Báo cáo bài tập lớn

Môn natural language processing 2025
Lớp: INT3406_3

Thành viên nhóm

Đào Mạnh Phú
23021660@vnu.edu.vn

Đoàn Khánh Nhật
23021652@vnu.edu.vn

Đặng Đức Minh
23021624@vnu.edu.vn

Tóm tắt

Nghiên cứu này được thực hiện để khảo sát thực nghiệm hai bài toán: Xây dựng mô hình dịch máy sử dụng Transformer cho ngôn ngữ Việt-Anh từ đầu và áp dụng cho bài toán phụ VLSP 2025 với miền dữ liệu y tế. Ở bài toán chính, chúng tôi xây dựng và tinh chỉnh kiến trúc Transformer từ đầu thông qua bốn chiến lược chủ đạo: tối ưu hóa tham số với weight tying, tăng cường biểu diễn ngữ nghĩa với Relative Positional Encoding, mở rộng chiều sâu với kiến trúc Deep Encoder (10 lớp) kết hợp Pre-Layer Normalization và cuối cùng là xây dựng mô hình tối ưu sử dụng tham số và kiến thức có được từ huấn luyện các mô hình tiền nhiệm. Bộ dữ liệu được sử dụng là PhoMT với ~3 triệu cặp câu. Kết quả được đánh giá dựa vào sự kết hợp giữa đo đạc truyền thống (SacreBLEU) và LLM-as-a-judge (Google Gemini) để mang lại cái nhìn đa chiều về chất lượng dịch thuật. Qua những cải tiến, từ mô hình cơ sở có điểm số BLEU là 9.11 -> mô hình cuối cùng là sự tổng hợp những kiến thức góp nhặt được có điểm số BLEU là 36.23. Từ những kinh nghiệm có được, chúng tôi tổng hợp và đánh giá những cải tiến về mặt cấu trúc đã thực hiện. Đối với bài toán phụ, thay vì sử dụng các mô hình ngôn ngữ lớn (LLM), chúng tôi lựa chọn tinh chỉnh và cải thiện mô hình tốt nhất có được trong bài toán chính, từ đó có được kết quả BLEU là 42.82. Sau đó, phân tích và đánh giá những ưu điểm của cách tiếp cận này cũng như các sai sót còn tồn tại.

I. Giới thiệu

Trong những năm vừa qua, nhờ vào sự phát triển mang tính đột phá của các mô hình học sâu (deep learning) và mô hình ngôn ngữ lớn (large language model), các phương pháp dịch máy truyền thống lỗi thời đã chuyển sang dịch máy neural tiên tiến hơn (RNN, LSTM). Tuy nhiên, các phương pháp này vẫn còn nhiều hạn chế, đặc biệt là trong các trường hợp câu có quan hệ phụ thuộc phức tạp và khó song song hóa (parallelize) quá trình huấn luyện.

Được ra mắt vào năm 2017, Transformer được cho ra mắt nhằm khắc phục các nhược điểm và thay thế các mô hình đi trước, mở ra thời kỳ mới cho bài toán dịch máy NLP (theo sau đó là các mô hình đào tạo trước như BERT và GPT).

Mục tiêu của đề tài này là xây dựng lại toàn bộ khối cấu trúc Transformer từ các thành phần cơ bản, bao gồm Encoder, Decoder, các khối Attention, … thay vì sử dụng thư viện có sẵn, sau đóáp dụng cho bài toán phụ. Mô hình sẽ được huấn luyện và đánh giá trên tập dữ liệu song ngữ phoMT nhằm kiểm chứng hiệu quả.

Thách thức của bài toán này nằm ở sự khác biệt về ngôn ngữ, khối lượng data nhỏ hơn so với các ngôn ngữ có số lượng người sử dụng lớn hơn như tiếng Trung, tiếng Pháp, tiếng Tây Ban Nha, … Do đó việc xây dựng kiến trúc Transformer lại từ đầu sẽ cho một cái nhìn tổng quát hơn, đặc biệt là trong bài toán dịch thuật Việt-Anh, những tham số nào quan trọng và những kỹ thuật nào giúp đạt được hiệu quả dịch tốt nhất.

II. Bài toán 1: Xây dựng Mô hình Dịch máy Seq2Seq với Kiến trúc Transformer

A. Tổng quan tài liệu

Kể từ khi ra đời vào năm 2017, kiến trúc Transformer đã xuất hiện nhiều biến thể khác nhau nhằm cải thiện sự ổn định và hiệu năng từ kiến trúc gốc.

- Regularization: xuất hiện vào năm 2019 với mục đích cho phép huấn luyện các mạng rất sâu bằng cách bỏ qua các lớp con một cách ngẫu nhiên.
- Positional encoding: được giới thiệu vào 2018 và cho rằng khoảng cách tương đối giữa các từ quan trọng hơn vị trí tuyệt đối giữa chúng:
- Normalization: ra đời muộn hơn vào 2020, được Ruibin Xiong và các đồng nghiệp chứng minh rằng Pre-Layer Normalization là yếu tố then chốt để huấn luyện các mạng Transformer sâu mà không cần giai đoạn warm-up phức tạp.

Đối với bài toán dịch máy Việt-Anh, các nghiên cứu trước đây thường tập trung vào việc thu thập dữ liệu hoặc áp dụng các mô hình đa ngôn ngữ. Ít có nghiên cứu nào đi sâu vào việc so sánh chi tiết các lựa chọn kiến trúc (như Post-LN vs Pre-LN, Absolute vs Relative PE) cụ thể cho cặp ngôn ngữ này khi huấn luyện từ đầu.

B. Cơ sở lý thuyết

Mô hình Transformer chuẩn bao gồm 2 khối Encoder và Decoder. Cả 2 khối đều được cấu tạo bởi các lớp (layer) xếp chồng lên nhau.

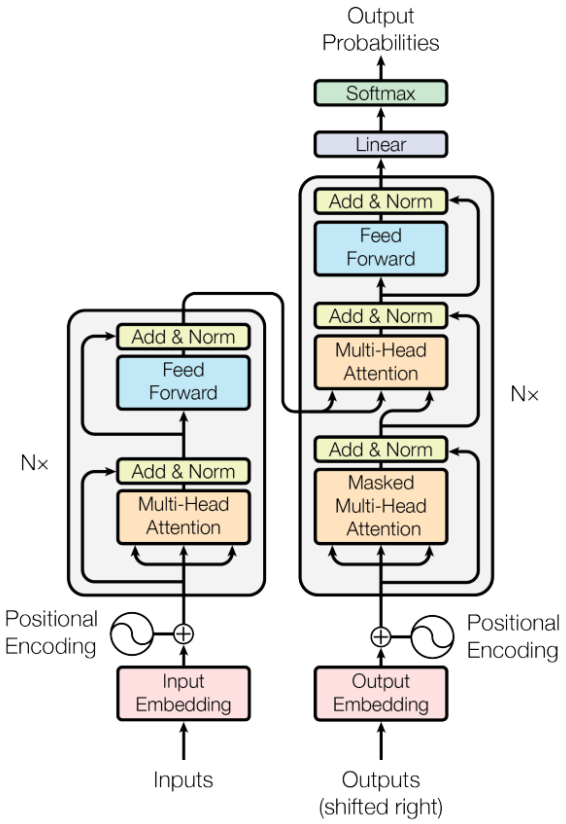
Mỗi Encoder layer bao gồm một module multi-head self-attention và môt mạng feed-forward theo vị trí (position-wise), trong khi mỗi decoder layer bao gồm thêm một module encoder-decoder attention nhằm khai thác thông tin biểu diễn của encoder.

Tất cả các sub-layer đều được kết hợp với residual connection, layer normalization và dropout để cải thiện khả năng hội tụ ổn định trong quá trình huấn luyện.

Do transformer không có cơ chế xử lý tuần tự tự nhiên, thông tin vị trí của token được bổ sung thông qua positional encoding.

Nhờ khả năng mô hình hóa các quan hệ phụ thuộc dài hạn và tính song song cao trong quá trình huấn luyện, Transformer đã trở thành kiến trúc nền tảng cho nhiều mô hình hiện đại trong xử lý ngôn ngữ tự nhiên.

Cấu trúc Transformer:



Hình 1: Cấu trúc Transformer cơ bản.

C. Phương pháp nghiên cứu

Nghiên cứu này sử dụng bộ dữ liệu song ngữ tiêu chuẩn PhoMT là một bộ dữ liệu lớn chất lượng cao dành cho mô hình ngôn ngữ Việt-Anh với 3.02 triệu cặp câu được cung cấp bởi VinAI vào năm 2021.

Dữ liệu bao gồm các cặp câu song ngữ Việt-Anh được chia thành ba tập: huấn luyện (training), kiểm định (validation) và kiểm tra (test).

Tập dữ liệu	Train	Validation	Test
Số lượng câu	2977999	18719	19151

D. Các bước tiền xử lý

1. Tokenization (Tách từ)

Token là nền tảng của tất cả các kỹ thuật NLP từ các phương pháp truyền thống cho đến các kỹ thuật học sâu nâng cao.

Với đặc điểm dữ liệu là ngôn ngữ, bước đầu tiên khi xử lý loại dữ liệu này là tokenize (tách từ) thành các đơn vị nhỏ hơn là token. Token là các khối xây dựng của NLP và tất cả các mô hình NLP đều xử lý văn bản thô ở cấp độ các Tokens. Chúng được sử dụng để tạo từ vựng trong một kho dữ liệu (tập dữ liệu trong NLP). Từ vựng này sau đó được chuyển thành số (ID) giúp xây dựng mô hình.

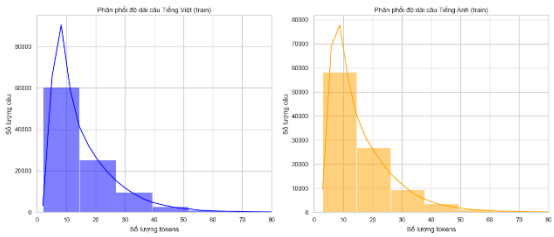
Tệp dữ liệu chứa 2 loại ngôn ngữ khác nhau, với sự khác biện về hình thái học giữa tiếng Việt (đơn âm tiết, từ ghép rời) và tiếng Anh (đa âm tiết, biến tố) nên 2 chiến lược xử lý riêng biệt sẽ được áp dụng.

1. Tiếng Việt (word-level): Sử dụng thư viện pyvi để thực hiện tách từ (word segmentation). Ví dụ cụm "Trí tuệ nhân tạo" sẽ được tách thành ['trí_tuệ', 'nhân_tạo'] thay vì 4 token rời rạc. Điều này giúp đảm bảo toàn vẹn ngữ nghĩa của từ ghép trong tiếng Việt.

2. Tiếng Anh (rule-based): Sử dụng bộ tách từ dựa trên quy tắc (Regex) để xử lý dấu câu và các hình thái từ co rút (contractions).

2. Lọc dữ liệu (Filtering)

Tệp dữ liệu chứa rất nhiều sequence (chuỗi) câu, trong đó, có những sequence quá ngắn hoặc quá dài và cần được loại bỏ do thiếu ngữ cảnh hoặc gây khó khăn cho mô hình, từ đó giảm hiệu quả trong quá trình huấn luyện.



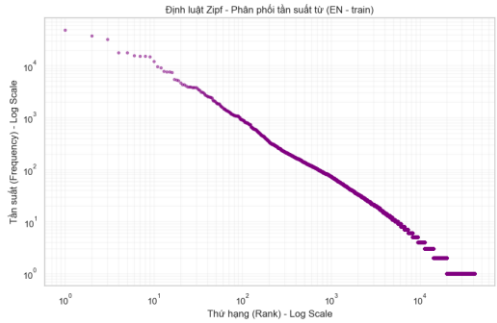
Hình 2: Phân phối độ dài câu trong 2 ngôn ngữ trong tập dữ li

Với độ dài tối thiểu là 3 tokens và độ dài tối đa 64 tokens, số lương mẫu huấn luyện giảm từ 2,977,999 xuống còn 2,953,209 cặp câu (~0.8%)

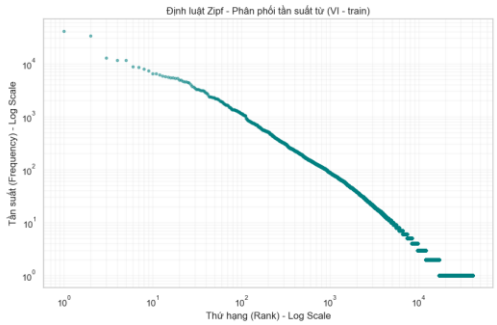
3. Xây dựng từ điển (Vocabulary Construction)

Từ điển là tập hợp tất cả các từ duy mà mô hình đã được học. Mô hình chỉ có thể hiểu và sinh ra các từ nằm trong tập hợp này.

Việc lựa chọn những từ nào được sử dụng để huấn luyện mô hình rất quan trọng. Nếu từ điển quá nhỏ, nhiều từ sẽ không thể được nhận diện và bị gán nhãn <unk> (unknown). Nếu từ điển quá lớn, kích thước mô hình sẽ tăng lên, làm chậm tốc độ huấn luyện và dự đoán, đồng thời rất dễ bị overfitting với các từ hiếm gặp.



Hình 3: Tần số xuất hiện của từ trong tiếng Anh trong tệp dữ liệu.



Hình 4: Tần số xuất hiện của từ trong tiếng Việt trong tệp dữ liệu.

4. DataLoader & Padding

Huấn luyện Deep learning dựa trên tính toán ma trận quy mô lớn. Do khối lượng mẫu rất nhiều nên để tăng tốc quá trình này, ta tận dụng khả năng tính toán song song của GPU.

GPU yêu cầu dữ liệu đầu vào phải là một ma trận. Tuy nhiên, giữa các câu sẽ có độ dài ngắn khác nhau. Để đảm bảo dữ liệu có thể được xử lý, ta tìm câu dài nhất trong lô (batch đó) và thêm các token place-holder vào <pad> khiến cho tất cả các câu có độ dài bằng nhau. Sau đó trong quá trình huấn luyện chỉ cần sử dụng masking để đảm bảo mô hình bỏ qua các token <pad>.

D. Số liệu thống kê và dữ liệu từ điển

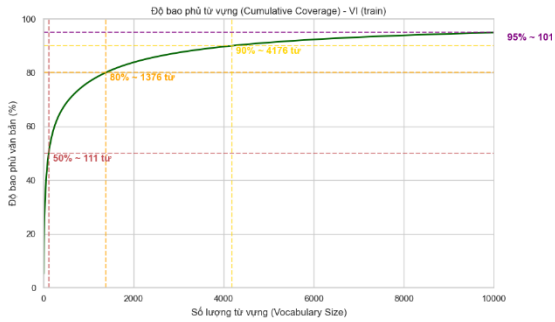
1. Số lượng mẫu

Tập dữ liệu	Số lượng (Raw)	Số lượng (filtered)
Train	2,977,999	2,953,299
Validation	18,719	18,719
Test	19,151	19,151

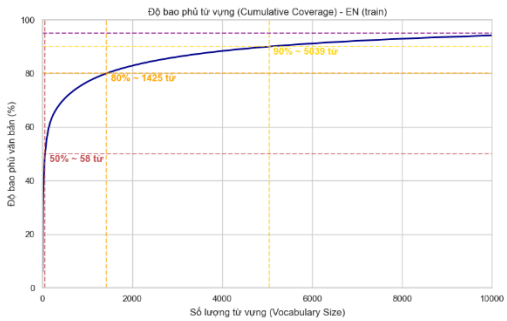
2. Đặc điểm từ điển

Với sample dữ liệu = 1,000,000 cặp từ trong dữ liệu gốc, chúng tôi lựa chọn 20,000 từ vựng để đại diện cho tập dữ liệu này.

Ngôn ngữ	Kích thước	Độ phủ
Tiếng Việt	20,004	97,63%
Tiếng Anh	20,004	96,61%



Hình 5: Độ bao phủ của từ vựng tiếng Việt trong tập dữ liệu



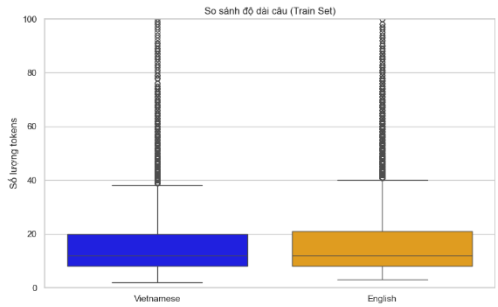
Hình 6: Độ bao phủ của từ vựng tiếng Anh trong tập dữ liệu.

Từ điển gồm 20,000 từ vựng và 4 token đặc biệt (<pad>, <sos>, <eos>, <unk>).

3. Đặc điểm độ dài câu.

Độ dài câu theo tập dữ liệu (train):

- Tiếng Việt: 15,84 token/ câu.
- Tiếng Anh: 16,15 token/ câu.



Hình 7: So sánh độ dài câu trung bình trong hai ngôn ngữ (đơn vị: token)

E. Chi tiết và kết quả các mô hình

Để tìm ra phương pháp tối ưu để cải thiện mô hình, nhóm đã tiếp cận bài toán theo các hướng khác nhau.

1. Mô hình thứ nhất – Baseline model

Mô hình cơ sở tuân theo kiến trúc Transformer tiêu chuẩn sử dụng cơ chế Post-Layer Normalization sẽ là điểm khởi đầu để phát triển các mô hình tiếp theo.

1.1. Self-attention và feed-forward

Self-attention là cơ chế cho phép mỗi token trong một chuỗi tự động học cách tập trung (attend) vào các token khác trong cùng chuỗi, giúp xây dựng biểu diễn ngữ cảnh phụ thuộc vào toàn bộ câu. Nói cách khác, mỗi từ không được biểu diễn độc lập, mà được mã hóa dựa trên mối quan hệ với các từ xung quanh.

Cách hoạt động (khái niệm):

Với mỗi token, mô hình tạo ra 3 vector:

+ Query (Q) – token "đặt câu hỏi"
+ Key (K) – token cung cấp thông tin
+ Value (V) – nội dung thông tin

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

-> Kết quả nhận được là một vector biểu diễn mới cho token, trong đó có kết hợp thông tin từ các token khác theo trọng số học được.

Cơ chế này giúp Transformer nắm bắt được các phụ thuộc dài hạn (long-range dependencies), cho phép tính toán song song (không như RNN).

Feed-forward Network (FFN) là một mạng neuron đầy đủ (fully connected network) được áp dụng độc lập và giống nhau cho từng token trong chuỗi.

Mỗi FFN gồm 2 lớp tuyến tính và một hàm kích hoạt ReLU phi tuyến tính ở giữa (các biến thể có thể dùng GELU).

$$\text{FFN}(x) = \max(0, xW_1 + b_1)\,W_2 + b_2$$

1.2. Post-LN Flow

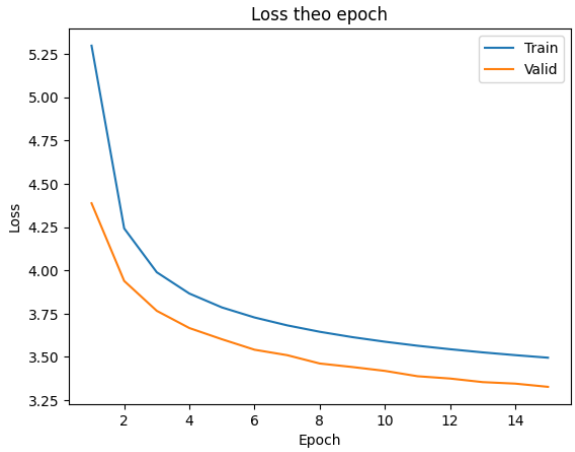
Post-Layer Normalization (Post-LN) flow là cách sắp xếp các phép toán trong một Transformer layer, trong đó Layer Normalization (LN) được áp dụng sau một phép cộng phần dư (residual connection).

$$x = LayerNorm(x + Sublayer(x))$$

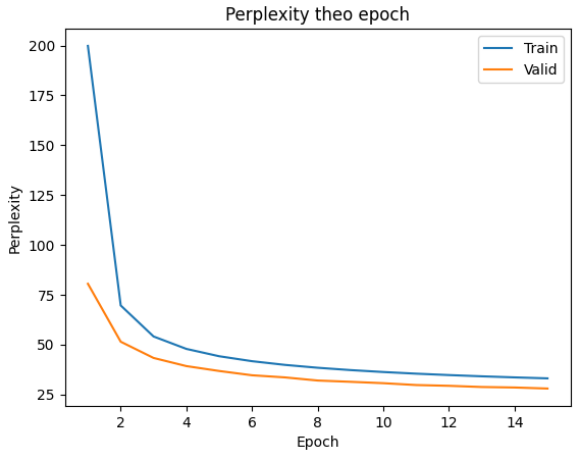
Đây là cấu trúc tiêu chuẩn nhưng thường gặp khó khăn về hội tụ khi mạng trở nên sâu hơn (deep

networks) do vấn đề vanishing gradient tại các lớp đầu.

Đồ thị Loss:



Đồ thị Perplexity:



Kết quả BLEU: 9.11
Kết quả Gemini score: 35.95

2. Mô hình thứ hai – Optimized baseline

Tối ưu mô hình chuẩn bằng cách tích hợp thêm Noam Scheduler và Label Smoothing.

2.1 Noam Scheduler

Là chiến lược điều chỉnh learning rate cho Transformer, kết hợp giai đoạn warmup và suy giảm theo căn bậc 2 của số bước huấn luyện giúp quá trình tối ưu ổn định và hiệu quả.

$$lr = d_{model}^{-0.5} \cdot \min(step^{-0.5}, step \cdot warmup\_steps^{-1.5})$$

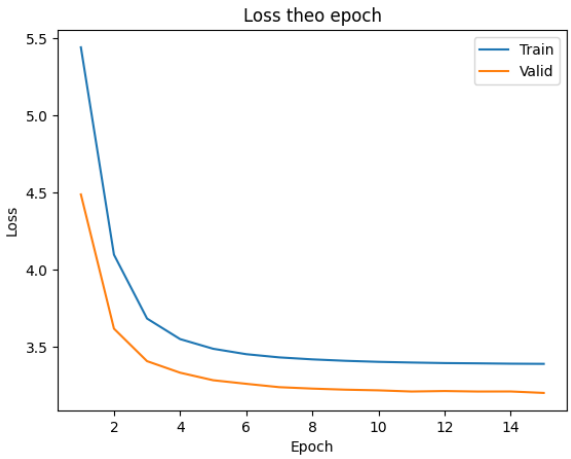
2.2. Label Smoothing

Là kỹ thuật regularization dùng trong huấn luyện mô hình phân loại, trong đó phân phối nhãn mục tiêu one-hot không còn là tuyệt đối mà được làm "mềm" bằng cách phân phối một phần xác suất nhỏ cho các lớp còn lại.

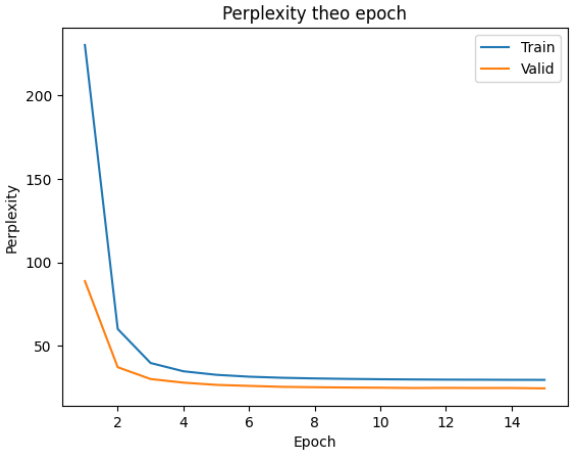
$$y_{ls} = (1 - \epsilon)y_{hot} + \frac{\epsilon}{K}$$

Kỹ thuật này giúp mô hình tránh dự đoán quá tự tin (over-confidence) và cải thiện khả năng tổng quát hóa.

Đồ thị Loss:



Đồ thị Perplexity:



Kết quả BLEU: 12.21
Kết quả Gemini score: 47.25

3. Mô hình thứ ba – GEGLUFeedForward, AMP, EMA.

Với việc sử dụng mô hình này thời gian huấn luyện giảm đáng kể và bộ nhớ được tiết kiệm mà không làm giảm quá đáng kể độ chính xác.

3.1. GEGLUFeedForward

Là mạng nơ ron được cải tiến thay vì dùng ReLU bằng GELU (Gaussian Error Linear Unit) là một hàm kích hoạt phi tuyến tính, trong đó đầu vào được điều chỉnh theo xác suất của một biến ngẫu nhiên Gaussian chuẩn, thay vì bị cắt ngưỡng cứng như ReLU.

$$\text{GELU}(x) = 0.5x\left(1 + \tanh\left(\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right)\right)$$

GELU cho phép luồng gradient âm nhỏ đi qua (thay vì triệt tiêu hoàn toàn như ReLU) giúp quá trình tối ưu mượt mà hơn, cải thiện hiệu năng trong bài toán dịch máy NLP.

3.2. AMP

AMP tự động sử dụng 2 kiểu dữ liệu khác nhau để đạt cân bằng giữa tốc độ và chính xác: FP16 với forward pass, tính toán trung gian và FP32 cho Optimizer, Loss, Gradient.

Từ đó giúp tăng tốc độ huấn luyện, tiết kiệm bộ nhớ mà vẫn giữ được chất lượng mô hình.

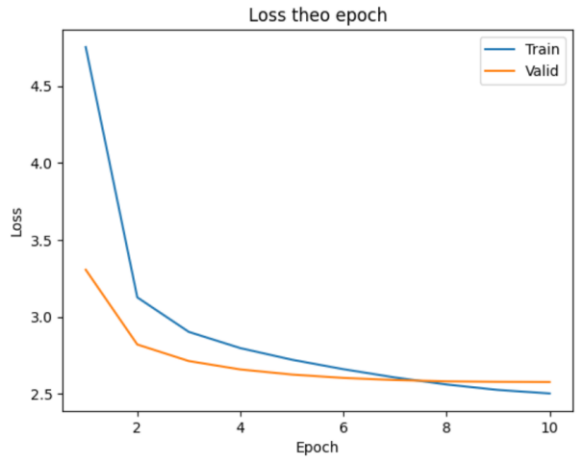
3.3. EMA

Là kĩ thuật làm mượt trọng số mô hình bằng cách lấy trung bình lũy thừa theo thời gian.

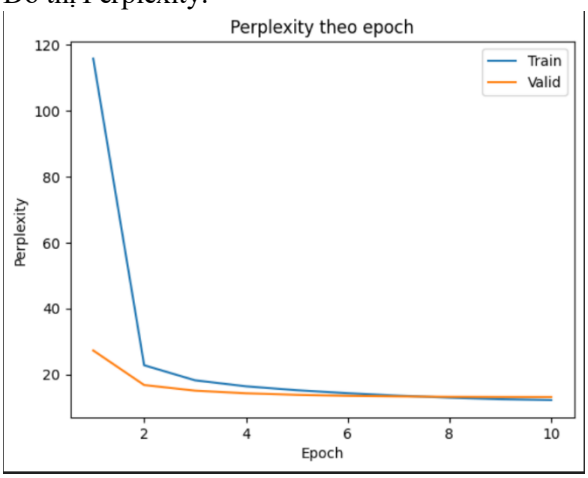
$$\theta_{ema} = \alpha \cdot \theta_{ema} + (1 - \alpha) \cdot \theta$$

Khi huấn luyện, weight thường dao động mạnh, EMA giúp làm mượt quá trình học, tăng generalization và kết quả eval ổn định hơn

Đồ thị Loss:



Đồ thị Perplexity:



Kết quả BLEU: 19.16

Kết quả Gemini score: 62.68

4. Mô hình thứ tư – Transformer enhanced sử dụng Relative positional encoding, LayerDrop, Head dropout

Nhằm cải thiện khả năng tổng quát hóa, mô hình này tích hợp các kỹ thuật can thiệp sâu vào cơ chế Attention.

4.1. Relative positional encoding

Relative positional encoding (RPE) là một phương pháp mã hóa vị trí trong Transformer, trong đó mô hình không sử dụng vị trí tuyệt đối của token, mà biểu diễn mối quan hệ tương đối về khoảng cách giữa các token trong chuỗi.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T + B_{rel}}{\sqrt{d_k}}\right)V$$

Việc này giúp tổng quát hóa các chuỗi dài lớn hơn, tăng khả năng nắm bắt phụ thuộc dài hạn, hiệu quả hơn Absolute PE trong bài toán NLP vì các từ ngữ trong một câu thường có mối liên hệ với nhau.

4.2. LayerDrop và Head dropout

LayerDrop là một kỹ thuật regularization theo cấu trúc, trong đó toàn bộ một Transformer layer có thể bị bỏ qua ngẫu nhiên trong quá trình huấn luyện.

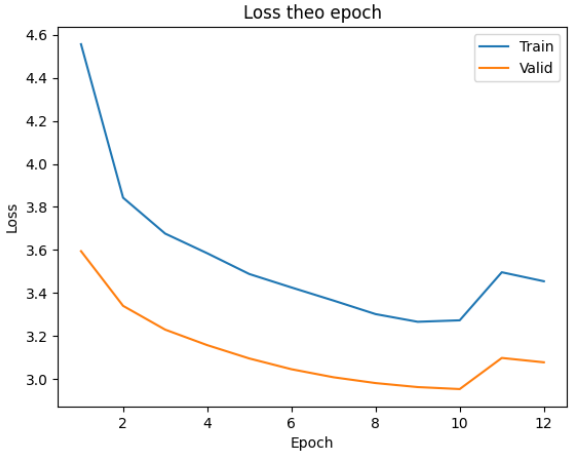
Cụ thể với xác suất p, một layer l không được áp dụng và đầu vào của nó được truyền trực tiếp qua residual connection.

x_{l+1} = Layer_l{x_l} (với xác suất 1-p)
x_l (với xác suất p)

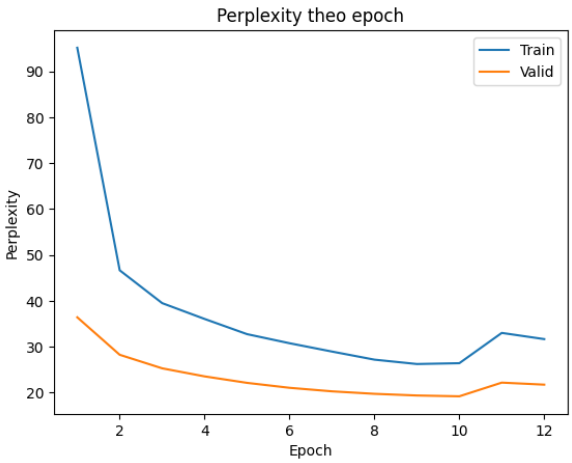
Sử dụng LayerDrop giúp mô hình học cách không phụ thuộc quá mức vào một layer cụ thể

Head dropout là một kỹ thuật regularization, trong multi-head attention, trong đó toàn bộ một hoặc nhiều attention head bị bỏ qua ngẫu nhiên trong quá trình huấn luyện.

Đồ thị Loss:



Đồ thị Perplexity:



Kết quả BLEU: 15.55
Kết quả Gemini score: 36.23

5. Mô hình thứ năm – Weight tying kết hợp Scheduled sampling và Noam LR & Cosine decay

5.1. Scheduled sampling

Scheduled sampling là một kỹ thuật huấn luyện cho các mô hình sinh chuỗi (seq2seq), trong đó đầu vào của decoder tại mỗi bước thời gian được chọn ngẫu nhiên giữa: Token đúng (ground-truth) và token do mô hình dự đoán. Xác suất lựa chọn được thay đổi theo thời gian huấn luyện.

Việc áp dụng kỹ thuật này sẽ làm giảm exposure bias (teacher forcing), cải thiện chất lượng sinh chuỗi dài.

5.2. Noam LR & Cosine decay

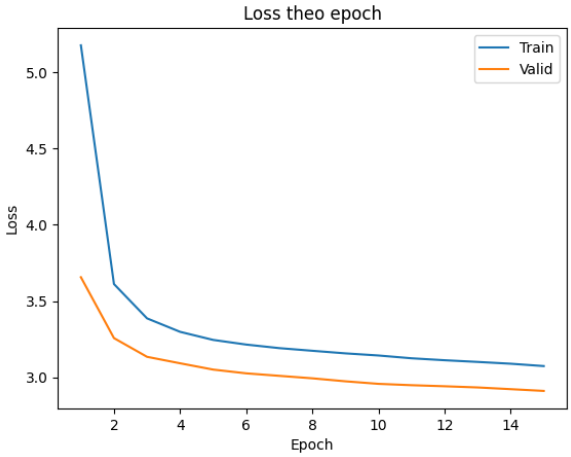
Noam LR (đã được giải thích ở phần 2.1)

Cosine decay là một chiến lược điều chỉnh learning rate giảm dần theo đường cong cosine, từ giá trị ban đầu về gần 0 trong suốt quá trình huấn luyện, giúp ổn định quá trình huấn luyện.

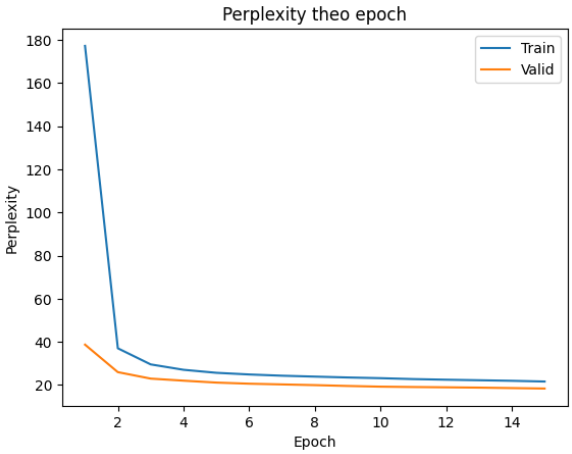
5.3. Label Smoothing

Giúp mô hình không qúa tự tin vào các nhãn cứng (one-hot), từ đó tăng khả năng tổng quát hóa, giảm overfitting.

Đồ thị Loss:



Đồ thị Perplexity:



Kết quả BLEU: 16.39
Kết quả Gemini score: 0.9 (bert F1)

6. Mô hình thứ sáu – Deep network

Đây là mô hình đạt hiệu năng cao trong thực nghiệm, chuyển từ cấu cầu hình nông (3 layers) sang sâu (10 layers encoder) được thiết kế để huấn luyện mạng sâu (deep network) một cách ổn định.

6.1. Pre-layer Normalization

Khác với baseline, mô hình này sử dụng Pre-LN để đặt chuẩn hóa trước khi vào sublayer:

$$x_{out} = x_{in} + \text{Sublayer}\big(\text{Norm}(x_{in})\big)$$

Điều này tạo ra một 'đường tắt' (identify path) cho gradient lan truyền ngược từ đầu ra về đầu vào, cho phép huấn luyện mạng sâu mà không cần warmup phức tạp.

6.2. RMSNorm (Root mean square normalization)

RMSNorm là một biến thể của Layer Normalization, được đề xuất nhằm giảm chi phí tính toán trong mô hình mạng neuron sâu, đặc biệt là kiến trúc Transformer quy mô lớn. Khác với

Layer Normalization truyền thống, RMSNorm loại bỏ bước chuẩn hóa theo trung bình (mean) và chỉ thực hiện chuẩn hóa dựa trên giá trị căn bậc hai trung bình (root mean square) của các phần tử trong vector đầu vào.

Cụ thể với đầu vào x, RMSNorm được định nghĩa như sau:

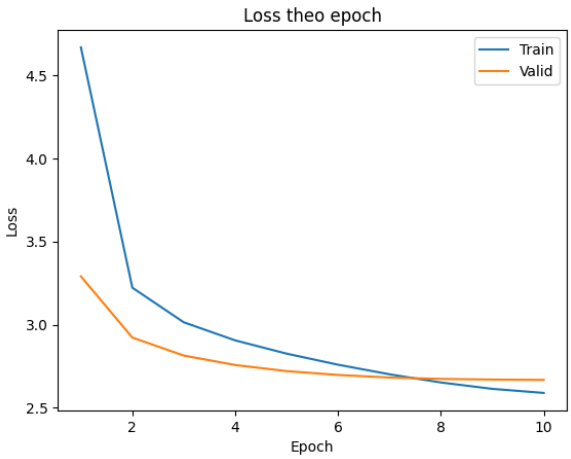
$$\text{RMSNorm}(x) = \frac{x}{\sqrt{\text{Mean}(x^2)+\epsilon}} \cdot \gamma$$

6.3. GEGLU Feed-forward

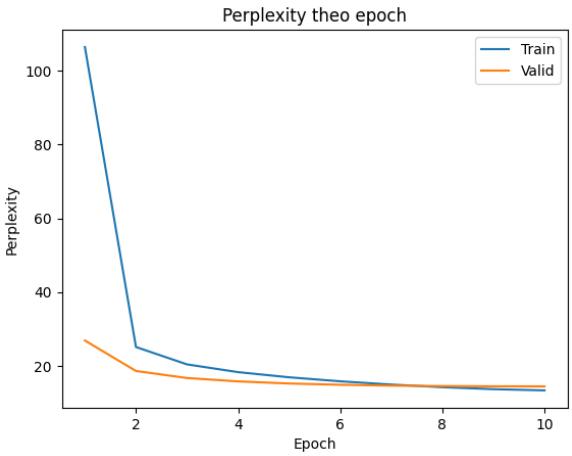
Mạng FFN sử dụng biến thể Gated Linear Unit

$$\text{GEGLU}(x, W, V) = (xW + b) \otimes \text{GELU}(xV + c)$$

Đồ thị Loss:



Đồ thị Perplexity:



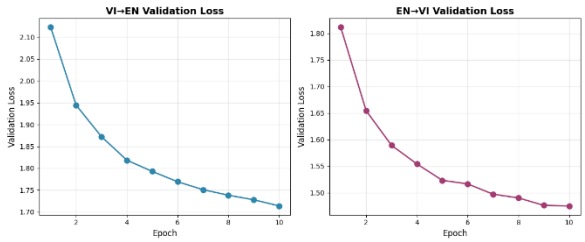
Kết quả BLEU: 25.61
Kết quả Gemini score: 56.54

7. Mô hình thứ bảy – Unrestrained Transformer

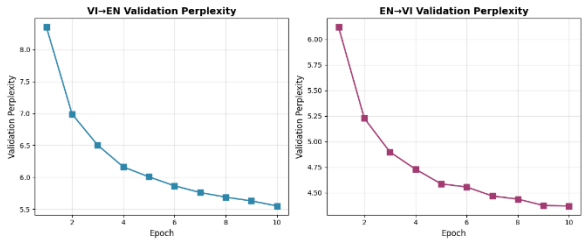
Ở các mô hình trước, với tập dữ liệu gồm ~ 3 triệu cặp câu, để đảm bảo thời gian huấn luyện nằm trong mức thực tế với mục đích chính là xác định mức độ ảnh hưởng của các cải thiện, mô hình chỉ được huấn luyện với 1-1,5 triệu cặp câu, trong đó bao gồm 20,000 token từ vựng. Ở phương pháp cuối cùng này tất cả data sẽ được sử dụng với 32,000 từ vựng. Kết hợp với đó là mô hình dịch Anh-Việt và Việt-Anh sẽ được huấn luyện tách biệt để đạt được kết quả dịch thuật tốt nhất

Do lượng parameters khổng lồ, mô hình chỉ áp dụng Pre-LN cùng với một số tối ưu hóa nhằm tăng hiệu năng (Bucket sampling & dynamic padding). Kết hợp với tinh chỉnh tham số (hyperparameters) từ những kiến thức có được trong quá trình huấn luyện các mô hình trước.

Đồ thị Loss trên 2 mô hình:



Đồ thị Perplexity trên 2 mô hình:



Kết quả BLEU: 36.23 (trung bình của 2 mô hình)
Kết quả Gemini: 59.40 (trung bình của 2 mô hình)

F. Đánh giá kết quả

Đánh giá các phương pháp sử dụng:

- Weight tying: Hoạt động theo cơ chế chia sẻ trọng số Embedding/ Output giúp giảm tới ~30% tham số sử dụng, tiết kiệm nhiều bộ nhớ. Tuy nhiên khả năng biểu diễn giảm đáng kể nếu vocab quá khác biệt. Nên sử dụng nếu có ít tài nguyên, đặc biệt là bộ nhớ.
- Advanced training: Bao gồm cơ chế scheduled sampling & Noam LR. Giảm exposure bias, tăng độ ổn định hội tụ. Mặc dù khá phức tạp để tích hợp nhưng cho hiệu quả tốt, thể hiện sự khác biệt nhiều nhất.
- Relative PE: Mã hóa vị trí tương đối giúp việc xử lý các câu dài được hiệu quả hơn. Bù lại chi phí tính toán tăng đáng kể. Nhưng mức độ cải thiện cao (đặc biệt là ngữ nghĩa)
- Deep architecture: Việc sử dụng nhiều encoder layer hơn tích hợp với Pre-LN giúp tăng khả năng trừu tượng hóa của mô hình nhưng cũng tăng tài nguyên sử dụng cũng như thời gian huấn luyện lên

nhiều. Có hiệu quả rất cao, chất lượng vượt bậc so

ID	Mô hình	Chiến lược nổi bật	BLEU	Gemini Score
1	Baseline	Post-LN Standard	9.11	35.95
2	Optimized Baseline	Noam Schedule + Label Smoothing	12.21	47.25
3	Compact	EMA + AMP + GELU	19.16	62.68
4	Enhanced	Relative PE + LayerDrop	15.55	36.23
5	Advanced	Scheduled sampling + Weight Tying	16.39	46.85
6	Deep network	Deep layer + Pre-LN + GEGLU	25.61	56.54
7	Unrestrained	Full Data + Pre-LN + Bucket Sampling	36.23	59.40

với sử dụng 3 layers như các mô hình khác.
Tổng hợp kết quả sau khi áp dụng các cải tiến:

Kết luận:
- Độ sâu là yếu tố quyết định: Việc mở rộng Encoder lên 10 lớp kết hợp với Pre-LN mang lại hiêu năng dịch vượt trội, khẳng định tầm quan trọng của khả năng biểu diễn ngữ nghĩa sâu.
- Relative PE hiệu quả hơn Absolute PE: Đặc biệt với các ngôn ngữ có cấu trúc ngữ pháp linh hoạt, biểu diễn vị trí tương đối giúp mô hình dễ hội tụ hơn.
- Chiến lược cho tài nguyên hạn chế: Nếu không thể chạy mô hình sâu, weight tying là lựa chọn tối ưu, giảm 30% tham số mà hiệu năng gần như không thay đổi.
- Sử dụng toàn bộ tệp dữ liệu: Việc không sample dữ liệu khiến cho thời gian huấn luyện tăng lên rất nhiều ~78% (từ ~45 phút/ epoch lên tới ~80 phút/ epoch) trong khi điểm số BLEU chỉ tăng ~44%, điểm số Gemini tăng ~3%. Điều này cho thấy mặc dù khả năng dịch thuật của mô hình được cải thiện rất nhiều, song không tương xứng với hạn chế nó mang lại. Việc tăng sample dữ liệu sử dụng cho huấn luyện là một phương pháp nhanh và đơn giản để cải thiện mô hình, tuy nhiên nên thử nghiệm nhiều để cân bằng giữa điểm số và thời gian cũng như tài nguyên sử dụng.

III. Bài toán 2: Áp dụng cho Bài toán phụ/VLSP Shared Task Machine Translation

A. Tổng quan

Mục tiêu của bài toán này là xây dựng hệ thống dịch máy chất lượng cao cho lĩnh vực Y tế, một lĩnh vực đặc thù với độ phức tạp cao về thuật ngữ chuyên môn và cấu trúc ngữ pháp.

B. Cách tiếp cận

Do giới hạn về tài nguyên nên việc lựa chọn fine-tuning các mô hình ngôn ngữ lớn (LLM) đa năng (như Qwen) vốn cồng kềnh và khó tối ưu triệt để sẽ không được ưu tiên. Thay vào đó, chúng tôi lựa chọn phương án Model-centric optimization: Xây dựng một kiến trúc Transformer chuyên biệt dành riêng cho bài toán này (Specialized Transformer) được huấn luyện từ đầu.

Phương pháp này cho phép:

- Tối ưu hiệu năng/ tài nguyên: Đạt tốc độ inference nhanh và chi phí tài nguyên cho training thấp hơn nhiều so với LLM.
- Khả năng tùy biến sâu: Can thiệp vào từng block của kiến trúc (Pre-LN, GEGLU) để tăng độ ổn định.

C. Các bước chuẩn bị và xử lý dữ liệu

1. Thống kê dữ liệu

Sử dụng bộ dữ liệu MedicalDataset_VLSP được cung cấp để sử dụng cho bài toán này, data sẽ được chia theo tỷ lệ 9:1 để đảm bảo mô hình được đánh giá khách quan trong quá trình huấn luyện.

Phân tập (Split)	Số lượng câu	Vai trò
Train	450,000	Huấn luyện mô hình
Validation	50,000	Tinh chỉnh tham số & early stopping
Public Test	3,000	Đánh giá hiệu năng thực tế

2. Kỹ thuật Tokenization nâng cao

Do đặc thù của bộ dữ liệu cho lĩnh vực y tế chứa rất nhiều từ vựng hiếm (tên thuốc, hợp chất hóa học) và từ vay mượn (tiếng Latin/ Anh). Nên các bộ tách từ thông thường được sử dụng cho bài toán dịch máy sẽ dễ gặp lỗi OOV (Out-of-Vocabulary) rất lớn. Do đó chúng tôi lựa chọn SentencePiece Model (SPM) với thuật toán Unigram:

- Vocabulary size: 24,000 (cho mỗi ngôn ngữ). Đây là kích thước cân bằng, đủ để bao phủ các từ

thông dụng nhưng không quá lớn làm loãng embedding matrix.

- Byte-fallback: Tính năng này có chức năng phân rã các từ chưa từng gặp (unknown words) thành các byte ký tự thay vì gán nhãn <unk>. Điều này cực kỳ quan trọng để bảo toàn thông tin cho các tên riêng hoặc mã số thuốc chưa từng xuất hiện trong tập train.

- Character coverage: 99,95% (cho tiếng Việt) để loại bỏ nhiễu lạ hoặc ký tự rác.

D. Kiến trúc mô hình

Mô hình được sử dụng là một biến thể của Transformer hiện đại, được đặc biệt chỉnh sửa để phù hợp với đặc thù của bài toán, khắc phục các điểm yếu về hội tụ của kiến trúc gốc.

1. Pre-Layer Normalization (Pre-LN)

Khác với kiến trúc Post-LN gốc, lớp chuẩn hóa (LayerNorm) được đặt trước khi vào các khối Attention và FFN.

Cơ chế: $x = x + Sublayer(Norm(x))$

Việc này tạo ra luồng gradient "sạch" (identity path) lan truyền ngược từ đầu ra về đầu vào, giúp loại bỏ nhu cầu phải có giai đoạn Warmup Learning Rate phức tạp và tăng tính ổn định khi huấn luyện mạng sâu.

2. GEGLU Feed-forward Network

Thay thế hàm kích hoạt ReLU/ GELU tiêu chuẩn bằng Gated Linear Unit (GEGLU) trong các khối Feed-Forward. Cơ chế cổng (gating) của GEGLU giúp mô hình chọn lọc thông tin ngữ nghĩa hiệu quả hơn, đặc biệt đối với các cấu trúc câu y khoa phức tạp.

3. Weight Tying

Để giảm thiểu số lượng tham số và tránh overfitting trên tập dữ liệu nhỏ Weight Tying được áp dụng để chia sẻ trọng số giữa ma trận embedding đầu vào của Decoder và lớp projection đầu ra. Điều này đóng vai trò như một kỹ thuật điều chuẩn (regularization) tự nhiên.

E. Thiết lập thực nghiệm

Mô hình được huấn luyên trên Nvidia T4x2 GPU trên Kaggle và sử dụng framework PyTorch.

Siêu tham số (hyperparameters):

- d_model = 512
- d_ff = 2048
- N = 6 layers
- h = 8 heads

Chiến lược batching: Áp dụng Token-based Dynamic Batching với kích thước 6,000 tokens/batch. Kỹ thuật này gom nhóm các câu có độ dài tương đồng để giảm thiểu padding, tăng hiệu suất tính toán.

Tối ưu hóa (optimization): AdamW optimizer (beta1=0.9, beta2=0.98, epsilon=1e-9) kết hợp với Noam Scheduler và cơ chế Label Smoothing (epsilon=0.1)

F. Đánh giá và phân tích kết quả

1. Kết quả định lượng (Quantitative)

Bảng dưới đây trình bày hiệu năng của mô hình trên tập Public Test VLSP

Metric	Điểm số	Đánh giá
BLEU	42.82	Độ chính xác cao, vượt trội so với baseline thông thường.
TER	49.45	Tỷ lệ chỉnh sửa thấp, phản ánh cấu trúc câu đầu ra ổn định
METEOR	0.64	Khả năng nắm bắt tốt sự tương đồng ngữ nghĩa

2. Phân tích định tính (Qualitative)

Phân tích lỗi (error analysis) trên 500 mẫu ngẫu nhiên cho thấy:

- Ưu điểm: Mô hình dịch chính xác các thuật ngữ chuyên sâu (ví dụ: "dyslipidemia" / "rối loạn lipid máu").
- Vấn đề còn tồn tại:
  + Capitalization: Một số tên riêng viết tắt bị chuyển thành chữ thường (ví dụ: 'who' thay vì 'WHO')
  + Under-generation: Xu hướng lược bỏ một số chi tiết bổ trợ trong các câu quá dài (>80 tokens).

G. Sử dụng LLM Pretrain model.

1. Mô hình cơ sở.

- NLLB-200 – là mô hình được huấn luyện chuyên biệt cho việc dịch đa ngữ, phiên bản 600M tham số nhẹ tuy nhiên vẫn đảm bảo được chất lượng.

2. Kĩ thuật tinh chỉnh: LoRA

- Cơ chế: Đóng băng mô hình gốc, chỉ thêm vào các ma trận hạng thấp vào các lớp Attention (q_proj, v_proj, k_proj) và Feed-Forward (fc1, fc2).
- Tham số huấn luyện: Chỉ train khoảng **1-3%** tổng số tham số mô hình.

- Thư viện: Sử dụng PEFT của Hugging Face.

3. Kết quả

- BLEU Score: 33.6
- METEOR: 0.6199
- TER: 62.31

H. Kết luận

Trong bài toán này, hiệu quả của kiến trúc Transformer được tối ưu hóa cho bài toán dịch máy y tế đã được chứng minh. Kết quả BLEU 42.82 khẳng định việc thiết kế cẩn trọng mô hình và quy trình xử lý dữ liệu có thể mang lại hiệu năng có sức cạnh tranh tương đối tốt mà không cần phụ thuộc vào các mô hình tiền huấn luyện (Pre-trained) khổng lồ hay mô hình ngôn ngữ lớn (LLM) khi mô hình đó chỉ đạt được BLEU 33.6.

References

[1] Vaswani, A., et al. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems.

[2] Kudo, T., & Richardson, J. (2018). SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing. EMNLP 2018.

[3] Shazeer, N. (2020). GLU Variants Improve Transformer. ArXiv preprint.

[4] Xiong, R., et al. (2020). On Layer Normalization in the Transformer Architecture. ICML 2020.

[5] Press, O., & Wolf, L. (2017). Using the Output Embedding to Improve Language Models (Weight Tying). EACL 2017.

[6] Fan, A., Grave, E., & Joulin, A. (2019). Reducing Transformer Depth on Demand with Structured Dropout (LayerDrop). ICLR 2020.

[7] Press, O., & Wolf, L. (2017). Using the Output Embedding to Improve Language Models (Weight Tying). EACL 2017.

[8] VLSP Shared Task Organizers. VLSP 2025 Medical Machine Translation Dataset.