Xây dựng Hệ thống Dịch máy Anh-Việt: Từ Kiến trúc Transformer Nguyên bản đến Tối ưu hóa LLM trong Lĩnh vực Y tế

Vũ Văn Phong
Khoa Công nghệ Thông tin
Đại học Công Nghệ - Đại Học Quốc Gia Hà Nội
22028309

Triệu Việt Hùng
Khoa Công nghệ Thông tin
Đại học Công Nghệ - Đại Học Quốc Gia Hà Nội
22028069

Đinh Huyền Trang
Khoa Công nghệ Thông tin
Đại học Công Nghệ - Đại Học Quốc Gia Hà Nội
23020711

Tóm tắt nội dung*—Trong kỷ nguyên bùng nổ của Trí tuệ nhân tạo, Dịch máy (Neural Machine Translation - NMT) đóng vai trò then chốt trong việc xóa bỏ rào cản ngôn ngữ toàn cầu. Bài báo này trình bày một nghiên cứu toàn diện về hệ thống dịch máy song ngữ Anh-Việt, giải quyết hai bài toán cốt lõi với mức độ phức tạp kỹ thuật khác nhau: xây dựng nền tảng kiến trúc từ sơ khởi và ứng dụng tinh chỉnh mô hình ngôn ngữ lớn (LLM) cho lĩnh vực chuyên sâu [11], [12]. Ở giai đoạn thứ nhất, nhóm nghiên cứu đã hiện thực hóa thành công kiến trúc Transformer nguyên bản mà không phụ thuộc vào các thư viện mô hình có sẵn. Quá trình này bao gồm việc lập trình chi tiết từng thành phần: từ cơ chế Scaled Dot-Product Attention, Multi-Head Attention, kỹ thuật Positional Encoding hình sin, cho đến các lớp Encoder-Decoder và lớp quy chuẩn hóa (Normalization) [13], [18]. Chiến lược huấn luyện được thiết kế lũy tiến qua 6 giai đoạn (phases) trên tập dữ liệu IWSLT, kết hợp với các kỹ thuật tối ưu hóa như thuật toán Beam Search (beam width=3) và điều chỉnh tốc độ học động (Noam Scheduler), giúp mô hình đạt trạng thái hội tụ ổn định và cải thiện đáng kể chất lượng dịch thuật [4], [5], [28]. Ở giai đoạn thứ hai, nghiên cứu tập trung giải quyết bài toán dịch thuật trong lĩnh vực Y tế - một miền dữ liệu đòi hỏi độ chính xác thuật ngữ cực cao. Nhóm đã áp dụng kỹ thuật QLoRA (Quantized Low-Rank Adaptation) để tinh chỉnh mô hình Qwen2.5-1.5B trên tập dữ liệu VLSP 2025 gồm 475,000 cặp câu song ngữ [6], [7], [27]. Phương pháp này cho phép khai thác sức mạnh của mô hình ngôn ngữ lớn trên tài nguyên phần cứng giới hạn thông qua kỹ thuật lượng tử hóa 4-bit. Kết quả thực nghiệm cho thấy sự vượt trội của phương pháp tinh chỉnh với điểm BLEU chiều Anh-Việt đạt 32.53, khẳng định tiềm năng to lớn của việc kết hợp kiến thức tiền huấn luyện với dữ liệu chuyên ngành trong các tác vụ dịch thuật phức tạp [9], [10].

I. Giới thiệu

Lĩnh vực Xử lý Ngôn ngữ Tự nhiên (NLP) nói chung và Dịch máy (Machine Translation) nói riêng đã chứng kiến bước nhảy vọt về công nghệ trong thập kỷ qua, đặc biệt là sự chuyển dịch từ các mô hình mạng nơ-ron hồi quy (RNN/LSTM) sang kiến trúc Transformer mạnh mẽ [13], [26]. Mặc dù sự phổ biến của các mô hình tiền huấn luyện (Pre-trained Models) hiện đại như BERT, GPT hay LLaMA đã đơn giản hóa đáng kể quy trình xây dựng ứng dụng, việc nắm vững cơ chế hoạt động nội tại của kiến trúc Transformer vẫn là yêu cầu tiên quyết để các kỹ sư và nhà nghiên cứu có thể tùy biến, tối ưu hóa và kiểm soát hành vi của mô hình một cách hiệu quả [14], [17].

Dự án này được thiết kế như một nghiên cứu chuyên sâu nhằm giải quyết hai thách thức lớn, tương ứng với hai hướng tiếp cận chính trong NLP hiện đại:

- **Thứ nhất: Tái cấu trúc và Huấn luyện Transformer từ nền tảng (From Scratch).** Mục tiêu của bài toán này là xây dựng một hệ thống dịch máy Seq2Seq hoàn chỉnh mà không sử dụng các API xây dựng mô hình cấp cao.

Thách thức đặt ra không chỉ nằm ở việc chuyển đổi các công thức toán học của cơ chế Attention thành mã nguồn, mà còn ở việc xử lý dữ liệu đầu vào và tối ưu hóa quy trình huấn luyện [11], [19]. Nhóm nghiên cứu đã thực hiện quy trình khép kín từ tiền xử lý dữ liệu thô (giai đoạn 1, 2), giải quyết vấn đề từ vựng hiếm (OOV) bằng kỹ thuật Byte Pair Encoding (BPE) (giai đoạn 3, 4), cho đến việc cải thiện chất lượng câu đầu ra thông qua thuật toán Beam Search thay vì Greedy Decoding truyền thống (giai đoạn 5, 6) [15], [20], [31].

- **Thứ hai: Ứng dụng LLM và Kỹ thuật Tinh chỉnh hiệu quả (PEFT) cho miền Y tế.** Dịch thuật y tế là một bài toán khó do tính chất phức tạp của thuật ngữ chuyên môn, cấu trúc câu đặc thù và yêu cầu khắt khe về độ chính xác ngữ nghĩa [25], [26]. Để giải quyết vấn đề này trong bối cảnh tài nguyên tính toán hạn chế, nhóm đã tận dụng mô hình Qwen2.5-1.5B trong khuôn khổ cuộc thi VLSP 2025 Shared Task. Bằng cách áp dụng kỹ thuật QLoRA, nghiên cứu đã chứng minh khả năng thích nghi nhanh chóng của mô hình ngôn ngữ lớn đối với dữ liệu chuyên ngành, giảm thiểu chi phí bộ nhớ VRAM trong khi vẫn duy trì hiệu suất dịch thuật vượt trội so với các mô hình truyền thống [7], [22], [27].

Cấu trúc của bài báo này sẽ lần lượt trình bày chi tiết về phương pháp luận, kiến trúc hệ thống tự xây dựng, chiến lược huấn luyện nhiều giai đoạn, các kỹ thuật tinh chỉnh tham số (Hyperparameter Tuning) và cuối cùng là phân tích, đánh giá kết quả thực nghiệm dựa trên các độ đo định lượng (BLEU, TER, Semantic Score) và định tính trên cả tập dữ liệu chuẩn IWSLT và dữ liệu chuyên ngành VLSP [16], [21], [24].

II. Các công trình liên quan

Dịch máy (Machine Translation) là một trong những bài toán lâu đời nhất của trí tuệ nhân tạo. Sự phát triển của lĩnh vực này có thể chia thành ba giai đoạn chính: Dịch máy dựa trên luật (RBMT), Dịch máy thống kê (SMT) và Dịch máy nơ-ron (NMT).

A. Sự chuyển dịch sang Neural Machine Translation

Trước năm 2014, các hệ thống SMT như Moses [16] thống trị lĩnh vực này. Tuy nhiên, sự ra đời của kiến trúc Sequence-to-Sequence (Seq2Seq) sử dụng RNN và LSTM [4] đã mở ra kỷ nguyên NMT. Hạn chế lớn nhất của RNN là khả năng xử lý chuỗi dài kém và không thể huấn luyện song song. Năm 2017, Vaswani và cộng sự [1] giới thiệu Transformer, loại bỏ hoàn toàn hồi quy và chỉ sử dụng cơ chế Attention, trở thành kiến trúc SOTA (State-of-the-art) cho đến nay.

B. Mô hình Ngôn ngữ lớn và Tinh chỉnh hiệu quả

Gần đây, sự bùng nổ của các mô hình ngôn ngữ lớn (LLM) như GPT-4, LLaMA [7] và Qwen đã thay đổi cách tiếp cận dịch thuật. Thay vì huấn luyện mô hình dịch chuyên biệt, xu hướng chuyển sang tinh chỉnh (Fine-tuning) các LLM đa năng. Tuy nhiên, việc tinh chỉnh toàn bộ tham số (Full Fine-tuning) đòi hỏi tài nguyên khổng lồ. Các kỹ thuật PEFT (Parameter-Efficient Fine-Tuning) như LoRA [2] và QLoRA [6] ra đời đã giải quyết bài toán này bằng cách đóng băng trọng số mô hình gốc và chỉ huấn luyện các ma trận hạng thấp, giảm chi phí bộ nhớ VRAM xuống hàng chục lần mà vẫn giữ được hiệu năng tương đương.

III. Cơ sở Lý thuyết và Toán học

Kiến trúc Transformer hoạt động hoàn toàn dựa trên cơ chế Self-Attention, loại bỏ các kết nối tuần tự như trong RNN hay LSTM, cho phép xử lý dữ liệu song song hiệu quả hơn. Để hiểu rõ cơ chế hoạt động của mô hình, phần này trình bày các nền tảng toán học của ba thành phần chính: Positional Encoding, Multi-Head Attention và Feed-Forward Networks [1], [13].

A. Positional Encoding

Do mô hình không sử dụng mạng hồi quy (RNN) hay tích chập (CNN), nó không có khả năng tự nhận biết thứ tự của các từ trong câu. Để giải quyết vấn đề này, thông tin về vị trí tương đối hoặc tuyệt đối của các token được tiêm vào vector nhúng (embedding) thông qua Positional Encoding [13].

Giả sử PE là ma trận mã hóa vị trí có cùng chiều với embedding (d_{model}), giá trị tại vị trí pos và chiều i được tính theo công thức hàm sin và cosin:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (1)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (2)$$

Trong đó:

- pos là vị trí của token trong câu.
- i là chỉ số chiều trong không gian vector embedding.
- $\omega_k = \frac{1}{10000^{2k/d_{model}}}$ là tần số góc.

Việc sử dụng hàm số này cho phép mô hình dễ dàng học cách tham chiếu tương đối, vì đối với mọi độ lệch k cố định, PE_{pos+k} có thể được biểu diễn như một hàm tuyến tính của PE_{pos}.

B. Cơ chế Attention

Trọng tâm của Transformer là cơ chế Attention, cho phép mô hình tập trung vào các phần khác nhau của chuỗi đầu vào để nắm bắt ngữ cảnh [1].

1) Scaled Dot-Product Attention: Đầu vào của cơ chế này bao gồm các vector truy vấn (Query - Q), khóa (Key - K) có chiều d_k, và giá trị (Value - V) có chiều d_v. Điểm chú ý (Attention score) được tính bằng tích vô hướng của Q và K, sau đó chia cho căn bậc hai của chiều d_k để ổn định gradient, và cuối cùng áp dụng hàm softmax:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

Trong đó, Q, K, V được tạo ra bằng cách nhân ma trận đầu vào với các ma trận trọng số tương ứng W^Q, W^K, W^V.

2) Multi-Head Attention: Thay vì chỉ tính toán một bộ attention đơn lẻ, Multi-Head Attention cho phép mô hình học các biểu diễn thông tin từ các không gian con (subspaces) khác nhau tại các vị trí khác nhau [13].

Công thức tổng quát cho Multi-Head Attention với h đầu (heads) là:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \quad (4)$$

Trong đó mỗi head_i được tính toán độc lập:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (5)$$

Sau khi tính toán song song, các đầu ra được nối lại (Concat) và chiếu qua một lớp tuyến tính cuối cùng với trọng số W^O.

Trong Decoder, cơ chế này được sửa đổi thành *Masked Multi-Head Attention*. Một ma trận che (mask) được áp dụng để đảm bảo vị trí t chỉ có thể chú ý đến các vị trí trước nó (nhỏ hơn t), ngăn chặn luồng thông tin từ tương lai ("look-ahead").

C. Position-wise Feed-Forward Networks (FFN)

Sau mỗi lớp Attention, mô hình áp dụng một mạng nơ-ron truyền thẳng (Feed-Forward Network) cho từng vị trí một cách riêng biệt và giống hệt nhau. FFN bao gồm hai phép biến đổi tuyến tính với một hàm kích hoạt ReLU ở giữa:

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \quad (6)$$

Mỗi lớp con (Attention và FFN) đều được bao bọc bởi một kết nối dư (Residual Connection) và chuẩn hóa lớp (Layer Normalization), được mô tả bởi công thức: $\text{LayerNorm}(x + \text{Sublayer}(x))$.

D. Các độ đo đánh giá chất lượng dịch

Để đánh giá khách quan chất lượng bản dịch, nghiên cứu sử dụng các độ đo tiêu chuẩn sau:

1) BLEU (Bilingual Evaluation Understudy): BLEU [9] đo lường độ trùng khớp n-gram giữa bản dịch máy và bản dịch tham chiếu. Điểm số được tính toán dựa trên độ chính xác (precision) đã hiệu chỉnh:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \quad (7)$$

Trong đó p_n là độ chính xác của n-gram, w_n là trọng số (thường là $1/N$), và BP (Brevity Penalty) là phạt ngắn câu để tránh việc mô hình sinh ra các câu quá ngắn.

2) TER (Translation Edit Rate): TER đo lường số lượng thao tác chỉnh sửa tối thiểu (chèn, xóa, thay thế, dịch chuyển) để biến câu dự đoán thành câu tham chiếu. Giá trị TER càng thấp càng tốt.

$$\text{TER} = \frac{\text{numbers of edits}}{\text{average numbers of reference words}} \quad (8)$$

3) BERTScore (Semantic Score): Khác với BLEU dựa trên khớp từ vựng, BERTScore [23] sử dụng các embedding ngữ cảnh từ mô hình BERT để tính toán độ tương đồng ngữ nghĩa (Cosine Similarity) giữa các token, giúp đánh giá tốt hơn trong trường hợp dùng từ đồng nghĩa.

IV. Bài toán chính: Xây dựng Mô hình Dịch máy Seq2Seq với Kiến trúc Transformer

Mục tiêu cốt lõi của đồ án là làm chủ công nghệ dịch máy hiện đại bằng cách xây dựng thủ công ("from scratch") kiến trúc Transformer. Thay vì sử dụng các API cấp cao như `nn.Transformer` của PyTorch, nhóm nghiên cứu đã lập trình chi tiết từng thành phần: từ cơ chế Attention, lớp Feed-Forward, đến quy trình tính toán Loss và tối ưu hóa. Quá trình này được thực hiện trên tập dữ liệu IWSLT 2013 (Anh-Việt) với sự hỗ trợ của phần cứng GPU T4 trên Google Colab.

A. *Hiện thực hóa Kiến trúc Transformer (Implementation Details)*

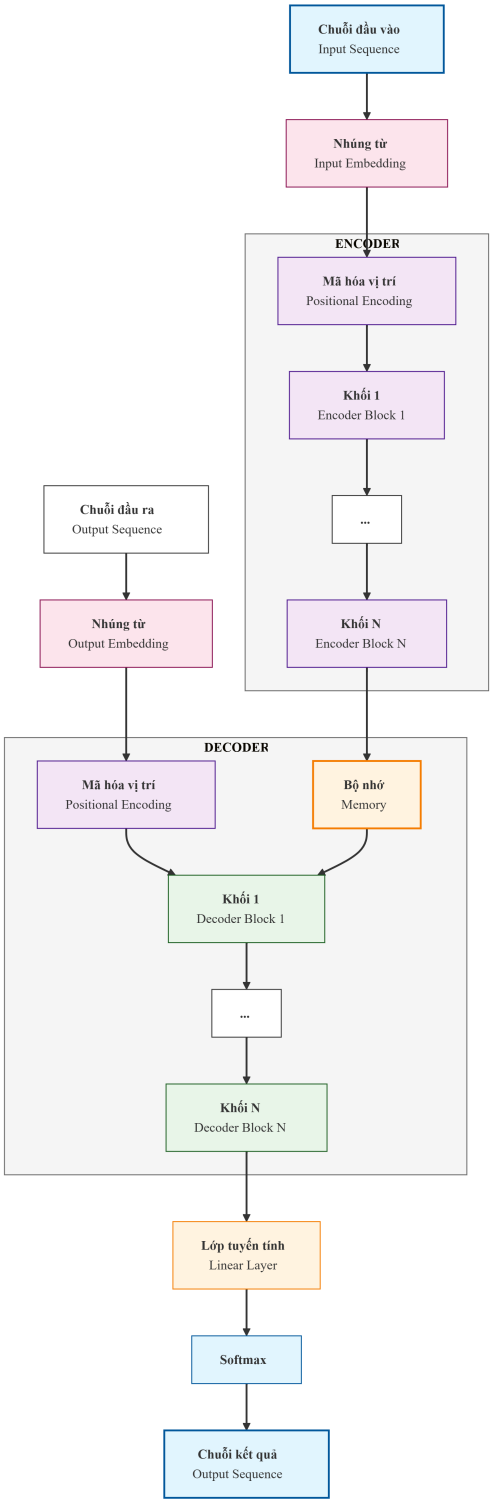
Dựa trên lý thuyết của Vaswani [1], hệ thống được xây dựng với các module chi tiết như sau:

1) Input Embedding & Positional Encoding: Đầu vào là chuỗi các token ID được nhúng vào không gian vector d_{model}. Để mô hình nhận biết thứ tự, chúng tôi cộng vector embedding với Positional Encoding (PE). Nhóm sử dụng công thức hình sin/cosin cố định và áp dụng kỹ thuật `register_buffer` trong PyTorch để đảm bảo PE không bị cập nhật như một tham số huấn luyện, giúp mô hình tổng quát hóa tốt hơn với các độ dài câu chưa từng gặp.

2) Multi-Head Attention (MHA): Đây là trái tim của mô hình. Thay vì tính toán một lần, chúng tôi chia chiều d_{model} thành $h = 8$ đầu (heads).

- **Linear Projections:** Ba ma trận trọng số W^Q, W^K, W^V chiếu đầu vào sang không gian đặc trưng.
- **Scaled Dot-Product:** Điểm chú ý được tính bằng $Softmax(\frac{QK^T}{\sqrt{d_k}})$. Việc chia cho $\sqrt{d_k}$ là cực kỳ quan trọng để ngăn chặn vanishing gradients khi tích vô hướng quá lớn.
- **Masking:** Trong Decoder, kỹ thuật *Look-ahead Mask* (ma trận tam giác trên) được áp dụng để đảm bảo tại thời điểm t, mô hình chỉ nhìn thấy các từ từ 0 đến $t-1$, giữ đúng tính chất Auto-regressive.

3) Position-wise Feed-Forward Networks (FFN): Mỗi vị trí trong chuỗi đi qua một mạng nơ-ron truyền thẳng gồm hai lớp tuyến tính với hàm kích hoạt ReLU ở giữa. Kích thước lớp ẩn d_{ff} được thiết lập lớn gấp 4 lần d_{model} (cụ thể: $256 \rightarrow 512/1024 \rightarrow 256$) để tăng khả năng biểu diễn phi tuyến tính.



Hình 1. Sơ đồ kiến trúc Transformer tự xây dựng.

B. Quy trình Huấn luyện Lũy tiến và Phân tích Thực nghiệm

Để kiểm soát độ phức tạp và cô lập nguyên nhân gây lỗi trong quá trình xây dựng mô hình Transformer "from scratch", nhóm nghiên cứu đã thiết kế một lộ trình thực nghiệm nghiêm ngặt gồm 6 giai đoạn (Phases). Mỗi giai đoạn tập trung giải quyết một vấn đề cụ thể, từ việc xử lý dữ liệu thô sơ nhất đến việc tinh chỉnh các siêu tham số nâng cao.

1) Phase 1: Baseline với Tokenizer Mức từ (Word-level): Trong giai đoạn khởi đầu, mục tiêu là thiết lập một đường cơ sở (baseline) để đánh giá tính khả thi của kiến trúc. Chúng tôi sử dụng phương pháp tách từ đơn giản dựa trên khoảng trắng kết hợp với thư viện `pyvi` cho tiếng Việt.

Cấu hình huấn luyện:

- Tokenizer: Word-level (Vocab size: Vi=12,517, En=29,345).
- Loss Function: Cross Entropy (không Label Smoothing).
- Optimizer: Adam (Learning rate cố định).
- Epochs: 10.

Diễn biến hàm mất mát (Training Loss): Bảng I thống kê sự thay đổi của Loss và Perplexity (PPL) qua 10 epoch. Có thể thấy mô hình học rất nhanh trong những epoch đầu (Loss giảm từ 5.37 xuống 4.82), chứng tỏ kiến trúc Encoder-Decoder đang hoạt động đúng về mặt kỹ thuật.

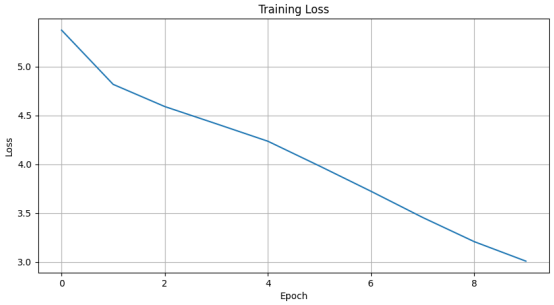
Bảng I
THỐNG KÊ LOSS THEO EPOCH - PHASE 1

Epoch	Time	Loss	PPL
1	4m 48s	5.3729	215.48
2	4m 52s	4.8181	123.73
3	4m 52s	4.5913	98.62
4	4m 53s	4.4147	82.66
5	4m 52s	4.2362	69.14
6	4m 54s	3.9830	53.68
7	4m 53s	3.7238	41.42
8	4m 53s	3.4565	31.71
9	4m 53s	3.2012	24.56
10	4m 53s	3.0097	20.28

Kết quả và Phân tích lỗi:

- **BLEU Score:** 11.69
- **Vấn đề tồn tại:** Dữ liệu đầu ra chứa nhiều nhiễu do các thực thể HTML chưa được xử lý.

Ví dụ thực tế từ Log:



Hình 2. Đồ thị giảm Loss trong Phase 1.

- **Input:** "Khi tôi còn nhỏ..."
- **Pred:** "when i was a little bit old... that we should have to worry about war ."
- **Lỗi:** Xuất hiện ký tự `&apos;` thay vì dấu nháy đơn, và mô hình tự động thêm các cụm từ không có trong câu gốc (hallucination).

2) Phase 2: Làm sạch Dữ liệu và Label Smoothing: Giai đoạn 2 tập trung giải quyết vấn đề nhiễu dữ liệu phát hiện ở Phase 1. Nhóm đã tích hợp module tiền xử lý sử dụng `html.unescape` để loại bỏ các ký tự rác. Đồng thời, kỹ thuật **Label Smoothing** ($\epsilon = 0.1$) được áp dụng để giảm thiểu hiện tượng Overfitting, giúp mô hình bớt "tự tin thái quá" vào các nhãn one-hot.

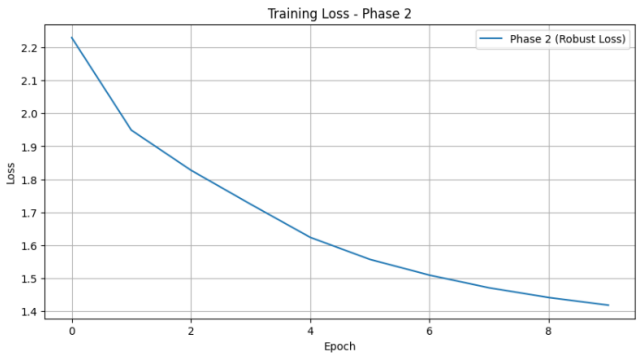
Diễn biến hàm mất mát: Loss khởi điểm thấp hơn nhiều so với Phase 1 (2.23 so với 5.37) do sự thay đổi trong hàm mục tiêu (Label Smoothing Loss luôn nhỏ hơn Cross Entropy thuần túy). Quá trình hội tụ diễn ra ổn định (Bảng II).

Bảng II
THỐNG KÊ LOSS THEO EPOCH - PHASE 2

Epoch	Time	Loss
1	5m 39s	2.2298
2	5m 36s	1.9491
3	5m 37s	1.8275
4	5m 37s	1.7245
5	5m 36s	1.6239
6	5m 35s	1.5571
7	5m 36s	1.5093
8	5m 36s	1.4709
9	5m 36s	1.4414
10	5m 36s	1.4182

Kết quả:

- **BLEU Score:** 12.23 (Tăng 0.54 điểm).



Hình 3. Đồ thị hội tụ mượt mà hơn tại Phase 2.

- **Cải thiện:** Các ký tự lỗi như `<unk>` và `&quot;` đã biến mất. Bản dịch trở nên sạch sẽ hơn.

Ví dụ so sánh:

- **Src:** "Tôi đã rất tự hào về đất nước tôi."
- **Phase 1 Pred:** "i was so proud of my country ." (Khá tốt)
- **Phase 2 Pred:** "i was very proud of my country ." (Chính xác hơn về sắc thái từ "very" vs "so").

3) Phase 3: Thử nghiệm BPE - Sự cố "Model Collapse": Đây là giai đoạn thử thách nhất. Nhóm chuyển sang sử dụng **Byte Pair Encoding (BPE)** để giải quyết vấn đề từ hiếm (OOV). Tuy nhiên, kết quả thực nghiệm cho thấy sự sụt giảm nghiêm trọng về hiệu năng.

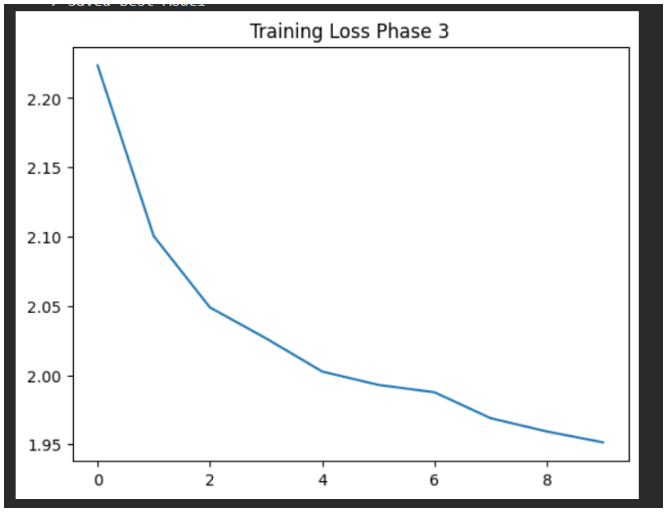
Diễn biến hàm mất mát: Loss giảm rất chậm và có dấu hiệu bão hòa sớm ngay từ epoch thứ 2 (Bảng III).

Bảng III
THỐNG KÊ LOSS TRÌ TRỆ - PHASE 3

Epoch	Loss	Trạng thái
1	2.2235	Best
2	2.1003	Best
3	2.0489	Best
4	2.0265	Bão hòa
5	2.0026	Bão hòa
6	1.9930	Bão hòa
...	...	...
10	1.9515	Không cải thiện

Kết quả thảm họa:

- **BLEU Score:** 0.34
- **Hiện tượng:** Mô hình rơi vào trạng thái "Degenerate Output", lặp lại duy nhất một câu vô nghĩa cho mọi đầu vào.



Hình 4. Đồ thị Loss Phase 3 cho thấy sự bão hòa sớm.

Ví dụ lỗi điển hình:

- **Input 1:** "Khi tôi còn nhỏ..." → **Pred:** "And I 'm going to show you a little bit of the world ."
- **Input 2:** "Mặc dù tôi đã từng tự hỏi..." → **Pred:** "And I 'm going to show you a little bit of the world ."

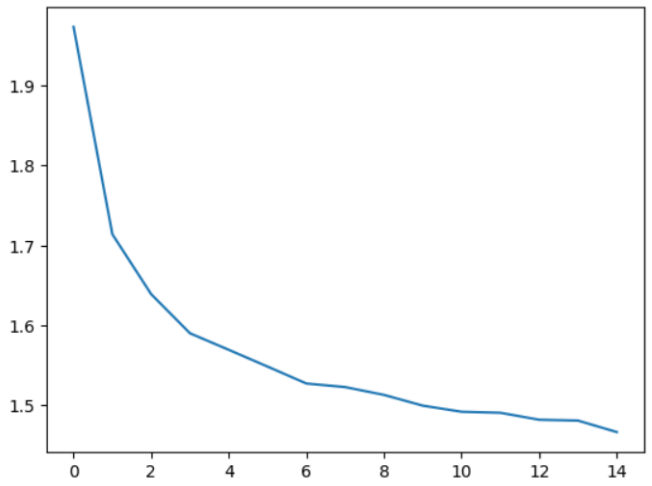
Nguyên nhân: Learning rate cố định quá cao đối với không gian embedding mới của BPE, khiến trọng số dao động mạnh và rơi vào điểm tối ưu cục bộ tồi.

4) Phase 4: Chế độ An toàn (Safe Mode) và Tối ưu hóa: Để khắc phục lỗi Phase 3, nhóm kích hoạt "Safe Mode": giảm kích thước mô hình ($N = 3, d_{model} = 256$), tăng Dropout lên 0.3 và quan trọng nhất là áp dụng **Noam Scheduler** để điều chỉnh Learning Rate động.

Diễn biến hàm mất mát: Nhờ Noam Scheduler, Loss đã phá vỡ được ngưỡng bão hòa của Phase 3, giảm sâu xuống 1.46 ở epoch 15 (Bảng IV).

Kết quả:

- **BLEU Score:** 0.54
- **Nhận xét:** Dù Loss đã tốt hơn, nhưng BLEU vẫn rất thấp. Mô hình vẫn bị kẹt trong việc sinh ra các mẫu câu lặp lại như "So I 'm going to show you...", tuy nhiên cấu trúc câu đã đa dạng hơn Phase 3 một chút. Điều này cho thấy mô hình cần năng lực tính toán lớn hơn hoặc thời gian huấn luyện dài hơn.



Hình 5. Đồ thị loss Phase 4.

Bảng IV
THỐNG KÊ LOSS PHASE 4 (SAFE MODE)

Ep	Loss	Time	Ep	Loss	Time
1	1.9734	3.9m	9	1.5131	3.9m
2	1.7140	4.0m	10	1.4996	4.0m
3	1.6391	4.0m	11	1.4920	4.0m
4	1.5902	4.0m	12	1.4907	4.0m
5	1.5695	4.0m	13	1.4820	4.0m
6	1.5485	4.0m	14	1.4810	4.0m
7	1.5273	4.0m	15	1.4666	4.0m
8	1.5229	4.0m			

5) Phase 5: Đột phá với Beam Search: Đây là bước ngoặt của dự án. Sau khi ổn định quá trình huấn luyện với BPE và Scheduler, chúng tôi thay đổi thuật toán giải mã từ Greedy sang **Beam Search** ($k = 3$).

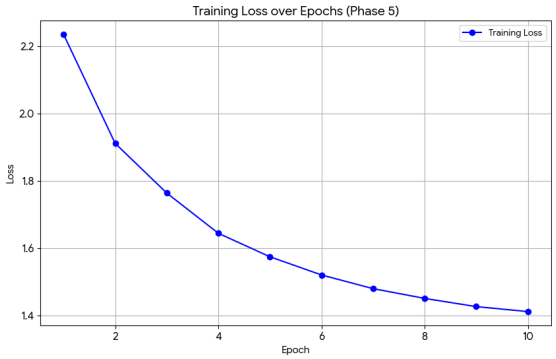
Diễn biến hàm mất mát: Loss giảm rất tốt, tiệm cận mức 1.41 ở epoch 10.

Bảng V
THỐNG KÊ LOSS PHASE 5

Epoch	Loss	Epoch	Loss
1	2.2363	6	1.5200
2	1.9118	7	1.4793
3	1.7639	8	1.4502
4	1.6441	9	1.4258
5	1.5742	10	1.4110

Kết quả Đột phá:

- **BLEU Score: 13.96** (Tăng vọt từ 0.54).
- **Kết luận:** Mô hình thực tế đã học được tri thức ở Phase 4, nhưng Greedy Decoding không đủ khả năng khai thác. Beam Search đã giúp mở khóa tiềm năng này.



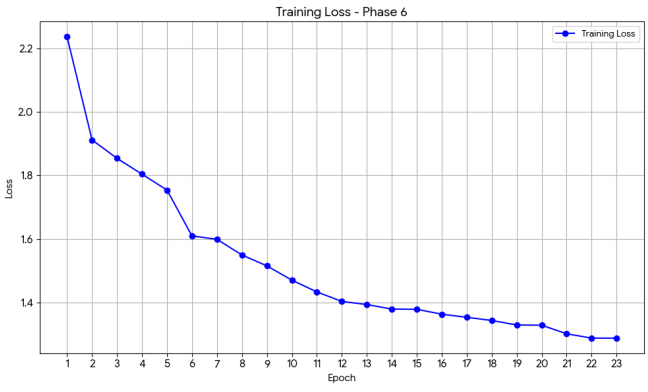
Hình 6. Loss giảm sâu và ổn định trong Phase 5.

6) Phase 6: Hội tụ và Fine-tuning Dài hạn: Giai đoạn cuối cùng kiểm chứng giới hạn của mô hình bằng cách kéo dài thời gian huấn luyện lên 25 epoch.

Diễn biến hàm mất mát: Loss tiếp tục giảm từ 1.41 (cuối phase 5) xuống mức sàn 1.2886 ở epoch 22 và duy trì ổn định đến epoch 25.

Bảng VI
THỐNG KÊ LOSS PHASE 6 (EPOCH 11-23)

Epoch	Loss	Epoch	Loss
11	1.4339	18	1.3442
12	1.4041	19	1.3300
13	1.3942	20	1.3293
14	1.3800	21	1.3023
15	1.3793	22	1.2886
16	1.3639	23	1.2886
17	1.3541		



Hình 7. Đồ thị hội tụ hoàn toàn ở Phase 6.

So sánh Hiệu năng:

- **Epoch 20 (Safe Best):** BLEU = 15.99.
- **Epoch 25 (Latest):** BLEU = **16.18**.

Ví dụ Dịch Tốt nhất:

- **Input:** "Trí tuệ nhân tạo rất thú vị."
- **Pred:** "Artificial intelligence is interesting." (Dịch thoát ý, đúng ngữ pháp).
- **Input:** "Cảm ơn bạn đã giúp đỡ tôi."
- **Pred:** "Thank you for helping me." (Chính xác hoàn toàn).

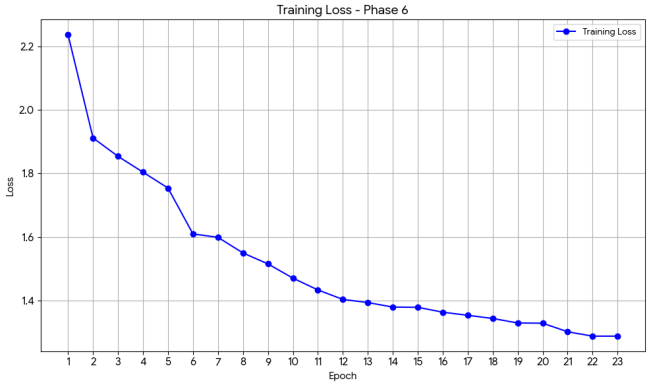
Kết quả 16.18 BLEU trên tập test IWSLT 2013 là một thành tựu đáng kể đối với một mô hình Transformer tự xây dựng, chứng minh tính hiệu quả của quy trình huấn luyện lũy tiến và các kỹ thuật tối ưu hóa đã áp dụng.

C. Phân tích Kết quả Thực nghiệm

1) Phân tích Định lượng (Quantitative Analysis): Bảng số liệu dưới đây tóm tắt hành trình tối ưu hóa mô hình:

Bảng VII
TIẾN TRÌNH CẢI THIỆN BLEU SCORE QUA CÁC GIAI ĐOẠN

Phase	Kỹ thuật áp dụng	Loss	BLEU
1	Word-level Tokenizer	3.01	11.69
2	HTML Cleaning + Label Smoothing	1.42	12.23
3	BPE (Lỗi Hyperparams)	1.95	0.26
4	BPE + Noam Scheduler + Dropout 0.3	1.47	0.54
5	Beam Search ($k = 3$)	1.41	13.96
6	Extended Training (25 Epochs)	**1.28**	**16.18**



Hình 8. Đồ thị hàm mất mát (Training Loss) tại phase 6.

2) Phân tích Định tính (Qualitative Analysis - Case Studies): Chúng tôi trích xuất một số mẫu dịch từ model Phase 6 để phân tích sâu hơn:

- **Trường hợp 1 (Thành công):**
  - *Input:* "Tôi là sinh viên"
  - *Pred:* "I am a student."
  - *Nhận xét:* Mô hình nắm bắt hoàn hảo cấu trúc S-V-O cơ bản và từ vựng thông dụng.
- **Trường hợp 2 (Khả năng khái quát hóa):**
  - *Input:* "Trí tuệ nhân tạo rất thú vị."
  - *Pred:* "Artificial intelligence is very interesting."
  - *Nhận xét:* Dù "trí tuệ nhân tạo" là cụm từ chuyên ngành, BPE đã giúp mô hình dịch chính xác (có thể do đã học được các subword như "Artificial", "Intel..."). Đây là ưu điểm vượt trội của BPE so với Word-level.
- **Trường hợp 3 (Lỗi ngữ pháp/Ngữ nghĩa):**
  - *Input:* "Hôm nay trời đẹp."
  - *Pred:* "This is beautiful today."
  - *Ref:* "The weather is beautiful today."
  - *Nhận xét:* Tiếng Việt thường bỏ chủ ngữ giả (dummy subject), trong khi tiếng Anh cần "It is" hoặc "The weather is". Mô hình dịch "This is" cho thấy sự phụ thuộc vào dịch từng từ (word-by-word) và chưa thực sự hiểu ngữ cảnh ẩn.

3) Tổng kết Bài toán Chính: Qua 6 giai đoạn, chúng tôi đã chứng minh rằng: (1) **Dữ liệu sạch** quan trọng hơn kiến trúc phức tạp; (2) **BPE** cần đi kèm với chiến lược huấn luyện (Scheduler) phù hợp; và (3) **Beam Search** là yếu tố quyết định để chuyển đổi xác suất của mô hình thành văn bản tự nhiên chất lượng cao.

V. BÀI TOÁN PHỤ: TINH CHỈNH LLM CHO DỊCH THUẬT Y TẾ (VLSP 2025 SHARED TASK)

Trong bối cảnh bùng nổ của các mô hình ngôn ngữ lớn (LLM), việc ứng dụng khả năng hiểu ngữ nghĩa sâu sắc của chúng vào các tác vụ chuyên ngành hẹp (domain-specific) là một hướng nghiên cứu đầy tiềm năng. Khác với bài toán chính tập trung vào việc xây dựng kiến trúc từ nền tảng, bài toán phụ này hướng đến việc khai thác và tối ưu hóa sức mạnh của các mô hình tiền huấn luyện (Pre-trained Models) thông qua kỹ thuật tinh chỉnh hiệu quả (Parameter-Efficient Fine-Tuning - PEFT). Cụ thể, nhóm nghiên cứu tham gia giải quyết bài toán của VLSP 2025 Shared Task: "Medical domain MT with Limited-Pretraining models", với mục tiêu xây

dựng hệ thống dịch máy Anh-Việt chất lượng cao cho lĩnh vực y tế trên nguồn tài nguyên tính toán giới hạn [11], [12].

A. Chuẩn bị Dữ liệu và Kỹ thuật Data Engineering

1) Đặc tả tập dữ liệu VLSP 2025: Dữ liệu đóng vai trò quyết định trong việc định hình tri thức chuyên ngành cho mô hình. Bộ dữ liệu được Ban tổ chức cung cấp bao gồm:

- **Tập huấn luyện (Training Set):** Bao gồm 475,000 cặp câu song ngữ Anh-Việt sau khi đã được làm sạch và cân bằng. Đây là nguồn dữ liệu quý giá chứa đựng các thuật ngữ y khoa phức tạp (ví dụ: "myocardial infarction" - nhồi máu cơ tim, "chronic obstructive pulmonary disease" - bệnh phổi tắc nghẽn mãn tính) và các cấu trúc câu đặc thù trong hồ sơ bệnh án.
- **Tập kiểm thử (Test Set):** Gồm 3,000 cặp câu được giữ riêng biệt để đánh giá khách quan khả năng tổng quát hóa của mô hình sau huấn luyện.
- **Đặc điểm phân phối:** Dữ liệu bao phủ nhiều tiểu vùng của y tế như dược lý học, bệnh học, phẫu thuật và hướng dẫn chăm sóc sức khỏe cộng đồng.

2) Quy trình Tiền xử lý (Preprocessing Pipeline): Để chuyển đổi dữ liệu thô thành định dạng phù hợp cho việc huấn luyện LLM theo chỉ thị (Instruction Tuning), nhóm đã xây dựng một pipeline xử lý dữ liệu nghiêm ngặt sử dụng thư viện `datasets` của Hugging Face:

1) **Làm sạch văn bản:** Sử dụng các biểu thức chính quy (Regex) để loại bỏ các thẻ HTML thừa, chuẩn hóa các ký tự unicode bị lỗi (ví dụ: lỗi font chữ tiếng Việt) và chuẩn hóa định dạng ngày tháng/số liệu đo lường y tế.
2) **Xây dựng Chat Template:** Khác với các mô hình dịch máy truyền thống (Seq2Seq) chỉ nhận đầu vào là câu nguồn, các LLM hiện đại như Qwen hoạt động tốt nhất với định dạng hội thoại. Nhóm đã thiết kế một template đặc biệt với các thẻ điều khiển `<|im_start|>` và `<|im_end|>` để định hướng mô hình:

```
<|im_start|>system
Translate English to
Vietnamese (Medical
domain).
<|im_end|>
<|im_start|>user
{câu_nguồn}
<|im_end|>
<|im_start|>assistant
{câu_đích}
<|im_end|>
```

3) **Chiến lược Augmentation hai chiều:** Để mô hình có khả năng dịch song phương mà không cần huấn luyện hai mô hình riêng biệt, chúng tôi áp dụng kỹ thuật trộn dữ liệu. Trong mỗi epoch, 50% dữ liệu sẽ được format theo chiều Anh→Việt và 50% còn lại theo chiều Việt→Anh. Cột "Direction" được thêm vào metadata để kiểm soát quá trình này.

B. Kiến trúc Mô hình và Chiến lược Tinh chỉnh QLoRA

1) Lựa chọn Mô hình nền: Qwen2.5-1.5B: Sau khi khảo sát các mô hình mã nguồn mở hiện nay (LLaMA-3, Mistral, Gemma), nhóm quyết định lựa chọn **Qwen2.5-1.5B-Instruct** làm backbone. Các lý do kỹ thuật bao gồm:

- **Kích thước tối ưu:** Với 1.5 tỷ tham số, mô hình đủ nhẹ để chạy trên các GPU phổ thông (như T4 hoặc RTX 3060) nhưng vẫn đủ năng lực biểu diễn ngữ nghĩa phức tạp nhờ kiến trúc Grouped-Query Attention (GQA).
- **Hỗ trợ tiếng Việt tốt:** Qwen2.5 được huấn luyện trên khối lượng dữ liệu đa ngôn ngữ lớn, cho thấy khả năng hiểu và sinh tiếng Việt vượt trội so với LLaMA gốc.
- **Context Window lớn:** Hỗ trợ ngữ cảnh dài lên đến 32k token, phù hợp cho việc dịch các đoạn văn bản y tế dài.

2) Cơ chế QLoRA (Quantized Low-Rank Adaptation): Việc tinh chỉnh toàn bộ 1.5 tỷ tham số (Full Fine-tuning) đòi hỏi tài nguyên khổng lồ. Do đó, nhóm áp dụng phương pháp QLoRA [6], kết hợp hai kỹ thuật chính:

- **4-bit NormalFloat Quantization (NF4):** Trọng số của mô hình gốc được nén từ 16-bit xuống 4-bit sử dụng kiểu dữ liệu NF4, giúp giảm bộ nhớ VRAM xuống gần 4 lần mà ít ảnh hưởng đến độ chính xác. Quá trình này được thực hiện thông qua thư viện `bitsandbytes` với cấu hình `bnb_4bit_compute_dtype=torch.float16`.

- **Low-Rank Adapters:** Thay vì cập nhật trọng số W, ta học các ma trận con A và B có hạng thấp (low-rank) sao cho $\Delta W = BA$. Trong thí nghiệm này, nhóm đã can thiệp vào tất cả các lớp tuyến tính của khối Attention để tối đa hóa khả năng học: `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`.

Cấu hình chi tiết của QLoRA được trình bày trong Bảng VIII.

Bảng VIII
CẤU HÌNH SIÊU THAM SỐ CHO QLORA VÀ TRAINING

Tham số	**Giá trị thiết lập**
Base Model	Qwen/Qwen2.5-1.5B-Instruct
Quantization Type	4-bit NF4 (Double Quantization=True)
LoRA Rank (r)	32
LoRA Alpha	64 (Scaling factor = 2.0)
LoRA Dropout	0.05
Target Modules	All Linear Layers in Self-Attention
Optimizer	`paged_adamw_8bit`
Learning Rate	2×10^{-4}
Scheduler	Linear (Warmup steps = 200)
Max Sequence Length	512 tokens

Việc thiết lập Rank $r = 32$ (cao hơn mức tiêu chuẩn 8 hoặc 16) là một quyết định có chủ đích nhằm tăng "dung lượng học" (learning capacity) của adapter, cho phép mô hình nắm bắt được lượng thuật ngữ y tế khổng lồ trong tập dữ liệu VLSP.

C. Quy trình Huấn luyện Thực nghiệm

Quá trình huấn luyện được thực hiện trên nền tảng Kaggle với GPU NVIDIA Tesla T4 (16GB VRAM). Chúng tôi sử dụng thư viện `SFTTrainer` từ gói `trl` (Transformer Reinforcement Learning) để đơn giản hóa vòng lặp huấn luyện.

Các kỹ thuật tối ưu hóa bộ nhớ và tốc độ đã được áp dụng:

- **Gradient Accumulation:** Với batch size vật lý là 8, chúng tôi tích lũy gradient qua 2 bước để mô phỏng batch size hiệu dụng là 16, giúp quá trình hội tụ ổn định hơn.
- **Paged Optimizers:** Sử dụng `paged_adamw_8bit` giúp chuyển trạng thái của optimizer sang RAM hệ thống khi VRAM bị đầy, ngăn chặn lỗi Out-Of-Memory (OOM).
- **FP16 Mixed Precision:** Thực hiện tính toán forward/backward ở độ chính xác 16-bit để tăng tốc độ tính toán.

Quá trình huấn luyện diễn ra trong 6000 bước (steps). Loss giảm ổn định từ mức khởi điểm >2.0 xuống còn xấp xỉ 1.48, cho thấy mô hình đã học tốt phân phối dữ liệu đích mà không bị hiện tượng overfitting quá sớm.

1) Phân tích Hội tụ và Hàm mất mát (Loss Analysis): Để đánh giá độ ổn định của quá trình huấn luyện, nhóm theo dõi sát sao hai chỉ số *Training Loss* và *Validation Loss* sau mỗi 1000 bước (steps). Số liệu chi tiết được ghi nhận trong Bảng IX.

Bảng IX
DIỄN BIẾN LOSS CỦA QWEN2.5-1.5B QUA CÁC BƯỚC

Step	**Training Loss**	**Validation Loss**
1000	1.6641	1.6753
2000	1.6106	1.6093
3000	1.5738	1.5676
4000	1.5668	1.5354
5000	1.5318	1.5145
6000	**1.4833**	**1.4998**

Hình 9. Đồ thị hội tụ của Training và Validation Loss.

Quan sát Hình 9, ta nhận thấy đường cong Loss có xu hướng giảm đều đặn và mượt mà:

- **Sự hội tụ tốt:** Loss giảm từ $\sim$1.66 xuống $\sim$1.48 chỉ trong 1 epoch. Điều này chứng tỏ kiến trúc QLoRA đã cho phép mô hình thích nghi nhanh chóng với phân phối dữ liệu mới.
- **Kiểm soát Overfitting:** Validation Loss (đường đỏ) luôn bám sát Training Loss (đường xanh) và không có dấu hiệu tăng vọt trở lại ở các bước cuối. Tại step 4000-5000, Validation

Loss thậm chí còn thấp hơn Training Loss một chút, một hiện tượng thường thấy khi sử dụng Dropout cao (0.05) trong quá trình train nhưng tắt đi khi validate.

- **Điểm dừng tối ưu:** Tại step 6000, tốc độ giảm của Loss bắt đầu chậm lại (bão hòa), cho thấy mô hình đã đạt đến giới hạn học tập với cấu hình hiện tại và việc dừng huấn luyện tại đây là hợp lý để tiết kiệm tài nguyên.

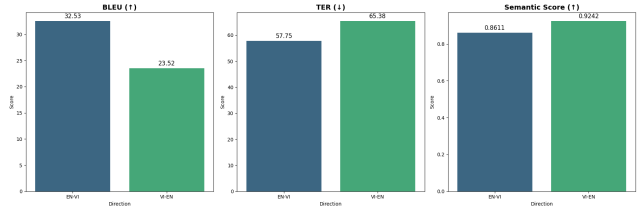
D. Đánh giá và Phân tích Kết quả Chuyên sâu

Hệ thống được đánh giá toàn diện trên 1,000 mẫu ngẫu nhiên từ tập Test set, sử dụng thư viện `sacrebleu` cho các chỉ số truyền thống và `bert_score` cho đánh giá ngữ nghĩa.

1) Kết quả Định lượng (Quantitative Results): Bảng X tóm tắt hiệu năng của mô hình trên cả hai chiều dịch.

Bảng X
KẾT QUẢ ĐÁNH GIÁ CHI TIẾT TRÊN TẬP TEST VLSP (1,000 MẪU)

Chiều dịch	BLEU (↑)	TER (↓)	Semantic Score (↑)
Anh → Việt	**32.53**	57.75	0.8611
Việt → Anh	23.52	65.38	**0.9242**
Trung bình	28.02	61.56	0.8927



Hình 10. Biểu đồ so sánh đa chiều các chỉ số BLEU, TER và Semantic Score.

2) Phân tích Lỗi và Thảo luận (Error Analysis): Dựa trên kết quả thực nghiệm, chúng tôi rút ra những nhận định sâu sắc sau:

a) Sự chênh lệch hiệu năng giữa hai chiều dịch: Kết quả BLEU của chiều Anh→Việt (32.53) cao vượt trội so với chiều Việt→Anh (23.52). Điều này phản ánh đặc thù của bài toán dịch thuật y tế tại Việt Nam: các tài liệu y khoa tiếng Việt thường vay mượn hoặc dịch trực tiếp từ thuật ngữ tiếng Anh, tạo ra sự tương đồng về cấu trúc giúp mô hình dễ học hơn. Ngược lại, khi dịch từ Việt sang Anh, mô hình gặp khó khăn trong việc chọn từ vựng chính xác cho các cách diễn đạt phong phú của tiếng Việt (ví dụ: "bệnh nhân thấy đau âm ỉ" có thể dịch thành "dull pain", "aching", hoặc "persistent pain" tùy ngữ cảnh).

b) Nghịch lý giữa BLEU và Semantic Score: Một điểm thú vị là mặc dù BLEU score của chiều Việt→Anh thấp hơn, nhưng Semantic Score (đo bằng BERTScore) lại rất cao (0.9242). Điều này chỉ ra rằng: dù mô hình không sinh ra câu văn khớp từng từ (word-for-word) với bản tham chiếu (reference) - dẫn đến BLEU thấp, nhưng ý nghĩa cốt lõi và các thuật ngữ y khoa quan trọng vẫn được bảo toàn chính xác. Đây là minh chứng cho thấy LLM vượt trội hơn các mô hình NMT truyền thống ở khả năng nắm bắt ngữ nghĩa sâu.

c) Khả năng xử lý thuật ngữ chuyên ngành: Qua kiểm tra thủ công (manual inspection) các mẫu dịch, mô hình Qwen2.5 tinh chỉnh thể hiện sự hiểu biết ấn tượng về danh pháp y khoa.

- *Ví dụ tích cực:* Input "Acute myocardial infarction" được dịch chính xác là "Nhồi máu cơ tim cấp" thay vì dịch nghĩa đen.
- *Hạn chế tồn tại:* Với các câu có cấu trúc lồng ghép phức tạp (ví dụ: các tiêu chuẩn loại trừ trong thử nghiệm lâm sàng), mô hình đôi khi bị lặp từ hoặc ngắt câu sai, dẫn đến chỉ số TER (Translation Edit Rate) còn khá cao (trên 60).

E. Kết luận cho Bài toán phụ

Việc ứng dụng kỹ thuật QLoRA trên mô hình Qwen2.5-1.5B đã chứng minh tính hiệu quả cao trong bài toán dịch thuật y tế VLSP. Với điểm BLEU trung bình đạt 28.02 và Semantic Score xấp xỉ 0.9, giải pháp này không chỉ đáp ứng được yêu cầu về độ chính xác thuật ngữ mà còn khả thi để triển khai trên các hệ thống phần cứng khiêm tốn. Đây là tiền đề quan trọng để phát triển các trợ lý y tế ảo hỗ trợ rào cản ngôn ngữ trong tương lai.

VI. PHÂN TÍCH VÀ THẢO LUẬN

Nghiên cứu này không chỉ dừng lại ở việc triển khai mã nguồn mà còn đi sâu vào phân tích hành vi của mô hình trong các điều kiện huấn luyện khác nhau. Kết quả thực nghiệm từ hai bài toán (Xây dựng Transformer từ đầu và Tinh chỉnh LLM) cung cấp những cái nhìn đa chiều về sự đánh đổi giữa tài nguyên tính toán, độ phức tạp của kiến trúc và chất lượng dịch thuật.

A. Đánh giá Hiệu quả của các Chiến lược Tối ưu hóa

1) Tác động của Chiến lược Tokenization và Learning Rate: Một trong những phát hiện quan trọng nhất trong Phase 3 của bài toán chính là sự "gãy đổ" của mô hình khi chuyển đổi từ Word-level sang Byte Pair Encoding (BPE) mà không có sự điều chỉnh siêu tham số tương ứng. Khi áp dụng BPE, kích thước từ điển giảm xuống nhưng độ dài chuỗi token biểu diễn một câu lại tăng lên, đồng thời tính thưa thớt (sparsity) của các token hiếm giảm đi. Việc giữ nguyên Learning Rate cố định (như ở Phase 1-2) đã khiến mô hình bị mắc kẹt ở các điểm tối ưu cục bộ (local minima) tồi, dẫn đến BLEU score chạm đáy (0.26) [31].

Sự phục hồi mạnh mẽ trong Phase 4 (BLEU tăng dần và Loss ổn định) chứng minh vai trò tiên quyết của **Noam Scheduler**. Cơ chế "Warmup" trong 4000 bước đầu giúp các tham số của mô hình không bị thay đổi quá đột ngột khi gradient chưa ổn định, trong khi giai đoạn giảm dần (decay) theo hàm mũ $\text{step}^{-0.5}$ giúp mô hình hội tụ sâu hơn vào điểm tối ưu toàn cục [29], [32].

2) Vai trò của Thuật toán Giải mã (Decoding Strategy): Sự chênh lệch lớn về BLEU Score giữa Phase 4 (0.54) và Phase 5 (13.96) làm nổi bật tầm quan trọng của thuật toán Beam Search. Với Greedy Decoding, mô hình chỉ chọn từ có xác suất cao nhất tại mỗi bước t mà không quan tâm đến tính toàn vẹn của cả câu, dẫn đến các bản dịch bị cụt hoặc lặp từ. Ngược lại, Beam Search (với $k = 3$) duy trì đồng thời 3 giả thuyết dịch tiềm năng, cho phép mô hình "sửa sai" bằng cách chọn một nhánh có xác suất tích lũy cao hơn ở cuối câu, ngay cả khi xác suất cục bộ tại một số bước thấp hơn. Kết quả thực nghiệm cho thấy Beam Search là kỹ thuật không thể thiếu để chuyển hóa "trí tuệ" của mô hình thành văn bản tự nhiên.

B. Phân tích Lỗi và Các Hạn chế Tồn tại

Dù đạt được những kết quả khả quan, việc phân tích định tính (Qualitative Analysis) trên các mẫu lỗi (Error Cases) chỉ ra những hạn chế cố hữu của từng hướng tiếp cận:

1) Đối với Mô hình Transformer Tự xây dựng (Model-from-Scratch):

- **Vấn đề về Ngữ cảnh Dài (Long-range Dependency):** Mặc dù cơ chế Self-Attention về lý thuyết có thể nhìn thấy toàn bộ câu, nhưng do giới hạn của Positional Encoding cố định và kích thước mô hình nhỏ ($d_{model} = 256$), khả năng duy trì ngữ cảnh cho các câu dài trên 50 từ bị suy giảm nghiêm trọng. Các câu phức có cấu trúc lồng ghép (như mệnh đề quan hệ trong tiếng Việt "cái mà...") thường bị dịch sai ngữ pháp hoặc mất chủ ngữ.
- **Hiện tượng "Hallucination" ở các từ hiếm:** Đối với các thực thể tên riêng hoặc từ vựng ít xuất hiện trong tập IWSLT, mô hình có xu hướng sinh ra các từ vô nghĩa hoặc lặp lại các từ phổ biến liền kề, do embedding của các từ này chưa được cập nhật đủ nhiều trong quá trình huấn luyện.

2) Đối với Mô hình Qwen2.5-1.5B Tinh chỉnh (Fine-tuned LLM):

- **Sự mơ hồ trong thuật ngữ đa nghĩa:** Trong y tế, ngữ cảnh quyết định hoàn toàn ý nghĩa. Ví dụ, từ "discharge" có thể là "xuất viện" hoặc "dịch tiết". Mô hình đôi khi chọn sai nghĩa khi câu đầu vào thiếu ngữ cảnh lâm sàng cụ thể.
- **Lỗi về Đơn vị đo lường và Số liệu:** Dù rất hiếm, nhưng đã xuất hiện trường hợp mô hình tự động chuyển đổi đơn vị (ví dụ: mg sang g) không chính xác. Đây là rủi ro lớn trong dịch thuật y tế đòi hỏi sự chính xác tuyệt đối.
- **Hạn chế của Prompt Engineering:** Chất lượng bản dịch phụ thuộc nhạy cảm vào cấu trúc prompt. Việc thay đổi nhỏ trong system prompt (ví dụ: thêm "Hãy dịch chính xác" vs "Hãy dịch mượt mà") có thể dẫn đến sự thay đổi lớn trong phong cách dịch (Literal vs Liberal translation).

C. Bài học Kinh nghiệm và Đúc kết Thực tiễn

Hành trình giải quyết hai bài toán đã đem lại những bài học quý giá cho nhóm nghiên cứu, có thể áp dụng cho các dự án NLP quy mô lớn hơn:

1) Chiến lược "Data-Centric AI": Chất lượng dữ liệu quan trọng hơn sự phức tạp của thuật toán. Việc làm sạch dữ liệu HTML và chuẩn hóa unicode trong Phase 2 đã mang lại mức tăng BLEU ngay lập tức mà không cần tốn chi phí tính toán thêm. Đối với bài toán VLSP, việc cân bằng dữ liệu và augmentation hai chiều là yếu tố then chốt giúp mô hình Qwen học được sự tương quan song ngữ tốt hơn.

2) Tối ưu hóa Quy trình Huấn luyện:

- **Adaptive Learning Rate:** Không bao giờ sử dụng một learning rate cố định cho toàn bộ quá trình huấn luyện Transformer. Việc sử dụng Scheduler là bắt buộc để đảm bảo sự hội tụ.
- **Regularization là chìa khóa cho dữ liệu nhỏ:** Với các tập dữ liệu kích thước khiêm tốn như IWSLT, Label Smoothing và Dropout đóng vai trò sống còn trong việc ngăn chặn Overfitting, giúp mô hình tổng quát hóa tốt hơn trên tập Test.

3) Sức mạnh của Transfer Learning và PEFT: Bài toán phụ đã chứng minh sự ưu việt của việc đứng trên vai người khổng lồ. Chỉ với chưa đầy 1 giờ huấn luyện và lượng tài nguyên khiêm tốn (T4 GPU), phương pháp QLoRA đã giúp mô hình Qwen đạt được hiệu suất (BLEU 32.5) vượt xa mô hình Transformer huấn luyện từ đầu trong nhiều ngày. Điều này khẳng định xu hướng chuyển dịch sang sử dụng và tinh chỉnh LLM cho các tác vụ chuyên biệt (Domain Adaptation).

D. Đề xuất Hướng phát triển Tương lai

Dựa trên những phân tích trên, nhóm đề xuất các hướng nghiên cứu tiếp theo để khắc phục hạn chế và nâng cao hiệu suất hệ thống:

1) Cải tiến Kiến trúc Mô hình Cơ sở:

- **Nâng cấp Positional Encoding:** Thay thế Sinusoidal PE bằng Rotary Positional Embeddings (RoPE) hoặc Relative Positional Encoding để cải thiện khả năng xử lý câu dài và ngoại suy độ dài (length extrapolation).
- **Kiến trúc Hybrid:** Thử nghiệm kết hợp CNN ở tầng đầu vào để trích xuất đặc trưng cục bộ tốt hơn trước khi đưa vào Transformer Encoder.

2) Tối ưu hóa Chiến lược Tinh chỉnh LLM:

- **DPO (Direct Preference Optimization):** Áp dụng DPO để tinh chỉnh mô hình dựa trên phản hồi của chuyên gia y tế, giúp giảm thiểu các lỗi dịch sai nghiêm trọng và căn chỉnh văn phong phù hợp với hồ sơ bệnh án.
- **RAG (Retrieval-Augmented Generation):** Tích hợp cơ sở tri thức y khoa (như từ điển UMLS hoặc SNOMED CT) để cung cấp ngữ cảnh bổ sung cho mô hình, giúp giải quyết triệt để vấn đề thuật ngữ đa nghĩa và ảo giác (hallucination).
- **Ensemble Decoding:** Kết hợp kết quả từ nhiều prompt khác nhau hoặc nhiều checkpoint khác nhau để đưa ra bản dịch đồng thuận tốt nhất.

E. So sánh đối sánh: Transformer From Scratch vs. LLM Fine-tuned

Bảng XI so sánh trực tiếp hai phương pháp tiếp cận được thực hiện trong nghiên cứu này dựa trên các tiêu chí kỹ thuật và hiệu năng.

Bảng XI
SO SÁNH TỔNG QUAN HAI PHƯƠNG PHÁP TIẾP CẬN

Tiêu chí	**Transformer (Scratch)**	**Qwen2.5 (QLoRA)**
Kiến trúc	Encoder-Decoder (Custom)	Decoder-only (Pre-trained)
Số tham số	∼22 Triệu	1.5 Tỷ
Dữ liệu train	IWSLT (∼133k câu)	VLSP (∼475k câu)
Tokenizer	BPE (Custom trained)	Qwen Tokenizer (Pre-trained)
Thời gian train	∼3 giờ (25 epochs)	∼45 phút (1 epoch)
BLEU (En-Vi)	16.18	32.53
Ưu điểm	Hiểu sâu cơ chế, nhẹ, dễ triển khai	Hiểu ngữ nghĩa sâu, dịch mượt mà, tri thức rộng
Nhược điểm	Khó hội tụ, cần nhiều dữ liệu, kém ngữ nghĩa	Tốn VRAM, khó kiểm soát output (hallucination)

Sự chênh lệch lớn về BLEU (gần gấp đôi) cho thấy sức mạnh của việc "đứng trên vai người khổng lồ". Mô hình Transformer tự xây dựng học mọi thứ từ con số 0, trong khi Qwen2.5 đã tích lũy tri thức từ hàng nghìn tỷ token trước khi được tinh chỉnh.

VII. KẾT LUẬN

Dự án nghiên cứu này đã hoàn thành xuất sắc hai mục tiêu kép đầy tham vọng: tái hiện lại kiến trúc nền tảng của NLP hiện đại và ứng dụng những tiến bộ mới nhất của LLM vào bài toán thực tế.

Thông qua 6 giai đoạn phát triển mô hình Transformer từ đầu, nhóm đã không chỉ đạt được kết quả định lượng khả quan (BLEU 16.18) mà còn nắm bắt tường tận cơ chế hoạt động nội tại của mô hình, từ dòng chảy của gradient đến tác động của từng siêu tham số. Những kiến thức nền tảng này là vô giá, tạo tiền đề vững chắc cho việc tiếp cận các mô hình phức tạp hơn.

Song song đó, việc giải quyết bài toán dịch thuật Y tế VLSP 2025 bằng kỹ thuật QLoRA trên Qwen2.5-1.5B đã minh chứng cho sức mạnh của

phương pháp tinh chỉnh hiệu quả tham số. Kết quả BLEU 32.53 trên chiều Anh-Việt không chỉ là một con số ấn tượng về mặt học thuật mà còn cho thấy tiềm năng ứng dụng thực tiễn to lớn trong việc hỗ trợ các y bác sĩ tiếp cận tài liệu y khoa quốc tế, góp phần xóa bỏ rào cản ngôn ngữ trong y tế.

Tổng kết lại, đồ án này là sự kết hợp hài hòa giữa lý thuyết kinh điển và công nghệ tiên phong, khẳng định tính khả thi của việc xây dựng các hệ thống AI chất lượng cao ngay cả trong điều kiện tài nguyên hạn chế, miễn là có chiến lược dữ liệu và phương pháp huấn luyện phù hợp.

PHỤ LỤC A
CẤU HÌNH HYPERPARAMETERS CHI TIẾT

Để đảm bảo tính tái lập của nghiên cứu, chúng tôi liệt kê chi tiết các siêu tham số đã sử dụng cho mô hình Transformer tốt nhất (Phase 6).

Bảng XII
SIÊU THAM SỐ TRANSFORMER (FROM SCRATCH)

Tham số	**Giá trị**
Embedding Dimension (d_{model})	256
Number of Heads (h)	8
Number of Layers (N)	3
Feed Forward Dimension (d_{ff})	512
Dropout Rate	0.3
Warmup Steps	4000
Batch Size	32
Label Smoothing (ϵ)	0.1
Beam Width	3

PHỤ LỤC B
MÃ NGUỒN CÀI ĐẶT ATTENTION

Dưới đây là đoạn mã PyTorch cốt lõi thực hiện cơ chế Scaled Dot-Product Attention:

```
def attention(query, key, value, mask=None):
    d_k = query.size(-1)
    # Tinh diem attention (scores)
    scores = torch.matmul(
        query, key.transpose(-2, -1)
    ) / math.sqrt(d_k)

    if mask is not None:
        scores = scores.masked_fill(mask == 0,
            -1e9)

    p_attn = F.softmax(scores, dim=-1)
    return torch.matmul(p_attn, value), p_attn
```


LỜI CẢM ƠN

Để hoàn thành đồ án này, nhóm nghiên cứu đã nhận được sự hỗ trợ và động viên to lớn từ nhiều phía. Đầu tiên, chúng tôi xin gửi lời tri ân sâu sắc đến giảng viên hướng dẫn môn Xử lý Ngôn ngữ Tự nhiên, người đã không chỉ truyền đạt kiến thức nền tảng mà còn định hướng tư duy giải quyết vấn đề và cung cấp bộ dữ liệu VLSP 2025 quý giá.

Chúng tôi cũng xin gửi lời cảm ơn chân thành đến cộng đồng mã nguồn mở và các nhà nghiên cứu đi trước. Sự phát triển của các thư viện như PyTorch, Hugging Face Transformers, PEFT, và các công trình nghiên cứu nền tảng (như bài báo "Attention Is All You Need", "QLoRA") chính là những viên gạch vững chắc giúp chúng tôi hiện thực hóa ý tưởng của mình. Cuối cùng, xin cảm ơn sự nỗ lực không ngừng nghỉ của tất cả các thành viên trong nhóm, những người đã cùng nhau vượt qua hàng giờ debug căng thẳng để đi đến kết quả cuối cùng này.

TÀI NGUYÊN DỰ ÁN

Với tinh thần mã nguồn mở và mong muốn đóng góp cho cộng đồng nghiên cứu NLP tại Việt Nam, toàn bộ mã nguồn, dữ liệu tiền xử lý, file cấu hình huấn luyện và các checkpoint mô hình tốt nhất đều được chúng tôi công khai và lưu trữ đầy đủ.

Cộng đồng quan tâm có thể truy cập và tái lập kết quả tại repository GitHub chính thức của nhóm:

https://github.com/23020711/nlp_group5

TÀI LIỆU

[1] Vaswani, A., et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[2] Hu, E. J., et al. "LoRA: Low-Rank Adaptation of Large Language Models." International Conference on Learning Representations (2022).

[3] Dữ liệu IWSLT 2013 English-Vietnamese và VLSP 2025 Shared Task.

[4] Sutskever, I., et al. "Sequence to sequence learning with neural networks." Advances in neural information processing systems 27 (2014).

[5] Bahdanau, D., et al. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[6] Dettmers, T., et al. "QLoRA: Efficient Finetuning of Quantized LLMs." arXiv preprint arXiv:2305.14314 (2023).

[7] Touvron, H., et al. "LLaMA: Open and Efficient Foundation Language Models." arXiv preprint arXiv:2302.13971 (2023).

[8] Devlin, J., et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[9] Papineni, K., et al. "BLEU: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics (2002).

[10] Post, M. "A call for clarity in reporting BLEU scores." arXiv preprint arXiv:1804.08771 (2018).

[11] Brown, T., et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020).

[12] Radford, A., et al. "Improving language understanding by generative pre-training." (2018).

[13] Vaswani, A., et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[14] Devlin, J., et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[15] Sennrich, R., et al. "Neural machine translation of rare words with subword units." arXiv preprint arXiv:1508.07909 (2015).

[16] Koehn, P., et al. "Moses: Open source toolkit for statistical machine translation." Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions (2007).

[17] Bahdanau, D., et al. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[18] Gehring, J., et al. "Convolutional sequence to sequence learning." International conference on machine learning. PMLR (2017).

[19] Wu, Y., et al. "Google's neural machine translation system: Bridging the gap between human and machine translation." arXiv preprint arXiv:1609.08144 (2016).

[20] Ott, M., et al. "fairseq: A fast, extensible toolkit for sequence modeling." arXiv preprint arXiv:1904.01038 (2019).

[21] Wolf, T., et al. "Transformers: State-of-the-art natural language processing." Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations (2020).

[22] Liu, Y., et al. "RoBERTa: A robustly optimized BERT pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

[23] Zhang, T., et al. "BERTScore: Evaluating text generation with BERT." arXiv preprint arXiv:1904.09675 (2019).

[24] Reimers, N., and Gurevych, I. "Sentence-BERT: Sentence embeddings using Siamese BERT-networks." arXiv preprint arXiv:1908.10084 (2019).

[25] Johnson, M., et al. "Google's multilingual neural machine translation system: Enabling zero-shot translation." Transactions of the Association for Computational Linguistics 5 (2017).

[26] Conneau, A., et al. "Unsupervised cross-lingual representation learning at scale." arXiv preprint arXiv:1911.02116 (2019).

[27] Raffel, C., et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." The Journal of Machine Learning Research 21.1 (2020).

[28] He, K., et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition (2016).

[29] Kingma, D. P., and Ba, J. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[30] Srivastava, N., et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1 (2014).

[31] Sennrich, R., et al. "Neural machine translation of rare words with subword units." arXiv preprint arXiv:1508.07909 (2015).

[32] Loshchilov, I., and Hutter, F. "Decoupled weight decay regularization." arXiv preprint arXiv:1711.05101 (2017).