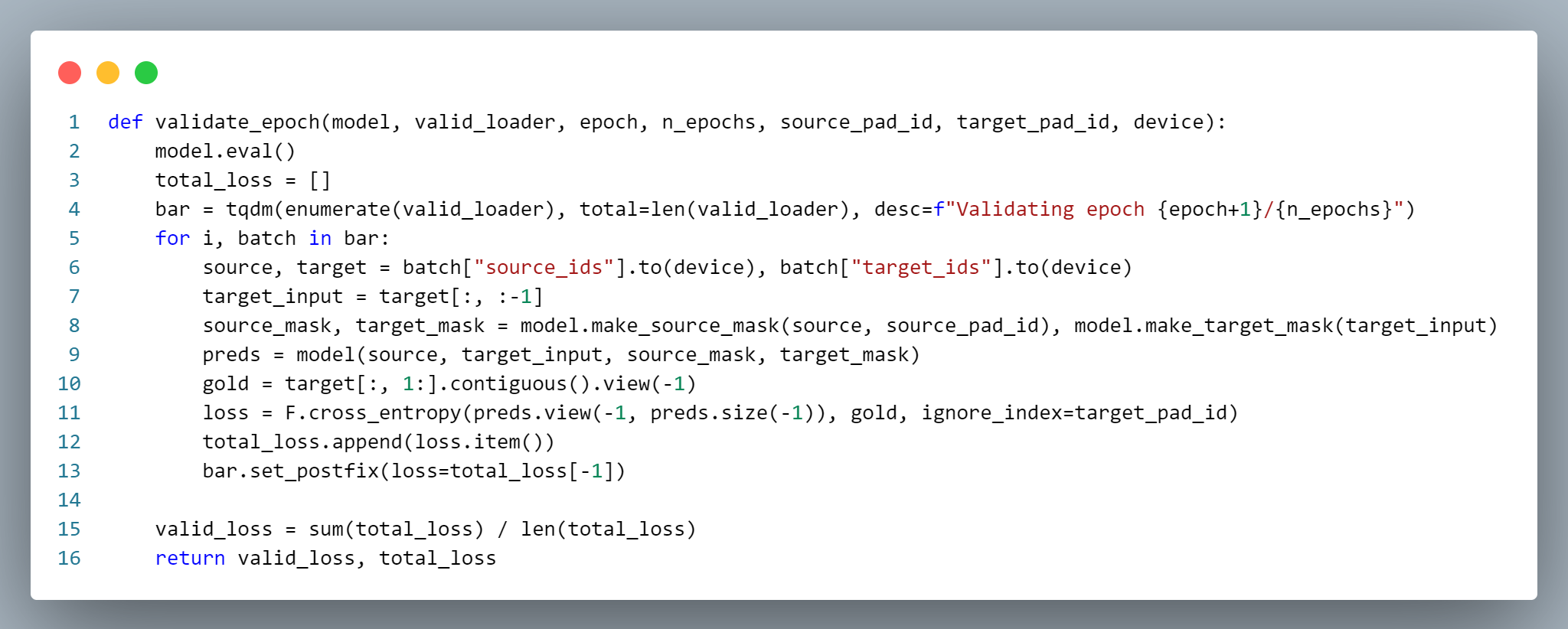
Define training functions to perform the model training process. The train\_epoch function will be called in each epoch to train the model on the training data.

During each epoch, we will iterate through each batch of training data. First, we will calculate the model's prediction output by passing the input through the encoder layer and decoder layer. We then calculate the error using the cross\_entropy function, comparing the predicted output with the actual output.

Uses the error to calculate the gradient of the loss function and updates the model's weights through Adam optimization. Adam is used to adjust the learning rate and update the weights effectively.

After training an epoch, we calculate the average error of all batches in that epoch. This average error will tell us the average error level of the model during training.

1. Evaluate model performance

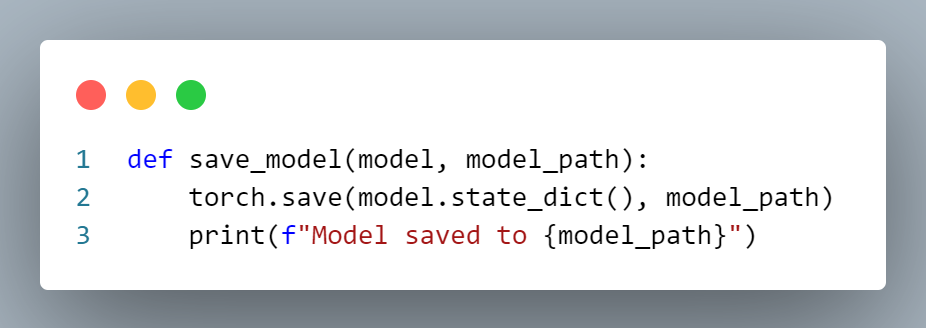


After each epoch, we will evaluate the model's performance on validation data. We use the ***“validate\_epoch”*** function to calculate the average error on the validation data.

The average error on the validation data will tell us the average error level of the model on data on which it has not been trained. This helps us evaluate the model's performance and determine whether it has learned correctly.

1. Save models and draw graphs

After training is complete, we store the model with the best performance to the specified path. This helps us retain the best model for later use.



We can also use the ***“save\_model”*** function to store the model after each epoch or after each weight update. This helps us track the training process and save intermediate versions of the model.

Finally, we can use the ***“plot\_loss”*** function to plot the training error and validation error from the training log. This chart helps us observe the change in error over time and evaluate the performance of the model.

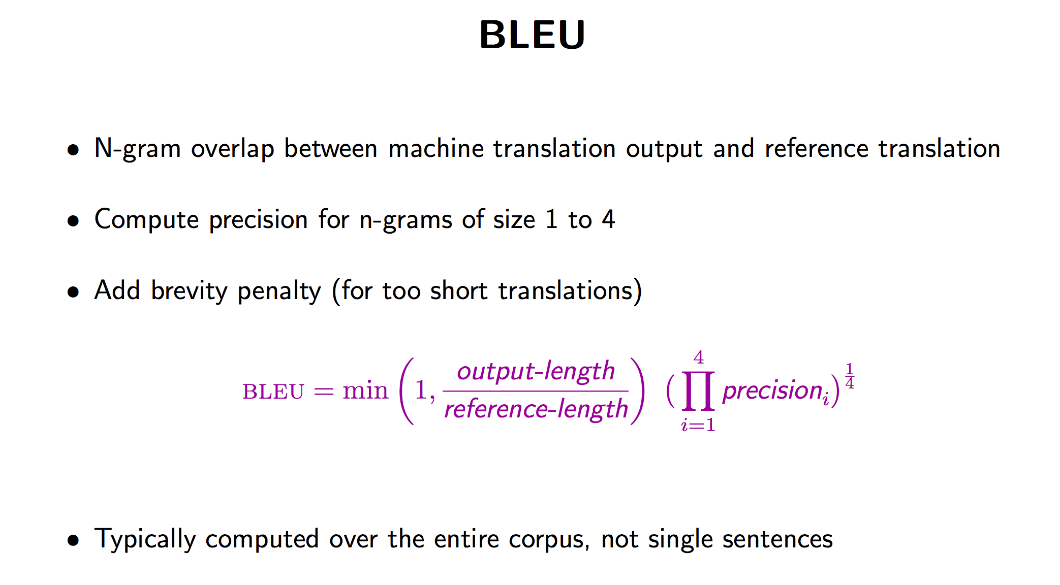
A graph of a loss

Description automatically generated

# CHAPTER 5: EVALUATE

BLEU (Bilingual Evaluation Study) is a method of evaluating the aggregate quality of machine translation versions. Published by Papineni et al. In 2002, BLEU used a comparison between word strings in the machine translator and word strings in a (human-generated) translation reference to calculate a total score.

Specifically, BLEU calculates the overlap ratio between word strings in the translation machine and word strings in the translation reference. It also measures the accuracy of positional words (n-grams) in machine translation relative to translation references. BLEU uses these measurements to calculate a total score, which typically ranges from 0 to 1.



# REFERENCE

[1] <https://viblo.asia/p/tim-hieu-ve-bleu-va-wer-metric-cho-1-so-tac-vu-trong-nlp-Eb85oA16Z2G#_a-n-gram-3>

[2] <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>

[3] <https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/>

[4] <https://phamdinhkhanh.github.io/2020/05/23/BERTModel.html>

[5] <https://www.linkedin.com/pulse/teaching-old-dog-new-tricks-difference-between-lewis-ms-ccrp-ches-3abqc/>

[6] <https://www.researchgate.net/figure/llustration-of-pre-training-Pre-training-is-a-part-of-transfer-learning-If-the_fig1_368698963>