Báo cáo bài tập lớn môn xử lý ngôn ngữ tự nhiên

Nguyễn Văn Hòa
23021556@vnu.edu.vn

Nguyễn Bích Đạt
23021520@vnu.edu.vn

Nguyễn Thế Hùng
23021568@vnu.edu.vn

Lê Văn Tuấn
23021705@vnu.edu.vn

Tóm tắt nội dung

Hiện nay, kiến trúc Transformer với cơ chế self-attention đã chứng minh hiệu năng vượt trội so với các mô hình tuần tự truyền thống như RNN hay LSTM, trở thành tiêu chuẩn mới trong việc xử lý các tác vụ ngôn ngữ tự nhiên phức tạp. Báo cáo này trình bày một nghiên cứu toàn diện nhằm tái hiện, đánh giá và tối ưu hóa kiến trúc Transformer cho cặp ngôn ngữ Anh-Việt, được chia thành bài chính.

Trong Bài 1, nghiên cứu tập trung vào việc xây dựng mô hình Transformer từ nền tảng để phân tích sâu sắc tác động của các cơ chế attention khác nhau lên hiệu năng dịch thuật. Thông qua các thực nghiệm có hệ thống trên bộ dữ liệu PhoMT với quy mô từ 100 nghìn đến 1 triệu cặp câu, chúng tôi so sánh hiệu quả giữa Scaled Dot-Product Attention và Luong Attention. Kết quả thực nghiệm chỉ ra rằng Scaled Dot-Product Attention mang lại độ ổn định và điểm số BLEU và COMET cao hơn đáng kể. Đồng thời, nghiên cứu cũng khẳng định ưu thế của chiến lược giải mã Beam Search so với Greedy Search trong việc cải thiện tính mạch lạc của bản dịch.

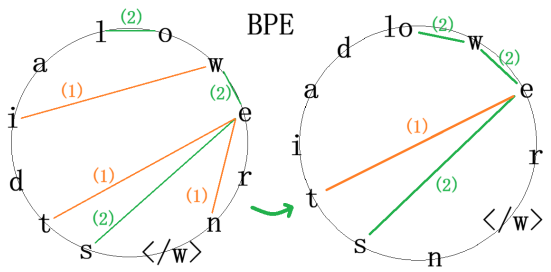
Trong khi đó, Bài 2 tập trung nghiên cứu giải quyết thách thức về sự khan hiếm dữ liệu và rào cản thuật ngữ trong dịch máy chuyên ngành y tế. Chúng tôi đề xuất áp dụng phương pháp thích nghi miền (Domain-Adaptive Pretraining) thông qua việc huấn luyện trung gian mô hình pre-trained (Helsinki-NLP/opus-mt-en-vi) trên tập dữ liệu y sinh lớn ViPubMed trước khi fine-tune cho nhiệm vụ cụ thể trên tập dữ liệu VLSP Shared Task 2025. Các kết quả định lượng bằng BLEU và COMET đã chứng minh rằng việc tiếp cận tri thức miền thông qua dữ liệu trung gian giúp mô hình cải thiện vượt bậc các chỉ số đánh giá. Code của nhóm để ở link git: https://github.com/nvhoa2005/ProjectNaturalLanguageProcessing

1 Introduction

Dịch máy là một trong những bài toán trong lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP). Mục tiêu của dịch máy là tự động chuyển đổi văn bản từ ngôn ngữ nguồn sang ngôn ngữ đích sao cho giữ nguyên nghĩa, đảm bảo tính mạch lạc, sự liên kết ngữ pháp, và phù hợp với ngữ cảnh của văn bản. Đây là một nhiệm vụ khó khăn bởi sự phức tạp và đa dạng trong ngôn ngữ tự nhiên, bao gồm các hiện tượng như đồng nghĩa, đa nghĩa, cấu trúc câu phức tạp, cũng như những khác biệt về cú pháp và ngữ pháp giữa các ngôn ngữ.

Trải qua nhiều thế hệ mô hình, từ các phương pháp thống kê truyền thống như Statistical Machine Translation (SMT) (Brown et al., 1993) cho đến các mô hình học sâu dựa trên mạng neural như Recurrent Neural Networks (RNN) và Long Short-Term Memory (LSTM) (Mienye et al., 2024), bài toán dịch máy vẫn gặp nhiều hạn chế. RNN và LSTM mặc dù có khả năng xử lý dữ liệu tuần tự và mô hình hóa phụ thuộc ngữ cảnh, nhưng chúng vẫn gặp khó khăn trong việc ghi nhớ thông tin dài hạn, dẫn đến giảm chất lượng dịch ở những câu văn dài hoặc phức tạp. Đồng thời, các mô hình này thường yêu cầu thời gian huấn luyện dài và khả năng tính toán tuần tự làm hạn chế hiệu suất khi triển khai trên tập dữ liệu lớn.

Sự xuất hiện của kiến trúc Transformer (Vaswani et al., 2017) đã đánh dấu một bước ngoặt quan trọng trong lĩnh vực dịch máy. Với cơ chế self-attention, Transformer có khả năng học được các mối quan hệ giữa tất cả các từ trong câu một cách song song, giúp mô hình nắm bắt ngữ cảnh dài hạn hiệu quả hơn nhiều so với các mô hình tuần tự trước đây. Ngoài ra, khả năng huấn luyện song song của Transformer đã cải thiện đáng kể tốc độ huấn luyện và cho phép mở rộng quy mô trên các tập dữ liệu lớn, tạo tiền đề cho sự phát triển của các mô hình dịch máy hiện đại như BERT (Devlin et al., 2019) hay mBART(Liu et al., 2020).



Hình 1: Minh hoạt phương pháp BPE với bộ corpus bao gồm: l o w </w>; l o w e r </w>; n e w e s t </w>; w i d e s t </w>; e a t </w>

Trong báo cáo này, nhóm đã nghiên cứu hai phần chính bao gồm:

- Xây dựng mô hình Transformer từ đầu: Tái hiện kiến trúc cơ bản và quy trình huấn luyện của Transformer nhằm hiểu sâu cơ chế self-attention, cơ chế multi-head attention, cũng như ảnh hưởng của các tham số siêu cấu hình đến chất lượng dịch. Nhóm cũng tiến hành thí nghiệm so sánh các cơ chế attention khác nhau để phân tích hiệu quả từng phương pháp.

- Ứng dụng mô hình Transformer vào fine-tuning trên bộ dữ liệu VLSP 2025 Shared Task Machine Translation: Nhóm sử dụng mô hình đã huấn luyện trước và tinh chỉnh trên bộ dữ liệu VLSP 2025 để đánh giá hiệu quả mô hình trong bối cảnh thực tế.

Qua các bước trên, báo cáo không chỉ cung cấp cái nhìn toàn diện về kiến trúc Transformer mà còn minh họa quá trình triển khai mô hình dịch máy hiện đại, từ việc hiểu lý thuyết cho đến ứng dụng thực tiễn.

2 Báo cáo bài 1

Trong phần này, chúng tôi sẽ nói chi tiết về các module trong code transformer, cùng với kết quả thử nghiệm các loại self-attention cùng các loại decode trên bộ dữ liệu PhoMT:

2.1 Các module chính

Kiến trúc Transformer được cấu thành từ nhiều module chức năng phối hợp chặt chẽ với nhau, mỗi module đảm nhiệm một vai trò riêng trong việc học và biểu diễn ngữ cảnh của chuỗi dữ liệu. Phần này trình bày các module chính của Transformer và nguyên lý hoạt động của chúng.

Input Embedding Dựa trên tập huấn luyện vốn có (tập hợp các câu), với mỗi câu bài báo gốc tách thành các từ và các từ tách thành các chữ. Tập hợp các chữ tách được từ tập dữ liệu huấn luyện tạo thành một bộ chữ cái. Sau đó, bằng cách lặp lại quy trình: xét các từ trong một câu (đã được thêm ký tự kết thúc từ), đếm các cặp chữ gần nhau hay xuất hiện, bài báo thu được một tập các cặp chữ, ghép cặp từ có tần suất xuất hiện lớn nhất lại với nhau, ta thu được một token thô. Tập chữ cái ban đầu sau khi cập nhật (nối 2 chữ cái có tần suất xuất hiện lớn nhất lại với nhau) sẽ tiếp tục lặp lại quy trình trên, cuối cùng ta thu được một bộ từ điển gồm rất nhiều token. Trong bài báo gốc, đây được gọi là kỹ thuật BPE (Byte Pair Encoding).

Module Input Embedding tiếp nhận các token trong từ vựng được xây dựng bằng kỹ thuật BPE. Trong giai đoạn huấn luyện, các câu văn bản được phân tách thành các token, sau đó toàn bộ tập token này được tổng hợp để hình thành một từ điển token, trong đó mỗi token được ánh xạ duy nhất sang một chỉ số nguyên (token ID). Nhờ đó, mỗi câu trong tập huấn luyện trước khi đưa vào mô hình được biểu diễn dưới dạng một dãy các token ID.

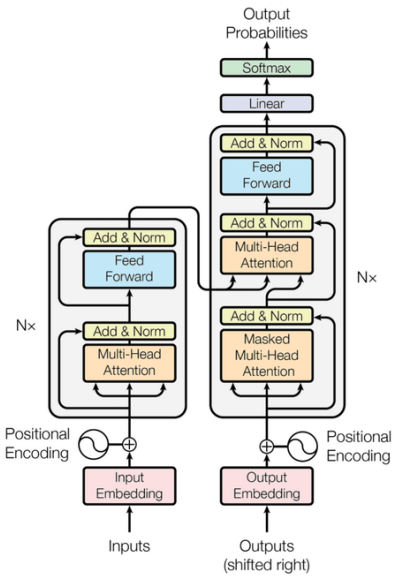
Các token ID này được ánh xạ thông qua ma trận embedding để tạo ra các biểu diễn vector liên tục của từ, trong đó mỗi token ID đóng vai trò tương đương với một biểu diễn one-hot ở không gian chỉ số. Các vector embedding thu được sau đó được chuẩn hoá bằng cách nhân với căn bậc hai của chiều không gian biểu diễn d_{model} rồi cộng trực tiếp với vector mã hoá vị trí (positional embedding) nhằm cung cấp thông tin về thứ tự của các token trong chuỗi. Ma trận positional embedding có cùng chiều d_{model} với embedding từ, cho phép hai thành phần này được kết hợp một cách nhất quán.

Positional Encoding được thiết kế nhằm cung cấp thông tin về thứ tự và vị trí của các token trong chuỗi, bù đắp cho việc kiến trúc Transformer không sử dụng các cơ chế xử lý tuần tự như RNN hay CNN. Bộ mã hoá vị trí này tạo ra các biểu diễn vị trí cố định (không học được) và được cộng trực tiếp vào embedding của token. Cụ thể , vector mã hoá vị trí được xác định theo các hàm sin và cos như sau:

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

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

trong đó pos là vị trí của token trong câu, i là



Hình 2: Kiến trúc Transformer

chỉ số chiều của vector mã hoá vị trí; Các chiều chẵn$(2i)$ sử dụng hàm sin, các chiều lẻ $(2i+1)$ sử dụng hàm cos

Thiết kế này cho phép mô hình suy luận được quan hệ vị trí tương đối giữa các token thông qua các tổ hợp tuyến tính của các vector mã hoá vị trí.

Encoder đầu ra của 2 khối trên là một ma trận X với $X \in R^{T \times d_{model}}$ (T là số chiều của chuỗi). Ma trận X này sau đó sẽ được biến đổi lần lượt qua các khối thành phần trong kiến trúc Transformer.

Ma trận X trước tiên được biến đổi bởi một head. Đặc trưng cho một head là bộ ba ma trận W_K, W_Q và W_V có kích cỡ 512x512. Sau khi nhân X với các ma trận trên, ta thu được các dạng tồn tại mới của X như sau:

$$Q = X \times W_Q \tag{3}$$

$$K = X \times W_K \tag{4}$$

$$V = X \times W_V \tag{5}$$

các ma trận Q, K và V đều có số chiều là T x 512, thực chất là dạng tồn tại mới của X trong 3 không gian Query - Key - Value. Không gian Query được hiểu là không gian biểu diễn một phương diện ngữ nghĩa của X, Key là không gian biểu diễn đáp ứng của X với phương diện ngữ nghĩa nói trên, Value là không gian biểu diễn giá trị cố hữu của X. Vì X là tập hợp biểu diễn của các token trong một câu nên đồng nghĩa Query, Key và Value đang biểu diễn phương diện ngữ nghĩa của token, đáp ứng của token và giá trị cố hữu của chúng.

Trong từng head, cơ chế attention được tính độc lập theo công thức:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

phép nhân ma trận trong biểu thức trên $(Q_i K_i^T)$ đại diện cho việc lấy tích vô hướng (dot-product) biểu diễn của một token trong chiều không gian Query với tất cả biểu diễn của token khác (bao gồm chính nó) trong chiều không gian Key. Công thức tích vô hướng $(uv = |u||v|Cos(u,v))$ cho biết mức độ đồng điệu của 2 vector trong một không gian nói chung, điều đồng nghĩa với việc token được xét đang đi tìm bổ từ cho nó thông qua định lượng $Q_i \times K_i^T$. Bằng việc chuẩn hoá chia tích trên cho $\sqrt{d}$, tích chấm sẽ không tạo ra giá trị quá lớn khi số chiều cao, nhờ đó Softmax không bị bão hoà (Gradient ổn định hơn). Đầu ra của hàm Softmax được nhân với (V_i) biểu diễn giá trị thực của token so với giá trị cố hữu của nó.

Thực tế kiến trúc Transformer không được thiết kế như vậy, chúng ta sẽ có nhiều head thay vì một, mỗi head được đặc trưng bởi bộ ba Q_i, K_i và V_i riêng biệt có kích cỡ bằng nhau và bằng 512 x 64. Việc tạo ra nhiều head thay vì 1 giống như chiếu X trên các không gian ngữ nghĩa khác nhau, trên các không gian đáp ứng ngữ nghĩa khác nhau đi cùng với các giá trị ngữ nghĩa cố hữu khác nhau. Việc tính toán attention sẽ được thực hiện theo head, không có sự pha trộn. Tập hợp các head riêng biệt nói trên tạo thành một khối multi-head, lúc này công thức Attention được tính như sau:

$$\text{Atten}(Q_i, K_i, V_i) = \text{Softmax}(\frac{Q_i K_i^T}{\sqrt{d_k}})V_i \tag{7}$$

trong đó i đại diện cho các head khác nhau (trong bài báo gốc $i = 8$) và d_k lúc này được hiểu là số chiều của vector Key (trong bài báo gốc $d_k = 64$).

Sau khi các head hoàn tất việc tính toán attention, đầu ra của chúng được nối lại tạo thành một ma trận có kích cỡ T x 512, bằng với kích cỡ của ma trận X ban đầu nhưng giàu thông tin hơn do được biến đổi bởi các head.

Add & LayerNorm (sau Attention): đầu ra của khối multi-head nói trên sẽ được cộng với X bản thể thông qua residual (Add), theo sau đó là LayerNorm với mục đích giúp mô hình giữ Gradient ổn định và huấn luyện sâu. Cụ thể:

$$\text{LayerNorm}(x) = \gamma\frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{8}$$

(ở đây x là một hàng trong X). Công thức trên không nằm trong bài báo gốc mà là phục dựng theo mô tả của bài báo gốc. Trong đó x là biểu diễn của các token (sau khi đi qua multi-head); μ, σ^2 lần lượt là trùng bình và phương sai của giá trị các chiều trong vector x, ϵ là bias và cặp γ, β là các tham số học được. Trước khi qua LayerNorm, đầu ra của multi-head được cộng với X trước khi đi qua multi-head:

$$x' = \text{LayerNorm}(x + \text{Attention}(x)) \quad (9)$$

đầu ra của LayerNorm sau đó tiếp tục được đưa qua một mạng **Feed Forward Network** (FFN):

$$\text{FFN}(x) = [\text{Max}(0, xW_1 + b_1)]W_2 + b2 \quad (10)$$

$$W_1 \in \mathbb{R}^{d_{model} \times d_{hidden}}, b_1 \in \mathbb{R}^{d_{hidden}} \quad (11)$$

$$W_2 \in \mathbb{R}^{d_{hidden} \times d_{model}}, b_2 \in \mathbb{R}^{d_{model}} \quad (12)$$

trong đó W_1 và W_2 là ma trận đại diện cho một mạng nơ-ron đa lớp. Số chiều của mạng trên được ký hiệu bởi d_{hidden}, lớn hơn d_{model}. Hàm Max trong công thức biểu diễn cho lớp phi tuyến Relu nằm giữa 2 lớp tuyến tính nói trên. Trước khi được đưa vào các khối Decoder, đầu ra của FFN tiếp tục được đưa qua Add & LayerNorm.

Toàn bộ các thành phần trên cấu tạo thành một Encoder, trong bài báo gốc, tác giả sử dụng 6 Encoder nối tiếp nhau, trong đó output của Encoder này đóng vai trò là input của Encoder kia.

Decoder tương đối giống với Encoder, chỉ khác khối này có thêm masked multi-head attention và câu đích được dịch phải 1 đơn vị. Masked multi-head attention giống với multi-head attention của Encoder nhưng trước khi đi qua SoftMax, ma trận trung gian X (X của câu đích - groundtruth) được cộng với ma trận tam giác dưới có các phần tử nằm trên đường chéo chính có giá trị âm vô cực:

$$G_i = \frac{Q_i K_i^T}{\sqrt{d_k}} + mask \quad (13)$$

$$\text{mAttn}(Q_i, K_i, V_i) = \text{Softmax}(G_i)V_i \quad (14)$$

kết quả sau khi đi qua Softmax ta thu được trọng số attention cho token tương lai bằng 0.

Trong hình mình hoạ kiến trúc Tranformer (hình 2), mũi tên đầu ra của Encoder gắn với multi-head của Decoder ám chỉ việc khối này cung cấp cho multi-head của Decoder 2 ma trận K, V với ý nghĩa tương tự giải thích trên. Gọi E là mà trận đầu ra của Encoder, đầu vào của Decoder nhận vào từ Encoder được tính như sau:

$$K_{enc} = E \times W_K \quad (15)$$

$$V_{enc} = E \times W_V \quad (16)$$

trong đó W_K và W_V là các ma trận của Decoder có cùng ý nghĩa với các ma trận tương ứng trong Encoder. Ma trận còn lại W_Q được nhân với X_{dec} - tương ứng với X của Encoder nhưng trích xuất từ câu ground truth (câu dịch tương ứng, dịch sang phải 1 đơn vị chữ). X_{dec} là sản phẩm của Masked Multi-Head Attention và Add & LayerNorm. Tại khối Multi-Head Attention của Decoder, sau khi tập hợp đủ Q_{dec} (từ Decoder), K_{enc} và V_{enc} (từ Encoder), mô hình tiếp tục thực hiện Attention:

$$\begin{aligned} &\text{Attention}(Q_{dec}, K_{enc}, V_{enc}) \\ &= \text{Softmax}\left(\frac{Q_{dec}K_{enc}^\top}{\sqrt{d_k}}\right)V_{enc} \end{aligned} \quad (17)$$

Công thức trên ám chỉ việc Decoder lấy thông tin ngữ cảnh từ Encoder để tổng hợp và đưa ra dự đoán.

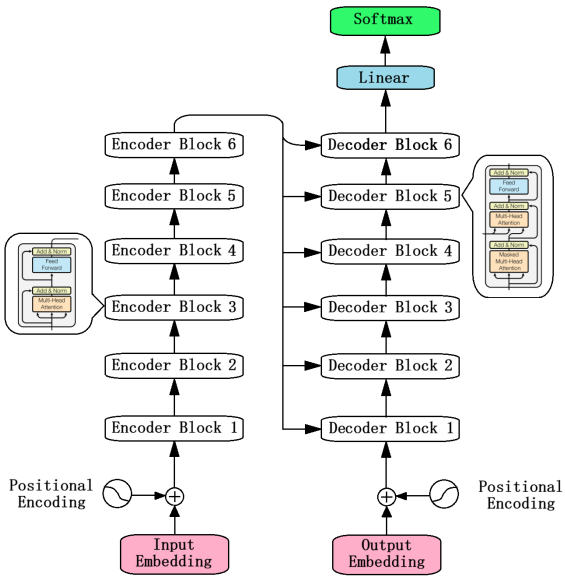
Giả sử đầu ra của Decoder là H (Tx512), một lớp Linear sẽ được áp dụng lên H nhằm chiếu các phần tử trong H (các hàng) lên không gian |V| chiều với V là tập từ vựng, theo sau đó là Softmax nhằm biến đổi giá trị của các perceptron trong mạng linear từ miền liên tục thành miền xác suất, đại diện cho dự đoán của mô hình về từ tiếp theo nó có thể sinh.

$$\begin{aligned} \text{Output} &= \text{Softmax}(HW_0 + b_0), \\ &\text{trong đó } W_0 \in \mathbb{R}^{512 \times |V|} \end{aligned} \quad (18)$$

Trong bài báo gốc, W_0 được dùng chung với ma trận encoding đầu vào để giảm số lượng tham số)

Bên cạnh kiến trúc Transformer chuẩn, mô hình trong nghiên cứu này sử dụng biến thể Pre-Layer Normalization (Prenorm) nhằm cải thiện tính ổn định trong quá trình huấn luyện. Khác với Transformer gốc, trong đó Layer Normalization được đặt sau phép cộng residual (Post-Norm), kiến trúc Prenorm áp dụng Layer Normalization trước mỗi khối con (sublayer), bao gồm self-attention và feed-forward network, trong khi vẫn giữ nguyên cấu trúc residual connection. Cụ thể, khác với thiết kế của bản gốc (công thức (9)), với đầu vào x, khối con trong Prenorm Transformer được biểu diễn dưới dạng:

$$x' = x + \text{SubLayer}(\text{LayerNorm}(x)) \quad (19)$$



Hình 3: Minh hoạ Stack Encoder - Decoder

Việc đưa Layer Normalization lên trước khối xử lý cho phép gradient truyền trực tiếp qua nhánh residual mà không bị biến đổi bởi phép chuẩn hóa, từ đó giúp mô hình huấn luyện ổn định hơn, đặc biệt khi số lượng tầng tăng. Do đầu ra của mỗi block trong kiến trúc Prenorm không được chuẩn hóa một cách tường minh, mô hình thường bổ sung một Layer Normalization cuối cùng sau block Transformer cuối để ổn định phân phối biểu diễn trước khi đưa vào các tầng dự đoán.

Biến thể Pre-Layer Normalization mang lại ưu điểm chính là cải thiện độ ổn định của quá trình huấn luyện, nhờ việc gradient có thể truyền trực tiếp qua các nhánh residual mà không bị ảnh hưởng bởi phép chuẩn hóa. Điều này đặc biệt hiệu quả đối với các mô hình Transformer sâu, giúp giảm phụ thuộc vào kỹ thuật warm-up learning rate và cho phép sử dụng tốc độ học lớn hơn. Tuy nhiên, do Layer Normalization không còn được áp dụng sau mỗi block, biểu diễn đầu ra của các tầng trung gian không được chuẩn hóa tường minh, dẫn đến việc mô hình thường phải bổ sung một LayerNorm cuối cùng để ổn định phân phối đầu ra. Ngoài ra, trong các kiến trúc Transformer nông, lợi ích của Prenorm so với Post-Norm có thể không rõ rệt.

2.2 Huấn luyện mô hình

Mô hình được thiết lập một kiến trúc Transformer tiêu chuẩn với các tham số mô hình được lựa chọn nhằm cân bằng giữa khả năng biểu diễn ngữ nghĩa và chi phí tính toán. Cụ thể, mô hình bao gồm 6 lớp encoder và 6 lớp decoder ($N = 6$), với kích thước vector ẩn d_{model} là 512 và số lượng đầu attention (h) là 8. Mạng FFN bên trong mỗi khối được thiết lập với kích thước $d_{ff} = 2048$, và kỹ thuật Dropout với tỷ lệ 0.1 được áp dụng tại các lớp con để hạn chế hiện tượng overfitting.

Quá trình tối ưu hóa tham số được thực hiện thông qua thuật toán Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$), với tốc độ học được cố định ở mức $\alpha = 10^{-4}$. Hàm mất mát được sử dụng là Cross-Entropy Loss, trong đó chúng tôi áp dụng cơ chế Label Smoothing hoặc loại bỏ padding index khỏi quá trình tính toán loss để mô hình tập trung vào các từ vựng thực tế. Chu trình huấn luyện kéo dài trong 5 epoch với batch size là 32. Tại mỗi bước huấn luyện, gradient được tính toán và lan truyền ngược để cập nhật trọng số nhằm giảm thiểu sai số dự đoán. Chúng tôi cũng theo dõi biến thiên của hàm mất mát sau mỗi epoch để đảm bảo mô hình đang hội tụ đúng hướng, đồng thời ghi nhận thời gian huấn luyện trung bình cho mỗi epoch nhằm đánh giá hiệu quả tính toán của thuật toán.

2.3 Kết quả đánh giá mô hình

Trong phần này, nhóm trình bày và phân tích kết quả đánh giá mô hình Transformer trên bộ dữ liệu PhoMT (Doan et al., 2021), đồng thời so sánh hiệu năng của mô hình khi sử dụng các cơ chế self-attention và chiến lược giải mã (decoding) khác nhau. Toàn bộ thí nghiệm được thực hiện trên mô trường máy ảo Kaggle với GPU T4x2 (16GB VRAM) và 30GB RAM, nhằm đảm bảo điều kiện tính toán đồng nhất giữa các cấu hình thử nghiệm.

Bộ dữ liệu Nhóm đã sử dụng bộ dữ liệu PhoMT, bao gồm 3.02 triệu cặp câu song ngữ Anh–Việt, là một trong những bộ dữ liệu dịch máy quy mô lớn và chất lượng cao cho tiếng Việt. Tuy nhiên, để khảo sát ảnh hưởng của kích thước dữ liệu huấn luyện đến hiệu năng của mô hình, nhóm không sử dụng toàn bộ tập dữ liệu mà tiến hành thí nghiệm với các tập con có kích thước tăng dần, từ 100 nghìn đến 1 triệu cặp câu.

Cách tiếp cận này cho phép đánh giá khả năng mở rộng của mô hình Transformer cũng như quan sát hiện tượng bão hòa dữ liệu, tức là khi việc bổ sung thêm dữ liệu huấn luyện không còn mang lại cải thiện đáng kể về chất lượng dịch.

Độ đo Để đánh giá chất lượng dịch của mô hình, nhóm đã sử dụng 2 độ đo phổ biến là BLEU (Bilingual Evaluation Understudy) (Papineni et al., 2002) và COMET (Crosslingual Optimized Metric for Evaluation of Translation) (Rei et al., 2020).

BLEU là chỉ số này đo lường mức độ tương đồng giữa câu dịch tự động và các câu tham chiếu thông qua việc so sánh các n-gram, được thể hiện bằng công thức:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \qquad (20)$$

trong đó p_n là độ chính xác (precision) của n-gram thứ n, w_n là trọng số của từng n-gram (thường $w_n = \frac{1}{N}$), và BP là brevity penalty được định nghĩa như sau:

$$\text{BP} = \begin{cases} 1 & \text{nếu } c > r \\ e^{1-r/c} & \text{nếu } c \leq r \end{cases} \qquad (21)$$

với c là độ dài câu dịch và r là độ dài câu tham chiếu. BLEU đánh giá chủ yếu dựa trên trùng khớp n-gram bề mặt mà chưa xét đến nghĩa sâu của câu.

COMET là độ đo dựa trên mô hình học sâu, được huấn luyện để dự đoán chất lượng dịch máy tương quan với đánh giá của con người. Khác với BLEU, COMET đánh giá ngữ nghĩa và ngữ cảnh của câu, dựa trên embeddings đa ngôn ngữ. Mô hình dự đoán điểm chất lượng $\hat{y}$ như sau:

$$\hat{y} = f_\theta(\text{Embed}(x), \text{Embed}(r)), \qquad (22)$$

trong đó $\text{Embed}(\cdot)$ là vector biểu diễn ngữ nghĩa của câu dịch x và câu tham chiếu r, còn f_θ là mô hình học sâu (MLP hoặc transformer) được huấn luyện trên dữ liệu đánh giá dịch máy. Nhờ đánh giá ở mức ý nghĩa, COMET có tương quan cao hơn với đánh giá của con người so với các metric truyền thống như BLEU.

Kết quả của mô hình Kết quả tổng thể của mô hình Transformer với hai cơ chế self-attention khác nhau – Scaled Dot-Product Attention và Luong Attention – được trình bày trong Bảng 1. Qua bảng kết quả có thể nhận thấy rằng khi kích thước tập dữ liệu huấn luyện tăng, hiệu năng của mô hình cải thiện rõ rệt ở giai đoạn đầu, đặc biệt từ 100 nghìn đến khoảng 700 nghìn cặp câu. Tuy nhiên, khi số lượng dữ liệu tiếp tục tăng lên, mức cải thiện trở nên chậm hơn và không còn ổn định, cho thấy mô hình bắt đầu đạt trạng thái bão hòa.

Đối với Scaled Dot-Product Attention, mô hình đạt điểm BLEU cao nhất là 30.86 và COMET là 0.81 khi huấn luyện trên 900 nghìn cặp câu. Trong khi đó, Luong Attention cũng cho kết quả cạnh tranh, với điểm BLEU cao nhất đạt 30.86 và COMET đạt 0.81 ở cùng quy mô dữ liệu. Nhìn chung, Scaled Dot-Product Attention cho kết quả ổn định hơn ở hầu hết các mốc dữ liệu, đặc biệt khi kích thước tập huấn luyện lớn, phản ánh sự phù hợp của cơ chế này với kiến trúc Transformer.

Phân tích quá trình hội tụ Quan sát đồ thị biến thiên của độ đo BLEU và COMET qua 5 epoch huấn luyện tại Hình 4, chúng tôi nhận thấy cả hai cơ chế attention đều thể hiện xu hướng hội tụ mạnh mẽ ngay từ giai đoạn đầu. Cụ thể, trong hai epoch đầu tiên, độ dốc lớn của đường biểu diễn minh chứng cho tốc độ học nhanh, khẳng định hiệu quả của chiến lược learning rate warmup trong việc giúp mô hình nhanh chóng thoát khỏi các vùng tham số khởi tạo ngẫu nhiên để hướng tới điểm tối ưu. Bước sang giai đoạn từ epoch thứ 3 đến thứ 5, đà tăng trưởng của các chỉ số bắt đầu chậm lại và đi vào ổn định, xác nhận việc dừng huấn luyện tại epoch 5 là quyết định tối ưu nhằm cân bằng giữa hiệu năng dịch thuật và chi phí tính toán, đồng thời tránh hiện tượng overfitting khi mô hình đã đạt đến trạng thái bão hòa trên tập dữ liệu huấn luyện.

Khi đi sâu vào so sánh giữa hai biến thể, cơ chế Scaled Dot-Product Attention thể hiện xu hướng ổn định hơn trên nhiều mốc so với Luong Attention trên cả hai phương diện hiệu năng tuyệt đối và độ ổn định. Không chỉ duy trì đường cong hiệu năng nằm ở vị trí cao hơn trên hầu hết các epoch, biến thể Scaled Dot-Product còn cho thấy quá trình tối ưu hóa mượt mà hơn, hạn chế được các dao động thất thường (fluctuations) trong quá trình lan truyền ngược mà Luong Attention thường gặp phải. Sự ổn định này gợi ý rằng việc chuẩn hóa tích vô hướng trong không gian nhiều chiều thực sự giúp gradient lan truyền hiệu quả hơn, dẫn đến kết quả hội tụ tốt hơn cho kiến trúc Transformer trên bộ dữ liệu PhoMT.

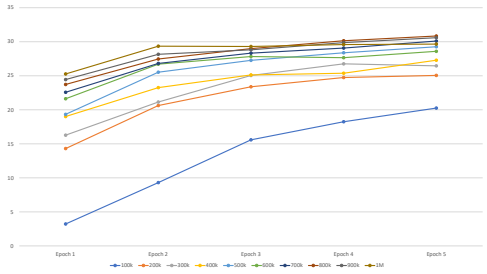
Đánh giá các thuật toán giải mã (Decoding) cho mô hình Transformer Bên cạnh việc so sánh các cơ chế self-attention, nhóm cũng tiến hành đánh giá ảnh hưởng của các chiến lược giải mã đến chất lượng dịch của mô hình Transformer. Kết quả so sánh giữa Greedy Search và Beam Search với các kích thước chùm khác nhau được trình bày trong Bảng 2. Do mô hình đã đạt trạng thái bão hòa tại mốc 700 nghìn cặp câu, các thí nghiệm giải mã chỉ được thực hiện trên tập dữ liệu này để đảm bảo tính công bằng và tiết kiệm chi phí tính toán. Một số mẫu dịch cụ thể của mô hình có thể xem ở phần Appendix A

Kết quả cho thấy Beam Search tốt hơn Greedy Search trên cả hai độ đo BLEU và COMET. Cụ thể,

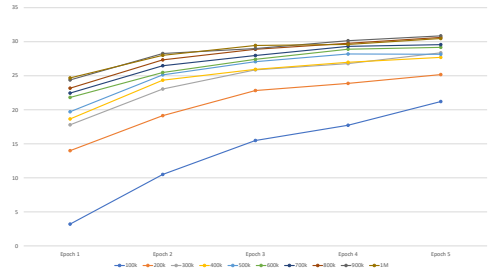
Bảng 1: Kết quả của mô hình Transformer trên bộ dữ liệu PhoMT

	Scaled Dot Product Attention		**Luong Attention**	
Dataset	**BLEU**	**COMET**	**BLEU**	**COMET**
PhoMT (100k)	20.25	0.66	21.21	0.67
PhoMT (200k)	25.05	0.74	25.17	0.73
PhoMT (300k)	26.75	0.74	28.37	0.74
PhoMT (400k)	27.28	0.75	27.70	0.76
PhoMT (500k)	29.26	0.78	28.20	0.78
PhoMT (600k)	28.59	0.79	29.17	0.79
PhoMT (700k)	30.10	0.80	29.57	0.79
PhoMT (800k)	30.84	0.80	30.63	0.80
PhoMT (900k)	30.59	**0.81**	**30.86**	**0.81**
PhoMT (1M)	29.65	0.80	30.47	0.80

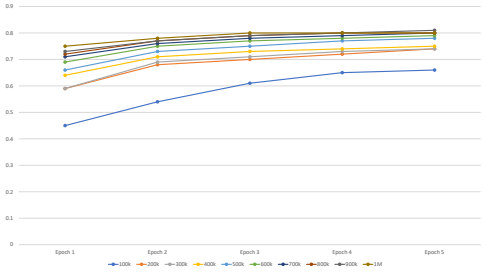
In đậm thể hiện điểm cao nhất



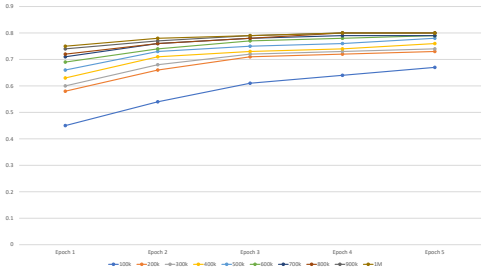
(a) Điểm BLEU trên từng epoch, sử dụng Dot-Product



(b) Điểm BLEU trên từng epoch, sử dụng Luong



(c) Điểm COMET trên từng epoch, sử dụng Dot-Product



(d) Điểm COMET trên từng epoch, sử dụng Luong

Hình 4: Phân tích điểm số đạt được của mô hình theo từng epoch

khi sử dụng Scaled Dot-Product Attention, điểm BLEU tăng từ 28.67 (Greedy Search) lên 29.58 với Beam Search (k = 2), và tiếp tục tăng khi kích thước chùm lớn hơn. Điểm số cao nhất đạt được tại Beam size = 5 với BLEU là 30.10 và COMET là 0.80. Tuy nhiên, mức cải thiện giữa các giá trị k lớn (từ k = 4 đến k = 5) là tương đối nhỏ, cho thấy hiệu năng đã tiệm cận ngưỡng bão hòa.

Xu hướng tương tự cũng được quan sát đối với Luong Attention. Mặc dù Beam Search tiếp tục mang lại cải thiện so với Greedy Search, sự khác biệt giữa các kích thước chùm lớn là không đáng kể. Điều này cho thấy việc lựa chọn Beam size cần cân nhắc giữa chất lượng dịch và chi phí tính toán, trong đó Beam size = 4 hoặc 5 có thể được xem là lựa chọn hợp lý trong bối cảnh thí nghiệm này.

3 Báo cáo bài 2

Các mô hình Transformer, mặc dù mạnh mẽ trong việc xử lý ngôn ngữ tự nhiên, vẫn gặp khó khăn khi đối mặt với các từ chuyên ngành y tế. Nguyên nhân chính là do dữ liệu huấn luyện của các mô hình này thường dựa trên ngôn ngữ phổ thông, thiếu các thuật ngữ chuyên sâu hoặc ngữ cảnh y học đặc thù. Khi gặp các từ như tên thuốc, triệu chứng hiếm gặp hay các khái niệm y học phức tạp, mô hình dễ hiểu sai ý nghĩa, gây ra lỗi trong dự đoán hoặc sinh văn bản. Do đó, để ứng dụng trong y tế, các mô hình Transformer cần được tinh chỉnh thêm với dữ liệu chuyên ngành hoặc kết hợp kiến thức từ các cơ sở dữ liệu y học đáng tin cậy. Trong phần này, nhóm sẽ ứng dụng và tối ưu hóa mô hình Transformer trên bộ dữ liệu miền y tế.

3.1 Bộ dữ liệu

Bộ dữ liệu được sử dụng cho Bài 2 là tập dữ liệu VLSP thuộc khuôn khổ Shared Task Machine Translation của cuộc thi VLSP 2025. Đây là một bộ dữ liệu chuyên biệt trong miền y tế, được xây dựng nhằm phục vụ các nghiên cứu và ứng dụng dịch máy từ tiếng Anh sang tiếng Việt trong bối cảnh tài liệu y khoa. Về quy mô, tập dữ liệu VLSP bao gồm 500 nghìn cặp câu song ngữ Anh-Việt trong tập huấn luyện (train set) và 3000 cặp câu cho tập kiểm tra (test set).

3.2 Phương pháp

So với các bộ dữ liệu dịch máy phổ thông như PhoMT, VLSP có một số đặc điểm khiến bài toán trở nên tách thức hơn. Thứ nhất, dữ liệu chứa mật độ cao các thuật ngữ y khoa chuyên ngành, bao gồm tên bệnh, triệu chứng, phương pháp điều trị, thủ thuật y tế và các khái niệm sinh học phức tạp. Nhiều thuật ngữ trong số này xuất hiện với tần suất rất thấp trong các tập dữ liệu dịch phổ thông, thậm chí không xuất hiện trong dữ liệu huấn luyện ban đầu của các mô hình pre-trained. Thứ hai, các câu trong VLSP thường có cấu trúc câu mang tính học thuật hoặc lâm sàng, với cách diễn đạt chính xác, súc tích và ít dư thừa, khác biệt đáng kể so với văn phong hội thoại hoặc báo chí.

Những đặc điểm trên khiến việc fine-tune trực tiếp mô hình trên VLSP gặp nhiều hạn chế, đặc biệt trong việc học và ghi nhớ các biểu thức ngôn ngữ đặc thù của miền y tế. Với quy mô dữ liệu tương đối nhỏ so với các tập huấn luyện phổ thông, VLSP khó có thể cung cấp đủ ngữ cảnh để mô hình học một cách đầy đủ các mối liên hệ giữa thuật ngữ, ngữ cảnh và cách diễn đạt chuẩn trong tiếng Việt. Do đó, việc bổ sung thêm dữ liệu huấn luyện trung gian trong cùng miền được xem là một hướng tiếp cận hợp lý nhằm cải thiện chất lượng dịch.

Nhóm đã áp dụng phương pháp thích nghi miền (Domain-Adaptive Pretraining) cho bài này (Gururangan et al., 2020). Cụ thể, bên cạnh bộ dữ liệu VLSP, nhóm sử dụng thêm tập dữ liệu ViPubMed (Phan et al., 2023) trong giai đoạn huấn luyện trung gian. ViPubMed là một tập dữ liệu y sinh quy mô lớn, bao gồm các cặp câu Anh–Việt được trích xuất và xây dựng từ các tài liệu khoa học và bài báo y sinh. Tập dữ liệu này có độ bao phủ thuật ngữ rộng và đa dạng ngữ cảnh, giúp mô hình tiếp cận với nhiều cách diễn đạt khác nhau trong miền y tế trước khi được tinh chỉnh cho nhiệm vụ cụ thể trên VLSP.

Nhóm đã triển khai hai giai đoạn huấn luyện mô hình như sau:

- **Domain Pretraining** Huấn luyện mô hình pre-trained trên tập dữ liệu y tế tổng quát ViPubMed. Nhóm thực hiện thay đổi kích thước dữ liệu huấn luyện: 300 nghìn, 500 nghìn và 700 nghìn cặp câu. Mục tiêu là để mô hình học được phân phối xác suất của các từ vựng y tế.

- **Task Fine-tuning** Sử dụng Checkpoint từ giai đoạn 1 để tiếp tục fine-tune trên tập dữ liệu VLSP nhằm tối ưu hóa cho nhiệm vụ cụ thể.

3.3 Huấn luyện mô hình

Khác với chiến lược huấn luyện từ đầu (training from scratch) được áp dụng trong Bài 1, ở giai đoạn này nhóm tiếp cận bài toán theo hướng tinh chỉnh

Bảng 2: So sánh các decoder trong mô hình Transformer

Decoder	BLEU	COMET
Scaled Dot-Product Attention		
Greedy	28.67	0.78
Beamsize=2	29.58	0.79
Beamsize=3	29.84	0.79
Beamsize=4	30.04	**0.80**
Beamsize=5	**30.10**	**0.80**
Luong Attention		
Greedy	28.31	0.78
Beamsize=2	29.01	0.79
Beamsize=3	29.28	0.79
Beamsize=4	29.60	0.79
Beamsize=5	29.57	0.79

In đậm thể hiện điểm cao nhất

(fine-tuning) mô hình ngôn ngữ lớn đã được huấn luyện trước. Cụ thể, mô hình được lựa chọn cho thí nghiệm là Helsinki-NLP/opus-mt-en-vi, một mô hình Transformer đã được huấn luyện trước trên Hugging Face cho tác vụ dịch từ tiếng Anh sang tiếng Việt. Việc sử dụng mô hình pre-trained giúp tận dụng tri thức ngôn ngữ phổ quát đã được học, từ đó rút ngắn thời gian hội tụ và nâng cao hiệu năng khi chuyển giao sang miền dữ liệu y tế đặc thù của VLSP. Quá trình huấn luyện được thực hiện theo cơ chế học có giám sát trên tập dữ liệu VLSP, trong đó dữ liệu đầu vào được mã hóa bằng tokenizer tương ứng của mô hình với độ dài chuỗi tối đa được thiết lập là 128 token nhằm đảm bảo cân bằng giữa khả năng bao quát ngữ cảnh và chi phí tính toán.

Để tối ưu hóa các tham số của mô hình trên miền dữ liệu mới, nhóm sử dụng hàm mất mát Cross-Entropy Loss tiêu chuẩn, tính toán sự sai biệt giữa phân phối xác suất của từ được dự đoán và từ đích thực tế. Quá trình tối ưu hóa được thực hiện thông qua thuật toán AdamW tích hợp trong thư viện Seq2SeqTrainer của Hugging Face, với tốc độ học (learning rate) được thiết lập ở mức nhỏ 2×10^{-5}. Đây là mức learning rate phù hợp cho giai đoạn fine-tuning, giúp mô hình điều chỉnh trọng số để thích nghi với các thuật ngữ y khoa mà không làm phá vỡ các tri thức ngôn ngữ cơ bản đã học được trước đó. Để tăng cường hiệu quả tính toán trên GPU, nhóm áp dụng kỹ thuật huấn luyện chính xác hỗn hợp (mixed-precision training - fp16) kết hợp với kích thước batch là 16 và tích lũy gradient (gradient accumulation) qua 2 bước.

Chu trình huấn luyện được thiết lập kéo dài trong 5 epoch. Đây là khoảng thời gian được xác định dựa trên quan sát thực nghiệm là đủ để mô hình hội tụ ổn định trên tập dữ liệu kích thước trung bình như VLSP mà không gây ra hiện tượng overfitting. Tại cuối mỗi epoch, mô hình được đánh giá trên tập validation (gồm 3000 cặp câu) thông qua hai chỉ số chính là giá trị hàm mất mát (validation loss) và điểm số BLEU. Việc theo dõi liên tục các chỉ số này cho phép kiểm soát chặt chẽ diễn biến huấn luyện, đảm bảo rằng mô hình không chỉ giảm thiểu sai số dự đoán mà còn thực sự cải thiện chất lượng dịch thuật trên các dữ liệu chưa từng gặp. Kết quả đánh giá trên tập kiểm thử sau epoch cuối cùng sẽ được sử dụng làm cơ sở để so sánh hiệu năng với các phương pháp tiếp cận khác như Domain-Adaptive Pretraining.

3.4 Kết quả thực nghiệm

Tương tự với bài 1, nhóm sử dụng hai độ đo phổ biến là BLEU và COMET để đánh giá chất lượng dịch máy trên bộ dữ liệu VLSP. Toàn bộ thí nghiệm được thực hiện trên môi trường Kaggle với GPU T4x2 (16GB VRAM) và 30GB RAM. Một số ví dụ cụ thể cho thí nghiệm này được trình bày trong phần Appendix B của bài báo này.

Baseline Để đánh giá hiệu quả của phương pháp thích nghi miền (domain-adaptive fine-tuning), nhóm thiết lập một mô hình baseline bằng cách fine-tune trực tiếp mô hình trên tập dữ liệu VLSP mà không sử dụng bất kỳ dữ liệu y tế bổ sung nào. Kết quả của mô hình này phản ánh năng lực của việc fine-tune đơn thuần trên một tập dữ liệu chuyên ngành có quy mô tương đối nhỏ và được sử

dụng làm mốc so sánh cho các phương pháp huấn luyện hai giai đoạn.

Kết quả Kết quả thực nghiệm được trình bày trong bảng 3 cho thấy phương pháp thích nghi miền mang lại sự cải thiện nhất quán về hiệu năng so với baseline. Cụ thể, mô hình fine-tune trực tiếp trên VLSP đạt điểm BLEU là 45.06 và COMET là 0.85. Khi mô hình được huấn luyện trung gian trên tập ViPubMed với 300 nghìn cặp câu trước khi fine-tune trên VLSP, điểm BLEU tăng lên 45.81 và COMET tăng lên 0.86. Xu hướng cải thiện tiếp tục được duy trì khi tăng quy mô dữ liệu ViPubMed lên 500 nghìn và 700 nghìn cặp câu, với điểm BLEU cao nhất đạt 46.57 và COMET đạt 0.86.

Kết quả này cho thấy việc huấn luyện trước trên dữ liệu y sinh tổng quát giúp mô hình học được phân phối xác suất của từ vựng và cấu trúc câu đặc thù trong miền y tế, từ đó cải thiện khả năng dịch khi chuyển sang tập VLSP. Tuy nhiên, mức cải thiện giữa các mốc 500 nghìn và 700 nghìn cặp câu là tương đối nhỏ, cho thấy mô hình đã bắt đầu đạt trạng thái bão hòa và lợi ích thu được từ việc bổ sung thêm dữ liệu huấn luyện trung gian không còn quá rõ rệt. Hiện tượng này tương đồng với quan sát ở Bài 1 khi hiệu suất mô hình bão hòa theo kích thước dữ liệu.

Phân tích quá trình hội tụ và độ ổn định của mô hình qua quá trình huấn luyện Dựa trên kết quả thực nghiệm và biểu đồ quá trình huấn luyện được trình bày trong Hình 5, tác động tích cực của phương pháp Domain-Adaptive Pretraining đối với khả năng hội tụ trở nên rất rõ ràng khi đặt cạnh mô hình Baseline. Sự khác biệt lớn nhất nằm ở điểm khởi đầu và tốc độ thích nghi; các mô hình đã qua huấn luyện sơ bộ trên ViPubMed—đặc biệt là phiên bản sử dụng 700k cặp câu—bước vào giai đoạn fine-tune với mức loss thấp hơn và các chỉ số dịch thuật khởi điểm cao hơn đáng kể. Điều này cho thấy mô hình đã hấp thụ được các phân phối xác suất và đặc trưng từ vựng của miền y tế ngay từ giai đoạn pre-training, cho phép chúng chỉ cần một lượng nhỏ gradient update để tinh chỉnh tham số thay vì phải học lại các cấu trúc ngôn ngữ cơ bản, qua đó rút ngắn đáng kể thời gian để đạt đến trạng thái hội tụ ổn định.

Bên cạnh lợi thế về tốc độ, các mô hình pre-trained còn thể hiện tính ổn định trong quá trình tối ưu hóa. Việc sử dụng dữ liệu trung gian giúp làm "mượt"không gian lỗi (loss landscape), thể hiện qua biên độ dao động nhỏ của hàm mất mát giữa các bước huấn luyện, từ đó giảm thiểu rủi ro mô hình bị kẹt tại các điểm tối ưu cục bộ kém chất lượng thường thấy khi fine-tune trực tiếp trên tập dữ liệu nhỏ như VLSP. Tuy nhiên, khi quan sát mức độ cải thiện biên giữa việc sử dụng 500 nghìn và 700 nghìn cặp câu pre-training, chúng tôi nhận thấy dấu hiệu của sự bão hòa tiệm cận. Điều này chỉ ra rằng, dù dữ liệu bổ sung giúp củng cố tri thức miền, nhưng kiến trúc mô hình và dung lượng thông tin giới hạn của tập đích VLSP mới là yếu tố quyết định trần hiệu năng cuối cùng, khẳng định vai trò bổ trợ nhưng không thể thay thế hoàn toàn dữ liệu fine-tune chất lượng cao của phương pháp pre-training.

4 Kết luận

Nghiên cứu này đã hoàn thành hai mục tiêu cốt lõi: (1) làm chủ và đánh giá kiến trúc Transformer cơ bản thông qua việc tái hiện và so sánh các biến thể kỹ thuật, và (2) tối ưu hóa mô hình cho bài toán dịch thuật chuyên ngành hẹp thông qua chiến lược thích nghi miền (Domain Adaptation).

Tổng hợp các kết quả thực nghiệm từ Bài 1 cho thấy, kiến trúc Transformer với cơ chế Scaled Dot-Product Attention không chỉ đạt hiệu năng dịch thuật cao hơn mà còn thể hiện tính ổn định tốt hơn trong quá trình hội tụ so với cơ chế Luong Attention truyền thống. Chúng tôi cũng quan sát thấy mối tương quan thuận rõ rệt giữa kích thước dữ liệu huấn luyện và chất lượng dịch; tuy nhiên, hiệu suất biên có xu hướng giảm dần và tiệm cận ngưỡng bão hòa khi dữ liệu đạt quy mô khoảng 700 nghìn cặp câu.

Trong Bài 2, việc giải quyết bài toán dịch máy y tế (VLSP) đã làm nổi bật vai trò then chốt của phương pháp Transfer Learning (Học chuyển tiếp). Chiến lược huấn luyện tận dụng tri thức từ tập dữ liệu lớn ViPubMed trước khi fine-tune trên tập dữ liệu đích đã đạt 46.57 điểm BLEU và 0.86 điểm COMET, chứng minh hiệu quả vượt trội so với baseline. Mô hình không chỉ cải thiện về các chỉ số định lượng (BLEU, COMET) mà còn thể hiện khả năng "hiểu"ngữ cảnh y khoa sâu sắc hơn, dịch chính xác các thuật ngữ chuyên môn phức tạp (như "nhồi máu cơ tim", "bệnh phổi tắc nghẽn mạn tính") thay vì dịch theo nghĩa đen hoặc sai lệch.

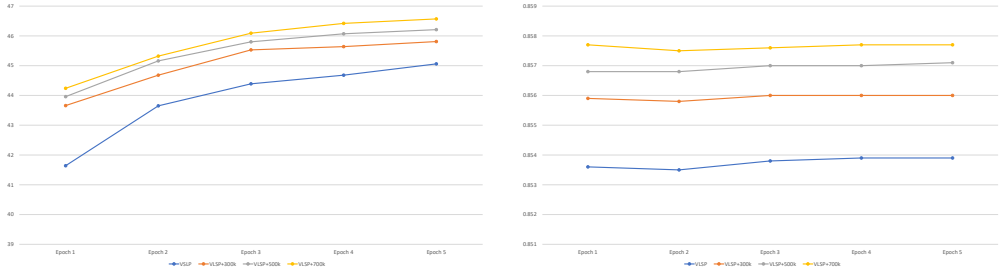
References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation:

Bảng 3: Kết quả fine-tune của trên bộ dữ liệu VLSP

Dataset	BLEU	COMET
VLSP	45.06	0.85
VLSP+300k	45.81	**0.86**
VLSP+500k	46.21	**0.86**
VLSP+700k	**46.57**	**0.86**

In đậm thể hiện điểm cao nhất



(a) Điểm BLEU trên từng epoch, trên các cách huấn luyện (b) Điểm COMET trên từng epoch, trên các cách huấn luyện

Hình 5: Phân tích điểm số đạt được của mô hình theo từng epoch

Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Long Doan, Linh The Nguyen, Nguyen Luong Tran, Thai Hoang, and Dat Quoc Nguyen. 2021. PhoMT: A high-quality and large-scale benchmark dataset for Vietnamese-English machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4495–4503, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Ibomoiye Domor Mienye, Theo G Swart, and George Obaido. 2024. Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, 15(9):517.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Long Phan, Tai Dang, Hieu Tran, Trieu H. Trinh, Vy Phan, Lam D. Chau, and Minh-Thang Luong. 2023. Enriching biomedical knowledge for low-resource language through large-scale translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3131–3142, Dubrovnik, Croatia. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

A Appendix A

Bảng A trình bày một số ví dụ minh họa sự khác biệt của hai chiến lược giải mã phổ biến trong mô

hình Transformer, bao gồm Greedy Search và Beam Search với kích thước chùm k=5. Thông qua các ví dụ này, có thể quan sát rõ rằng Beam Search nhìn chung cho kết quả dịch tốt hơn Greedy Search xét trên các tiêu chí về độ trôi chảy, tính tự nhiên cũng như mức độ mạch lạc của câu sinh ra.

Cụ thể, Greedy Search tại mỗi bước chỉ lựa chọn từ có xác suất cao nhất mà không xét đến ảnh hưởng dài hạn của lựa chọn đó lên toàn bộ chuỗi đầu ra. Cách tiếp cận mang tính cục bộ này khiến mô hình dễ rơi vào các vòng lặp xác suất cao, dẫn đến hiện tượng lặp từ hoặc lặp cụm từ không cần thiết, như có thể thấy rõ ở một số ví dụ trong bảng (ví dụ các câu chứa cụm "tôi luôn muốn" hoặc "cha mẹ" được lặp lại nhiều lần). Những lỗi này làm giảm đáng kể tính tự nhiên của câu dịch, mặc dù về mặt ngữ nghĩa tổng thể câu vẫn có thể hiểu được.

Ngược lại, Beam Search duy trì đồng thời nhiều giả thuyết dịch tiềm năng và đánh giá chúng dựa trên xác suất tích lũy của toàn bộ chuỗi. Nhờ đó, chiến lược này có khả năng tránh được các lựa chọn ngắn hạn kém tối ưu mà Greedy Search thường mắc phải. Khi một nhánh giải mã có xu hướng lặp hoặc suy giảm chất lượng, các nhánh khác trong chùm tìm kiếm vẫn có thể được giữ lại và tiếp tục phát triển, giúp mô hình lựa chọn được câu hoàn chỉnh hơn ở bước cuối. Điều này thể hiện rõ qua các kết quả dịch bằng Beam Search trong bảng, khi các câu đầu ra thường ngắn gọn hơn, ít lặp từ, có cấu trúc ngữ pháp rõ ràng và sát với cách diễn đạt tự nhiên của tiếng Việt.

Ngoài ra, Beam Search còn cho thấy ưu thế trong việc xác định điểm kết thúc câu hợp lý. Trong nhiều trường hợp, Greedy Search tiếp tục sinh thêm từ dù nội dung câu đã hoàn chỉnh, trong khi Beam Search có xu hướng dừng đúng thời điểm, góp phần cải thiện tính súc tích và mạch lạc của bản dịch. Điều này đặc biệt quan trọng đối với các tác vụ dịch máy, nơi mà việc sinh ra câu quá dài hoặc dư thừa thông tin có thể làm giảm chất lượng tổng thể của hệ thống.

Từ các phân tích trên, có thể kết luận rằng Beam Search là chiến lược giải mã phù hợp hơn Greedy Search trong bối cảnh dịch máy bằng Transformer, đặc biệt khi yêu cầu cao về tính tự nhiên, độ trôi chảy và khả năng tránh lặp trong câu sinh ra.

B Appendix B

Bảng B trình bày một số ví dụ minh họa sự khác biệt giữa hai cách huấn luyện mô hình dịch máy trong miền y tế, bao gồm (i) dùng thẳng mô hình để dịch và (ii) fine-tune mô hình trên tập ViPubMed trước, sau đó tiếp tục fine-tune trên VLSP. Thông qua các ví dụ cụ thể, có thể quan sát rõ ảnh hưởng của quá trình huấn luyện trung gian trên dữ liệu y sinh chuyên ngành đối với chất lượng bản dịch từ tiếng Anh sang tiếng Việt.

Xét tổng thể, phương pháp dịch trực tiếp cho kết quả dịch tương đối chính xác về mặt ý nghĩa, tuy nhiên trong nhiều trường hợp vẫn còn tồn tại các hạn chế liên quan đến việc lựa chọn thuật ngữ chuyên ngành, cách diễn đạt chưa thực sự tự nhiên, cũng như sự thiếu nhất quán trong việc sử dụng các cụm từ y khoa quen thuộc trong tiếng Việt. Điều này xuất phát từ việc kích thước và mức độ bao phủ thuật ngữ của tập VLSP còn hạn chế, khiến mô hình khó học được đầy đủ các biểu thức đặc thù của miền y tế.

Ngược lại, khi mô hình được fine-tune trước trên tập ViPubMed - một tập dữ liệu lớn và giàu ngữ cảnh y sinh - trước khi tiếp tục fine-tune trên VLSP, chất lượng bản dịch cho thấy sự cải thiện rõ rệt. Cụ thể, các bản dịch trong cột thứ tư thường sử dụng thuật ngữ y khoa chính xác và chuẩn hóa hơn, chẳng hạn như "nhồi máu cơ tim cấp", "đái tháo đường típ 2", "bệnh phổi tắc nghẽn mạn tính" hay "gây mê toàn thân", thay vì các cách diễn đạt mang tính mô tả hoặc chưa hoàn toàn chuẩn thuật ngữ như trong cột baseline. Việc sử dụng đúng thuật ngữ không chỉ nâng cao độ chính xác chuyên môn mà còn làm cho bản dịch phù hợp hơn với văn phong của các tài liệu y khoa chính thống.

Ngoài độ chính xác về thuật ngữ, phương pháp huấn luyện hai giai đoạn còn cho thấy ưu thế về tính tự nhiên và tính mạch lạc của câu dịch. Các bản dịch sau khi fine-tune qua ViPubMed thường có cấu trúc câu rõ ràng, cách kết hợp từ ngữ tự nhiên hơn và ít mang dấu vết của dịch từng từ. Ví dụ, các cụm như "được gọi là một kẻ giết người thầm lặng", "sử dụng dịch truyền tĩnh mạch để điều trị mất nước" hay "bác sĩ kê đơn kháng sinh phổ rộng" thể hiện phong cách diễn đạt gần với ngôn ngữ chuyên ngành được sử dụng trong thực hành lâm sàng và các tài liệu y học tiếng Việt.

Kết quả này cho thấy quá trình fine-tune trung gian trên ViPubMed đã giúp mô hình học được không chỉ từ vựng chuyên ngành mà còn cả các mẫu câu, cấu trúc diễn đạt và ngữ cảnh thường gặp trong miền y sinh. Nhờ đó, khi tiếp tục fine-tune trên VLSP, mô hình có thể nhanh chóng thích nghi với đặc thù của tập dữ liệu mục tiêu trong khi vẫn duy trì được kiến thức miền đã học trước đó. Cách tiếp cận này giúp cải thiện đáng kể hiệu quả của

Bảng 4: Các ví dụ sử dụng Greedy Search và Beam Search ($k = 5$) trong mô hình Transformer

No.	Source	Greedy Search	Beam Search (k=5)
0	I have always wanted to do that.	Tôi luôn muốn làm điều đó, tôi luôn luôn muốn làm điều đó.	Tôi luôn muốn làm điều đó.
1	Today, I, as the mother of a six - year - old, walk into Barnes and Noble, and see this.	Hôm nay, tôi như mẹ của một đứa trẻ 6 tuổi đi vào Byrnes và Noble, và xem đây, và thấy điều này.	Hôm nay, tôi như mẹ của một đứa trẻ 6 tuổi đi vào Byrnes và Noble, và thấy điều này.
2	But taken together, I am sorry, I do not see help when I look at that shelf.	Nhưng tôi xin lỗi, tôi không thấy gì khi nhìn vào cái kệ đó.	Nhưng tôi xin lỗi, tôi không thấy sự giúp đỡ khi nhìn vào chiếc kệ đó.
3	Their marital satisfaction is lower.	Sự hài lòng hôn nhân của họ thấp hơn, thấp hơn, thấp hơn, hài lòng hôn nhân.	Thoả mãn hôn nhân của họ thấp hơn.
4	We provided food, clothing, shelter, and moral instruction to our kids, and they in return provided income.	Chúng tôi cung cấp thực ăn, cộng đồng và hướng dẫn đạo đức cho trẻ em, và chúng trả lại thu nhập, và cung cấp thu nhập cho chúng tôi, và chúng sẽ cung cấp cho chúng tôi, và chúng trả lại thu nhập.	Chúng tôi cung cấp thực phẩm, cộng đồng, và hướng dẫn đạo đức cho trẻ em, và chúng trả lại thu nhập, và trả lại thu nhập.
5	This was true even when I was young.	Điều này đúng ngay cả khi tôi còn nhỏ, tôi vẫn còn nhỏ	Điều này đúng ngay cả khi tôi còn trẻ.
6	I was even invited to speak at TED.	Tôi thậm chí được mời lên diễn tại TED. TED.	Thậm chí chỉ tôi còn được mời nói tại TED. TED.
7	Guess what it was?	Đoán xem nó là gì ?	Đoán xem nó là gì ?
8	A child's happiness is a very unfair burden to place on a parent.	Trẻ em không hạnh phúc là một gánh nặng không công bằng để đặt lên cha mẹ, cha mẹ, cha mẹ, cha mẹ, cha mẹ, cha mẹ, con cái là một gánh nặng không công bằng.	Một đứa trẻ không hạnh phúc là một gánh nặng không công bằng đối với cha mẹ, cha mẹ, cha mẹ, cha mẹ là một gánh nặng không công bằng.
9	The weather in Hanoi is very beautiful today.	Thời tiết ở Hansi rất đẹp ngày nay, ngày nay, thời tiết ở Hansi rất đẹp.	Thời tiết ở Hanso ngày hôm nay rất đẹp.

mô hình trong các bài toán dịch máy chuyên ngành.

Từ các ví dụ được trình bày trong bảng, có thể kết luận rằng việc fine-tune mô hình trên một tập dữ liệu y sinh lớn như ViPubMed trước khi huấn luyện trên VLSP mang lại lợi ích rõ rệt so với việc fine-tune trực tiếp trên VLSP. Phương pháp này không chỉ cải thiện độ chính xác thuật ngữ mà còn nâng cao tính tự nhiên và tính nhất quán của bản dịch, qua đó góp phần nâng cao chất lượng tổng thể của hệ thống dịch máy y tế.

Bảng 5: Các ví dụ minh họa về trước và sau khi fine-tune mô hình trên bộ VLSP

No.	Source	Baseline	Fine-tuned trên ViPubMed trước
0	The patient suffered from acute myocardial infarction.	Bệnh nhân bị viêm tim mạch.	Bệnh nhân bị nhồi máu cơ tim cấp.
1	She has a family history of type 2 diabetes mellitus.	Cô ấy có tiền sử gia đình loại 2 tiểu thuyết gia đình.	Bệnh nhân có tiền sử gia đình mắc đái tháo đường típ 2.
2	Chronic obstructive pulmonary disease is strictly linked to smoking.	Bệnh dai dẳng có liên quan đến việc hút thuốc.	Bệnh phổi tắc nghẽn mãn tính có liên quan chặt chẽ đến hút thuốc lá.
3	The biopsy results confirmed a malignant tumor in the liver.	Kết quả xét nghiệm cho thấy có một khối u ở gan.	Kết quả sih thiết xác nhận u ác tính ở gan.
4	Hypertension is often referred to as a silent killer.	Áp lực thường được gọi là kẻ giết người im lặng.	Tăng huyết áp thường được gọi là một kẻ giết người thầm lặng.
5	Common symtoms include high fever, dry cough, and fatigue.	Những triệu chứng thông thường như sốt cao, ho khô và mệt mỏi	Các triệu chứng gặp bao gồm sốt cao, ho khan và mệt mỏi.
6	The doctor prescribed broad-spectrum antibiotics for the respiratory infection.	Khám nghiệm kháng sinh phổ rộng khánh sinh kháng sinh cho nhiễm trùng hô hấp	Bác sĩ kê đơn kháng sinh phổ rộng để điều trị nhiễm trùng hô hấp.
7	He was hospitalized due to acute renal failure?	Ông ấy bị suy giảm nghiêm trọng.	Bệnh nhân nhập viện vì suy thận cấp.
8	Intravenous fluids were administered to treat severe dehydration	Các chất lỏng không thấm đã được chữa trị nghiêm trọng.	Sử dụng dịch truyền tĩnh mạch để điều trị mất nước.
9	The surgery was performed under general anesthesia.	Cuộc phẫu thuật được thực hiện dưới chế độ tổng quát.	Phẫu thuật được thực hiện dưới gây mê toàn thân.