Báo cáo Dự án mô hình dịch máy Anh–Việt: Kiến trúc Transformer và mô hình pre-trained cho Medical Domain tại VLSP 2025

Nguyễn Đăng Khoa **Nguyễn Văn Minh** **Phạm Quang Hưng** **Nguyễn Viết Cường**
Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội
23020093@vnu.edu.vn, 23020117@vnu.edu.vn, 23020084@vnu.edu.vn, 21020173@vnu.edu.vn

Tóm tắt nội dung

Báo cáo này trình bày quá trình nghiên cứu và phát triển hệ thống dịch máy Anh-Việt thông qua hai giai đoạn. Giai đoạn 1 tập trung xây dựng mô hình Transformer từ đầu (from-scratch) dựa trên kiến trúc gốc để nắm bắt cơ chế Attention. Giai đoạn 2 giải quyết bài toán thực tế tại cuộc thi VLSP 2025: Dịch máy miền Y tế với tài nguyên hạn chế. Nhóm đề xuất phương pháp tinh chỉnh (Fine-tuning) mô hình Qwen-2.5 (0.5B tham số) sử dụng kỹ thuật LoRA và mô hình kiến trúc T5 (envit5-translation). Kết quả cho thấy mô hình T5 giúp xử lý hiệu quả các thuật ngữ chuyên ngành trong y văn.

1 Giới thiệu

Dịch máy là một nhánh quan trọng của Xử lý Ngôn ngữ Tự nhiên, mang lại lợi ích trực tiếp cho các vùng và quốc gia kết nối với nhau trên thế giới, đặc biệt là đối với các nền kinh tế đang phát triển nhanh như Việt Nam. Đặc biệt, dịch máy trong miền y tế (Medical Domain) đặt ra thách thức lớn do yêu cầu cao về độ chính xác của thuật ngữ, sự khan hiếm dữ liệu song ngữ chất lượng cao và sự phức tạp của các từ viết tắt, tên thuốc (VLSP Organizers, 2025).

Tại VLSP 2025, bài toán được đặt trong bối cảnh tài nguyên giới hạn (Limited-Pretraining models), yêu cầu các mô hình tham gia không vượt quá 3 tỷ tham số nhưng vẫn phải đảm bảo chất lượng dịch thuật trung thực (VLSP Organizers, 2025). Báo cáo này mô tả chi tiết phương pháp tiếp cận của nhóm đối với cả mô hình Transformer cơ bản và mô hình tiền huấn luyện hiện đại.

2 Một số nghiên cứu liên quan

Trong những năm gần đây, các mô hình ngôn ngữ tiền huấn luyện (Pretrained Language Models - LMs) đã đóng một vai trò quan trọng và mới mẻ trong việc phát triển nhiều hệ thống Xử lý Ngôn ngữ Tự nhiên (NLP). Việc tận dụng các mô hình lớn như BERT (Devlin et al., 2018), ALBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019), GPT-3 (Brown et al., 2020), BART (Lewis et al., 2020), và T5 (Raffel et al., 2020) đã trở thành một xu hướng hiệu quả trong xử lý ngôn ngữ tự nhiên. Tất cả các mô hình lớn này đều tuân theo kiến trúc Transformer do Vaswani et al. (2017) đề xuất với cơ chế attention. Kiến trúc này đã được chứng minh là rất phù hợp để tinh chỉnh (finetune) cho các tác vụ xuôi dòng (downstream tasks), tận dụng khả năng học chuyển giao (transfer learning) từ các checkpoint tiền huấn luyện lớn của chúng.

Sự thành công của các mô hình tiền huấn luyện trên miền dữ liệu tổng quát (BERT, RoBERTa, BART, T5, v.v.) đã mở ra hướng đi trong việc tạo ra các mô hình ngôn ngữ cho miền cụ thể (domain-specific), chẳng hạn như CodeBERT (Feng et al., 2020) và CoTexT (Phan et al., 2021) cho ngôn ngữ lập trình, TaBERT (Yin et al., 2020) cho dữ liệu bảng, BioBERT (Lee et al., 2020) và PubmedBERT (Gu et al., 2021) cho ngôn ngữ y sinh.

Y văn đang ngày càng trở nên phổ biến và dễ tiếp cận hơn đối với cộng đồng khoa học thông qua các cơ sở dữ liệu lớn như Pubmed[1], PMC[2], và MIMIC-IV (Johnson et al., 2021). Điều này cũng dẫn đến việc nhiều nghiên cứu, kho ngữ liệu hoặc dự án được công bố nhằm thúc đẩy lĩnh vực Xử lý Ngôn ngữ Tự nhiên Y sinh.

Đối với các mô hình dịch máy cho Medical Domain cho Tiếng Việt, ViPubmed (Phan et al., 2022) đã có đột phá trong nghiên cứu dữ liệu và các chuẩn mực đánh giá trong ngành y sinh học rất có giá trị nhưng lại rất hạn chế ở các ngôn ngữ ít tài nguyên ngoài tiếng Anh, chẳng hạn như tiếng Việt. Trong bài báo này, họ sử dụng một mô hình dịch thuật tiên tiến tiếng Anh - tiếng Việt để dịch và tạo ra cả dữ liệu được huấn luyện trước và dữ liệu giám sát trong lĩnh vực y sinh học. Hơn nữa, ViMedNLI

[1] https://pubmed.ncbi.nlm.nih.gov
[2] https://www.ncbi.nlm.nih.gov/pmc

- một nhiệm vụ NLP mới bằng tiếng Việt được dịch từ MedNLI sử dụng mô hình dịch En-vi vừa công bố và được các chuyên gia con người tinh chỉnh cẩn thận.

3 Phân tích dữ liệu ngôn ngữ

3.1 Dataset: IWSLT English–Vietnamese

Trong bài toán dịch máy Anh–Việt, chất lượng dữ liệu song ngữ đóng vai trò quyết định đối với khả năng học được ánh xạ ngữ nghĩa và cấu trúc cú pháp giữa hai ngôn ngữ. Nhóm sử dụng tập dữ liệu **IWSLT English–Vietnamese** (thường được biết đến như một biến thể của bộ dữ liệu IWSLT/TED Talks cho tác vụ dịch máy), bao gồm các cặp câu (sentence pairs) tiếng Anh và tiếng Việt. IWSLT là một trong các bộ dữ liệu phổ biến trong nghiên cứu dịch máy, đặc trưng bởi miền diễn thuyết nói (spoken-style) và độ dài câu trung bình vừa phải, phù hợp để huấn luyện và đánh giá các mô hình Transformer (Cettolo et al., 2012; Ott et al., 2019; Vaswani et al., 2017).

Phân tích dữ liệu Phần này nhằm trả lời bốn câu hỏi chính: (1) phân phối độ dài câu ra sao và ảnh hưởng thế nào đến batching/huấn luyện Transformer; (2) đặc trưng từ vựng và các hình thái "vocab đặc biệt" (số, viết tắt, tên riêng, token hiếm) của tiếng Anh/tiếng Việt; (3) các lỗi dữ liệu thường gặp và tiêu chí lọc/chuẩn hoá; và (4) lựa chọn chiến lược tokenization phù hợp cho EN và VI để tối ưu BLEU và ổn định quá trình học.

3.2 Phân tích dữ liệu trong dịch máy

Dịch máy bằng Transformer giả định rằng dữ liệu đầu vào là các cặp câu đồng bộ, ít nhiễu, và phân phối thống kê tương đối ổn định. Nếu dữ liệu chứa sai lệch (misalignment), ký tự lạ, hoặc nhiễu hình thức (formatting noise), mô hình sẽ học các tương quan sai, làm giảm chất lượng tổng quát hoá và tăng rủi ro suy biến khi suy luận (degenerate generation). Các nguyên nhân khiến tiền xử lý là bắt buộc gồm:

(i) Parallel misalignment Một số dòng có thể bị lệch do thiếu dòng, xuống dòng bất thường, hoặc ghép đoạn. Khi đó, mô hình sẽ học ánh xạ sai giữa câu nguồn và câu đích, làm nhiễu tín hiệu huấn luyện. Do Transformer tối ưu hoá likelihood theo cặp câu, misalignment trực tiếp làm giảm chất lượng tối ưu và có thể gây "hallucination" trong dịch.

(ii) Nhiễu ký tự và chuẩn hoá Unicode Các dạng dấu ngoặc, dấu nháy cong, ký tự không ngắt dòng (NBSP), hoặc dấu gạch khác chuẩn có thể làm tăng kích thước từ vựng biểu kiến và gây phân mảnh token. Trong môi trường dữ liệu kỹ thuật, hiện tượng đa ngôn ngữ, ký hiệu toán học và thuật ngữ chuyên ngành trộn lẫn là phổ biến; các mô tả về tài liệu kỹ thuật cho thấy dữ liệu có thể chứa ký hiệu toán học và nội dung song ngữ Việt–Anh (vbk, 2025). Từ đó, ngay cả với dữ liệu MT "tưởng như sạch", việc chuẩn hoá Unicode và lọc ký tự bất thường vẫn là bước quan trọng để giảm nhiễu.

(iii) Ngoại lệ về độ dài và hiệu ứng lên huấn luyện Transformer có chi phí $O(n^2)$ theo độ dài chuỗi trong attention (Vaswani et al., 2017). Do đó, một số ít câu quá dài sẽ làm tăng thời gian huấn luyện, giảm hiệu quả GPU, và có thể khiến batch phải giảm kích thước dẫn đến nhiễu gradient. Vì vậy, phân tích và lọc theo độ dài (length filtering) là thao tác thực dụng để tối ưu hoá tài nguyên và ổn định tối ưu.

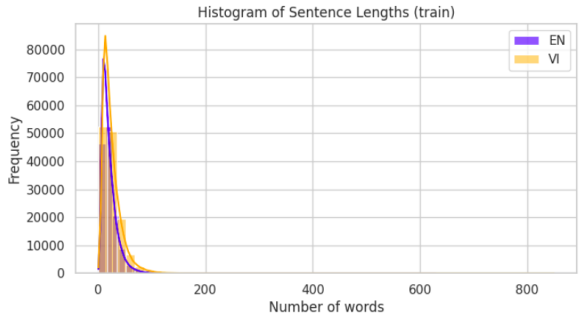
(iv) Tách từ và mô hình hoá tiếng Việt Tiếng Việt là ngôn ngữ cách âm tiết bằng khoảng trắng nhưng đơn vị từ có thể gồm nhiều âm tiết (ví dụ: "*giáo dục*", "*công nghệ thông tin*"). Nếu coi khoảng trắng như ranh giới từ tuyệt đối, mô hình có thể bị phân mảnh quá mức ở phía đích, dẫn đến chuỗi token dài hơn và học dịch kém ổn định. Do đó, cần cân nhắc tokenization dựa trên subword (BPE/SentencePiece) hoặc kết hợp tiền xử lý tách từ tiếng Việt.

3.3 Xử lý dữ liệu Tiếng Việt

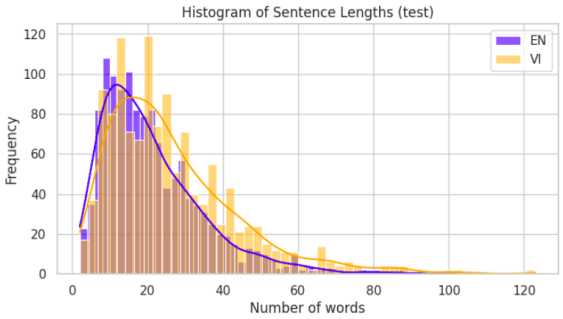
Dấu thanh và biến thể chính tả Tiếng Việt sử dụng dấu thanh và ký tự có dấu, khiến khoảng cách Levenshtein giữa các biến thể có/không dấu tương đối nhỏ nhưng khác nghĩa lớn. Dữ liệu thực tế đôi khi có lỗi mã hoá (ví dụ: mất dấu do lỗi encoding) hoặc sử dụng nhiều biến thể dấu nháy/dấu gạch. Nhóm lựa chọn chuẩn hoá Unicode (NFC) nhằm đảm bảo mỗi ký tự có dấu được biểu diễn nhất quán.

Chuẩn hoá khoảng trắng và dấu câu Các lỗi phổ biến: nhiều khoảng trắng liên tiếp, khoảng trắng trước dấu câu, hoặc dính dấu câu vào từ theo kiểu không nhất quán. Các lỗi này làm tăng khối lượng vocab cho tập huấn luyện.

Tên riêng, số và ký hiệu Trong dữ liệu TED/IWSLT, tên người, địa danh, chữ viết tắt (AI, GDP, ...), và số (năm, tỉ lệ phần trăm) xuất hiện



Hình 1: Phân phối độ dài câu tiếng Anh và Tiếng Việt (theo số từ) (tập train).



Hình 2: Phân phối độ dài câu tiếng Anh và Tiếng Việt (theo số từ) (tập test).

thường xuyên. Nếu không xử lý, mô hình có thể học dịch sai số hoặc chuẩn hoá không nhất quán. Thực nghiệm thường cho thấy việc giữ nguyên số và một số ký hiệu có lợi cho BLEU, nhưng cần chuẩn hoá định dạng (ví dụ: “1,000” vs “1000”).

3.4 Exploratory Data Analysis (EDA)

3.4.1 Thống kê độ dài câu

Thống kê độ dài câu ở cả hai phía nguồn (EN) và đích (VI) theo số token thô (whitespace tokens) trước khi áp dụng subword tokenization.

Dữ liệu thường có phần lớn câu nằm trong vùng độ dài ngắn (10-100 từ); tuy nhiên, một tỷ lệ nhỏ câu dài bất thường (do gộp đoạn hoặc chứa liệt kê) có thể chi phối chi phí huấn luyện. Đặc biệt, phía tiếng Việt có thể dài hơn tiếng Anh về số token thô do đặc trưng tách âm tiết bằng khoảng trắng. (Hình 1)

Quy tắc lọc theo độ dài. Dựa trên thống kê, nhóm áp dụng các tiêu chí lọc sau:

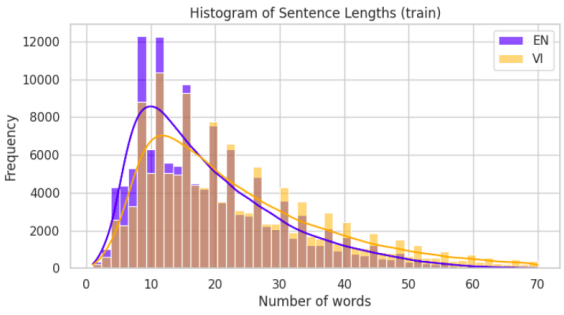
- Loại bỏ cặp câu rỗng hoặc chỉ gồm dấu câu.
- Loại bỏ cặp câu nếu $\max(\text{len}_{en}, \text{len}_{vi}) > L_{\max}$, với $L_{\max}$ thiết lập theo phân vị (ví dụ 95 hoặc 95.5 percentile).

```
Original: 133317 pairs
After filtering by length (≤70): 129706 pairs
After removing empty: 129555 pairs
Original: 1553 pairs
After filtering by length (≤70): 1539 pairs
After removing empty: 1539 pairs
Original: 1268 pairs
After filtering by length (≤70): 1233 pairs
After removing empty: 1233 pairs
=======
Train len: 129555 129555
Valid len: 1539 1539
Test len: 1233 1233
```

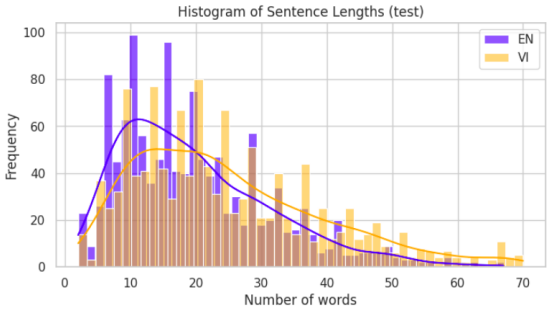
Hình 3: Dữ liệu câu tiếng Anh và Tiếng Việt sau khi lọc.

- Loại bỏ cặp câu có tỉ lệ độ dài bất thường: $\frac{\text{len}_{vi}}{\text{len}_{en}} \notin [r_{\min}, r_{\max}]$ để giảm nguy cơ misalignment.

Các tiêu chí này giúp giảm nhiễu, tăng hiệu quả batching và làm ổn định huấn luyện Transformer. Trong quá trình phát triển mô hình transformer để phù hợp với hạ tầng cho phép, dữ liệu data được giới hạn độ dài bằng cách lọc và dữ lại 70 percentile. (Hình 4)



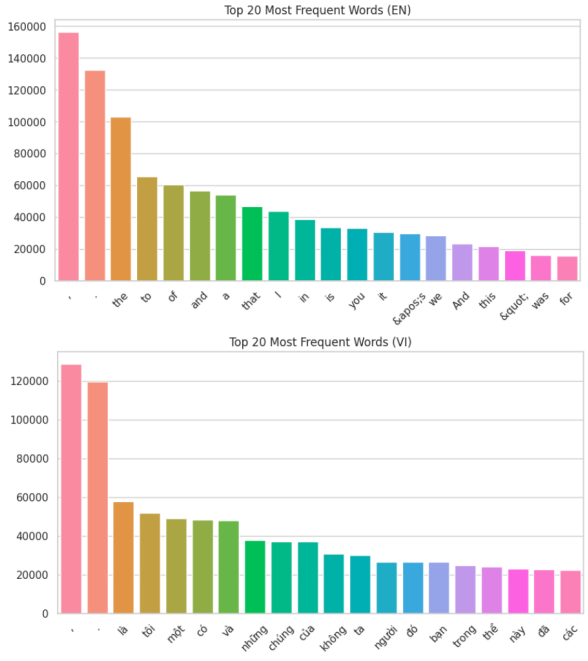
Hình 4: Phân phối độ dài câu tiếng Anh và Tiếng Việt (theo số từ) (tập train).



Hình 5: Phân phối độ dài câu tiếng Anh và Tiếng Việt (theo số từ) (tập test).

3.4.2 Phân phối từ vựng

- Số lượng từ loại (type) và số lần xuất hiện (token frequency).
- Tỷ lệ từ hiếm (hapax legomena, xuất hiện 1 lần).
- Tỷ lệ ký tự ngoài ASCII ở tiếng Anh và phân phối dấu tiếng Việt.



Hình 6: Phân phối top 20 từ vựng EN-VI tập training

Vocab đặc biệt và nhiễu

- Token chứa ký tự điều khiển hoặc ký tự "không in được" (zero-width, \u200b, ...).
- Token chứa html tags.
- Chuỗi lặp dấu câu bất thường ("....", "????", "—").

Từ các thống kê trên, nhóm xây dựng danh sách quy tắc làm sạch (cleaning rules) theo hướng bảo toàn nội dung ngữ nghĩa, nhưng loại bỏ nhiễu định dạng.

3.5 Phương pháp tokenization

Tokenization quyết định không gian biểu diễn đầu vào/đầu ra của Transformer. Nhóm xem xét hai hướng chính: tokenization theo từ (word-level) và tokenization theo đơn vị con (subword-level). Trong thực tế, Transformer hiện đại gần như luôn dùng subword (BPE/SentencePiece) để cân bằng giữa độ phủ từ vựng và độ dài chuỗi (Sennrich et al., 2016; Kudo and Richardson, 2018).

3.5.1 English Tokenization

Với tiếng Anh, các bước chuẩn thường gồm:

- Tách dấu câu cơ bản và chuẩn hoá dấu nháy.
- Duy trì phân biệt chữ hoa/thường hoặc lowercasing tuỳ cấu hình (nhóm ưu tiên giữ nguyên để bảo toàn tên riêng).
- Học mô hình subword (BPE/SentencePiece) trên tập huấn luyện để thu được vocab cỡ V (ví dụ 8k–32k).

3.5.2 Vietnamese Tokenization

Với tiếng Việt, nhóm thảo luận ba lựa chọn:

(i) Giữ nguyên theo âm tiết (whitespace-separated syllables) Ưu điểm: đơn giản, không cần bộ tách từ; nhược điểm: chuỗi dài, làm chi phí attention tăng và mô hình khó học cụm từ đa âm tiết.

(ii) Tách từ tiếng Việt trước, sau đó subword Có thể dùng bộ tách từ tiếng Việt để ghép các âm tiết thuộc cùng một từ bằng dấu gạch dưới (ví dụ công_nghệ). Ưu điểm: giảm độ dài chuỗi và phản ánh đúng hơn đơn vị từ vựng; nhược điểm: sai số tách từ có thể lan truyền sang mô hình dịch. (thư viện sử dụng: PyVi)

(iii) SentencePiece Unigram/BPE trực tiếp trên văn bản thô Đây là lựa chọn thường mạnh trong thực nghiệm vì mô hình subword tự học các mảnh phù hợp mà không phụ thuộc mạnh vào tách từ thủ công (Kudo and Richardson, 2018). Nhóm ưu tiên hướng này để giảm phụ thuộc công cụ và tăng tính tái lập.

Thiết lập vocab và chia sẻ vocab Nhóm xem xét cả (a) vocab riêng cho EN và VI, và (b) shared vocab. Với cặp ngôn ngữ khác hệ chữ, vocab riêng thường ổn định hơn; tuy nhiên shared vocab có thể hữu ích cho ký hiệu chung (số, dấu câu, viết tắt). Trong báo cáo này, nhóm trình bày cấu hình mặc định sử dụng hai vocab riêng để tối ưu hoá độ bao phủ.

Bảng 1: Tóm tắt tiêu chí làm sạch và lọc dữ liệu song ngữ.

Nhóm tiêu chí	Mô tả
Rỗng/thiếu nội dung	Loại bỏ câu rỗng, câu chỉ có dấu câu, hoặc câu có tỷ lệ ký tự chữ quá thấp.
Độ dài cực trị	Loại bỏ nếu vượt $L_{\max}$ theo phân vị; đồng thời kiểm tra tỉ lệ độ dài EN/VI trong khoảng $[r_{\min}, r_{\max}]$.
Ký tự bất thường	Loại bỏ/chuẩn hoá zero-width, control chars, NBSP; chuẩn hoá Unicode (NFC).
Trùng lặp	Loại bỏ duplicate pairs để giảm overfitting.

Tác động đến huấn luyện Transformer Sau tiền xử lý, dữ liệu có phân phối độ dài gọn hơn, giảm long-tail, vocab bớt nhiễu ký tự và giảm tỷ lệ token hiếm giả tạo. Điều này giúp mô hình: (i) hội tụ ổn định hơn, (ii) tăng hiệu quả GPU nhờ batching theo bucket độ dài, và (iii) cải thiện chất lượng dịch đo bằng BLEU do giảm lỗi sao chép ký tự và lỗi dấu câu.

4 Task 1: Mô hình Transformer

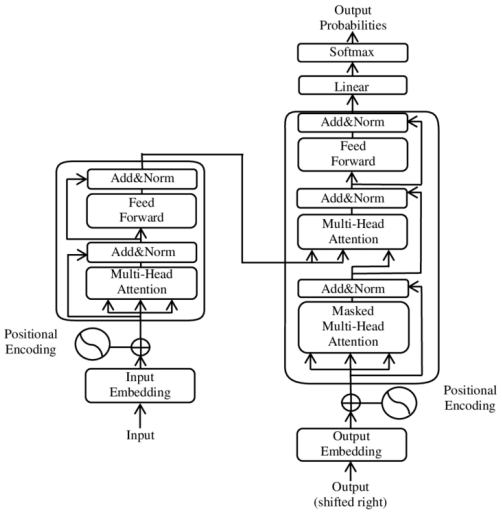
4.1 Thử nghiệm 1: Mô hình Base Transformer

4.1.1 Kiến trúc mô hình

Nhóm quyết định phát triển mô hình Transformer theo kiến trúc Transformer trong bài báo nổi tiếng (Vaswani et al., 2017). Kiến trúc tuân theo chuẩn Encoder - Decoder thông thường, mỗi bộ phận Encoder và Decoder được cấu thành từ 8 lớp EncoderLayer và DecoderLayer nhỏ. Mô hình Transformer lớn được xây dựng bằng cách xếp chồng (stack) 6 lớp Transformer

Theo kiến trúc được thể hiện trong Hình 8, nhóm đã xác định các thành phần nền tảng để xây dựng lên mô hình Transformer hoàn chỉnh, bao gồm:

Positional Encoder Thành phần này đảm nhiệm việc biểu diễn thông tin về vị trí tương đối và tuyệt đối của các token trong chuỗi bằng cách kết hợp mã hóa vị trí với embedding đầu vào. Trong kiến trúc Transformer tiêu chuẩn, các hàm sine và cosine được sử dụng để mã hóa vị trí theo dạng tín hiệu hình sin, cho phép mô hình nắm bắt quan hệ thứ tự trong chuỗi. Cách tiếp cận này có cấu trúc đơn giản và có tiềm năng cải thiện thêm.



Hình 7: Kiến trúc mô hình Transformer tiêu chuẩn

$$\mathrm{PE}_{pos,2i} = \sin(pos/10000^{2i/\mathrm{d}_{model}})$$

$$\mathrm{PE}_{(pos,2i+1)} = \cos(pos/10000^{2i/\mathrm{d}_{model}})$$

Embedder Thành phần chịu trách nhiệm ánh xạ các tokens rời rạc vào miền không gian vector liên tục nhằm trích xuất các thông tin về ngữ nghĩa và cú pháp của từ.

Attention Layers Dựa trên cơ chế attention được đề xuất trong bài báo (Vaswani et al., 2017), kiến trúc Encoder–Decoder trong bài toán dịch máy sử dụng ba dạng attention nhằm phục vụ các mục đích khác nhau dựa trên công thức Attention cơ bản

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

Multi-Head Self-Attention (non-masked) được sử dụng trong Encoder, cho phép mỗi token trong câu nguồn chú ý đến toàn bộ các token còn lại để học các phụ thuộc ngữ cảnh toàn cục.

Masked Multi-Head Self-Attention được sử dụng trong Decoder nhằm ngăn mô hình chú ý đến các tokens trong tương lai khi sinh.

Cross-Attention (Encoder–Decoder Attention) cũng được sử dụng trong Decoder, cho phép mỗi bước sinh token dựa trên các biểu diễn ẩn được trích xuất từ Encoder.

Feed-Forward Network (FFN) Đối với mỗi token, đầu ra của cơ chế attention là tuyến tính. Vì vậy, lớp FFN được sử dụng để biến đổi các

features một cách phi tuyến sử dụng hàm kích hoạt ReLU

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

Layer Normalization Theo tiêu chuẩn huấn luyện mô hình Transformer, các lớp LayerNorm được đặt sau mỗi đầu ra của các lớp biến đổi (Attention, FFN) để chuẩn hóa các hàm kích hoạt. Tuy nhiên, cơ chế này có thể được chỉnh sửa để hướng tới huấn luyện ổn định hơn, được trình bày trong mục 4.2.

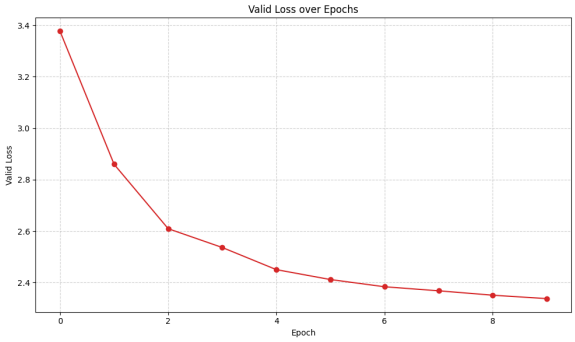
Encoder Mỗi Encoder của một Transformer block được cấu thành từ 6 lớp EncoderLayer nhỏ. Mỗi lớp nhỏ được xây dựng theo kiến trúc $Multi - HeadAttention \rightarrow LayerNorm \rightarrow FeedForwardNetwork \rightarrow LayerNorm$ Tại mỗi thành phần LayerNorm, đầu ra được cộng thêm với kết nối Residual từ đầu ra của lớp trước đó.

Decoder Mỗi Decoder của một Transformer block có kiến trúc khá tương đồng Encoder. Điểm khác biệt của Decoder nằm ở cơ chế Masked Attention khi sinh token và cơ chế Cross Attention hướng tới các hidden states sinh từ lớp Encoder.

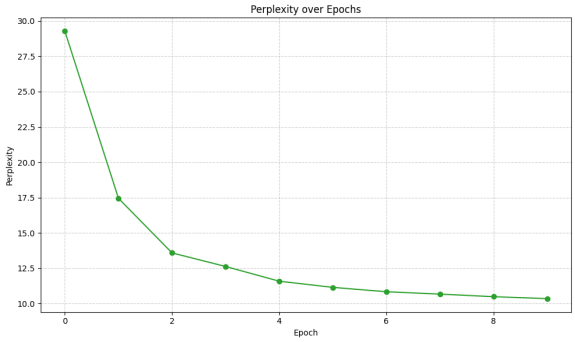
4.1.2 Huấn luyện

Bên cạnh các thành phần chính để xây dựng lên kiến trúc Transformer tiêu chuẩn, nhóm phát triển thêm các cơ chế phụ để hỗ trợ quá trình huấn luyện và suy diễn đạt kết quả cao hơn:

- **Optimizer**: Optimizer được thiết lập với các tham số theo tiêu chuẩn $\beta_1 = 0.9$, $\beta_2 = 0.98$, warmupSteps = 4000 và $\epsilon = 10^{-9}$. Tham số lr được set bằng 0.2 dựa theo số chiều đầu ra của mô hình (512)
- **Learning Rate Scheduler**: LRS được xây dựng dựa trên công thức tiêu chuẩn. Learning rate sẽ lớn trong các epoch đầu nhằm tiến đến điểm tối ưu nhanh chóng và giảm dần khi đã đến gần điểm tối ưu.
- **Regularization**: Sử dụng hai cơ chế *Dropout* và *Residual Connection*
- **Label Smoothing**: Đánh đổi giữa perplexity và BLEU score, giúp đầu ra của mô hình dịch trở nên tự nhiên hơn
- **Beam Search**: Là kỹ thuật decode thường được sử dụng nhiều nhất trong các bài toán dịch, tham số beam num = 5. Kỹ thuật decode được sử dụng trong phần tính toán BLEU score trong bước validation là Greedy Search theo (Vaswani et al., 2017)



Hình 8: Kết quả huấn luyện log theo val loss



Hình 9: Kết quả huấn luyện log theo perplexity

Mô hình được huấn luyện trong 10 epochs, dừng theo cơ chế Early Stopping khi BLEU score không cải thiện trong 3 epochs. Hàm loss được sử dụng là Cross Entropy Loss.

Chuẩn bị dữ liệu

Nhóm huấn luyện mô hình trên tập dữ liệu IWSLT, bao gồm các tệp train.en và train.vi với 133,318 cặp câu song ngữ. Dữ liệu được tiền xử lý và tokenize bằng framework en_core_web_sm của thư viện spaCy cho cả hai ngôn ngữ. Đây là một giải pháp đơn giản, được sử dụng nhằm đánh giá hiệu quả ban đầu của mô hình, do bản chất framework en_core_web_sm chỉ hoạt động tốt cho Tiếng Anh và các ngôn ngữ có nghĩa ngắt theo khoảng trắng. Tiếng Việt là ngôn ngữ có nghĩa có thể tổng hợp từ nhiều tokens khác nhau ngăn cách bởi những khoảng trắng, không phù hợp với cách tokenize của SpaCy. Để cải thiện chất lượng dịch, nhóm tiếp tục thử nghiệm một số phương pháp tiêu chuẩn cho bài toán, được trình bày trong các mục tiếp theo.

Bảng 2: Ảnh hưởng của các siêu tham số đến chất lượng dịch của mô hình Transformer trên tập IWSLT EN–VI (hiện tại là filler

N	d_{model}	h	d_{ff}	BLEU	PPL
6	512	8	2048	30.3	10.4
6	256	8	1024	23.9	22.6
6	512	4	2048	24.8	20.7
8	512	8	2048	26.1	18.3
4	1024	8	2048	**31.2**	**10.5**

4.1.3 Kết quả huấn luyện

Theo đồ thị Hình 9 và Hình 10, mô hình Transformer ban đầu đã có dấu hiệu học được trên dữ liệu IWSLT. BLEU Score đạt được tốt nhất 30.3 trên tập test, perplexity đạt mức 10.3578 dựa trên tập validation. Nhóm đã thử nghiệm và huấn luyện với một số siêu tham số khác và thu được kết quả trình bày trong Bảng 2. Đây là con số ổn định cho mô hình Transformer Base, tuy nhiên một số cải tiến có thể được phát triển để tăng hiệu năng của mô hình.

4.1.4 Phân tích và đánh giá khả năng của mô hình

Khó khăn về dịch tên riêng và thuật ngữ chuyên biệt

Mô hình gặp phải vấn đề trong việc xử lý các tên riêng, nhân vật, và các thuật ngữ cụ thể. Thay vì thực hiện phiên âm hoặc giữ nguyên, mô hình cố gắng dịch các từ này theo nghĩa đen hoặc dịch nhầm, dẫn đến mất mát thông tin cốt lõi của câu. Điều này xảy ra do các tên riêng hay thuật ngữ được tokenize qua các phương pháp như BPE hay thư viện của Spacy thường có tần suất rất nhỏ trong dữ liệu huấn luyện, dẫn đến mô hình thường coi tên riêng là một token như các từ phổ dụng và cố gắng sinh trên token lạ đó. Một số vấn đề cụ thể thường gặp là:

1. **Mất thông tin cốt lõi:** Các tên riêng như *Jatayu*, *Sita*, và *Ravana* bị dịch sai thành các cụm từ vô nghĩa hoặc chung chung (*"những con người của chúa"*, *"những con người da trắng"*).

2. **Lỗi thuật ngữ:** Thuật ngữ *Hindu mythology* bị dịch nhầm thành *"thần thoại giáo thần thoại giáo dục"*.

Ví dụ minh họa

- **Câu gốc (Source):**

  In Hindu mythology, Jatayu was the vulture god, and he risked his life in order to save the goddess Sita from the 10-headed demon Ravana.

- **Câu dịch của Mô hình (Model Output):**

  Trong thần thoại giáo thần thoại giáo dục người ta là những con người của chúa, và ông ta đã liều lĩnh cuộc sống của mình để cứu những con người da trắng từ 10 con gián đoạn.

- **Dịch chuẩn (Reference Translation):**

  Trong thần thoại Hindu , Jatayu là thần kền kền , Ông đã mạo hiểm mạng sống để cứu nữ thần Sita từ tay quỷ 10 đầu Ravana .

Đánh giá: Kết quả dịch cho thấy mô hình chưa thể phân biệt được giữa *từ thông thường* và *tên riêng/thực thể* (Named Entity Recognition - NER), dẫn đến bản dịch thiếu chính xác. Điều này đặt ra yêu cầu cần có phương pháp xử lý các tên riêng và thuật ngữ ngay khi bắt đầu tokenize.

Khó khăn về dịch các câu có tính phụ thuộc giữa các từ cách xa nhau (long dependencies)

Đồng thời, một vấn đề liên quan đến sử dụng mô hình Transformer cho bài toán là khả năng dịch yếu khi dịch các câu có độ dài lớn và phụ thuộc trải dài trong một câu.

Ví dụ minh họa

- **Câu gốc (Source):**

  A total maverick from a remote province of Afghanistan , he insisted that his daughter , my mom , go to school , and for that he was disowned by his father .

- **Câu dịch của Mô hình (Model Output):**

  Một cuộc khủng bố hoàn toàn từ một vùng xa xôi ở Afghanistan, ông ấy đề nghị con gái của mình, mẹ tôi, đi học, và vì ông ấy đã không từ chối bởi cha ông.

- **Dịch chuẩn (Reference Translation):**

  Một người hoàn toàn bị ruồng bỏ từ một tỉnh xa ở Afghanistan , ông nài nỉ để con gái ông , mẹ của tôi được đến trường , và vì vậy nên ông bị cha của ông ấy từ bỏ .

Đánh giá: Bên cạnh các lỗi về từ vựng *maverick* được dịch thành *khủng bố* và hiện tượng từ hiếm ví dụ như *disowned* do sử dụng phương pháp tokenize của Spacy, một vấn đề đáng đề cập là vấn đề dịch thuật với các cụm có phụ thuộc khoảng cách xa từ "*that*"trong "*and for that*"cần được dịch dựa theo ngữ cảnh phần phía trước giống như cách dịch chuẩn. Hai cải tiến lớn nhất có thể phát triển hiện tại là sử dụng phương pháp tokenization tiên tiến hơn, cụ thể là Sentence Piece BPE, và phương pháp mã hóa vị trí tốt hơn Positional Encoding truyền thống.

4.2 Thử nghiệm 2: Tối ưu kiến trúc SubLayer và mã hóa vị trí

4.2.1 Cải tiến

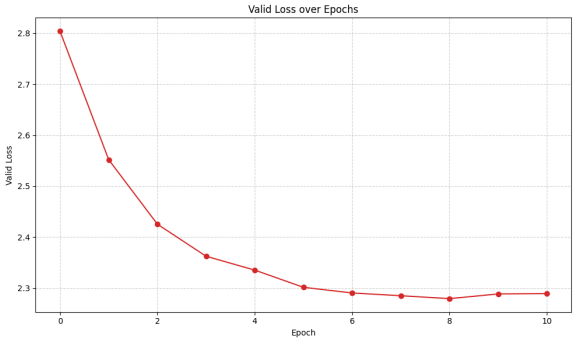
Mã hóa thông tin vị trí bằng RoPE Các biến thể của Mã hóa vị trí tuyệt đối (Absolute Positional Encoding) thông thường chỉ đơn thuần cộng kết quả đầu ra của hàm sóng hình sin (sinusoid) với các embedding của đầu vào để thêm thông tin vị trí.

Bài báo (Su et al., 2023) đã đề xuất một phương pháp tiên tiến hơn là Mã hóa vị trí tương đối bằng phép quay (Rotary Positional Embedding - RoPE), trong đó thông tin vị trí được tích hợp bằng phép nhân với đầu ra của hàm sóng hình sin. Phương pháp này đã được chứng minh bằng thực nghiệm cho kết quả BLEU score tốt hơn so với các phương pháp mã hóa vị trí tuyệt đối truyền thống như kiến trúc Base Transformer.

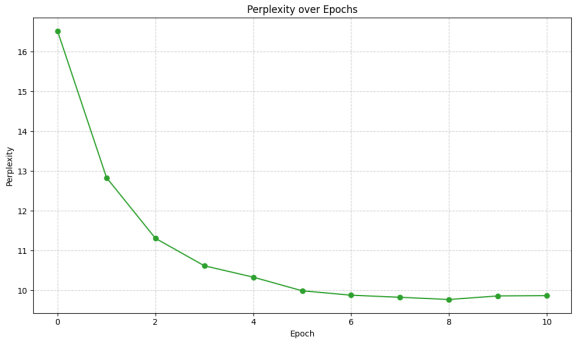
Ý tưởng cốt lõi của RoPE cũng tương đồng với tư tưởng mã hóa vị trí tương đối (Relative Positional Encoding) đã được áp dụng trong mô hình BERT. Đây là động lực chính để nhóm nghiên cứu lựa chọn tích hợp RoPE vào mô hình Base Transformer nhằm cải thiện hiệu năng.

Chuẩn hóa bằng hình thức Pre-Layer thay cho Post-Layer Đồng thời, theo bài báo (Xiong et al., 2020), việc chuẩn hóa bằng phương pháp Post-Layer Normalization, hay chuẩn hóa sau khi kết quả đã qua các lớp FFN và Attention, không tối ưu bằng phương pháp Pre-Layer Normalization do hậu chuẩn hóa được thực hiện giữa các block residual, dẫn đến gradients tại các tham số gần lớp đầu ra là rất lớn. Việc dùng learning rate lớn trong trường hợp đó sẽ khiến việc huấn luyện thiếu ổn định.

Theo thử nghiệm được thực hiện trong (Xiong et al., 2020), phương pháp Pre-Layer Norm thường đạt được kết quả tốt ngang hoặc hơn so với phương pháp Post-Layer Norm mà không cần warmup từ



Hình 10: Kết quả huấn luyện log theo val loss



Hình 11: Kết quả huấn luyện log theo perplexity

Optimizer.

Thay đổi Optimizer Để việc hội tụ diễn ra nhanh và kết quả đạt mức tối ưu, nhóm đã thay đổi tham số của Adam Optimizer lr = 1.0 thay vì 0.2 giúp việc hội tụ điểm tối ưu diễn ra nhanh hơn.

4.2.2 Kết quả huấn luyện

Theo đồ thị Hình 11 và Hình 12, tốc độ hội tụ của mô hình tới điểm tối ưu chỉ mất 2 epochs, nhờ Adam Optimizer được khởi tạo tối ưu hơn. Việc tích hợp thêm cơ chế RoPE để mã hóa vị trí và Pre-Layer Normalization cũng cho thấy cải thiện có ý nghĩa, cụ thể BLEU score mô hình cải tiến đạt được trên test set là 31.9, tăng gần 2 BLEU score so với mô hình ban đầu.

4.3 Thử nghiệm 3: Sử dụng hàm kích hoạt SwiGLU và phương pháp tokenization BPE

Sử dụng hàm kích hoạt SwiGLU Hàm kích hoạt SwiGLU thường được sử dụng trong các mô hình ngôn ngữ lớn như LLaMA, T5 v1.1 nhằm ổn định huấn luyện và được chứng minh bằng thực nghiệm trong bài báo (Shazeer, 2020) sẽ cho hiệu năng tốt hơn so với hàm kích hoạt ReLU, cụ thể biến thể SwiGLU và GEGLU được chứng minh giúp cải thiện perplexity mô hình. Theo đồ thị Hình 13, 14,

tốc độ hội tụ tới điểm tối ưu khi dùng hàm SwiGLU gần xấp xỉ so với mô hình sử dụng ReLU về tham số perplexity và val loss. Cải thiện hàm SwiGLU sẽ thực sự có ý nghĩa khi mô hình gồm nhiều lớp và sâu hơn.

Sử dụng tokenization bằng BPE Framework tokenization en_core_web_sm của thư viện SpaCy là phương pháp tokenization dựa theo từ (word-level tokenization). Đây là phương pháp thường được sử dụng khi tokenize các ngôn ngữ thuộc hệ Ấn - Âu, hay các ngôn ngữ có mỗi từ thường là một đơn vị ngữ nghĩa độc lập và không bị tách rời bởi khoảng trắng và được phân chia bằng khoảng trắng.

Ngôn ngữ Tiếng Việt không nằm trong hệ ngữ Ấn - Âu và có ngữ nghĩa của một từ phụ thuộc vào một cụm từ, thay vì một từ như Tiếng Anh, ví dụ, ta có từ *sinh viên* cần được tokenize bằng một token sinh_viên thay vì hai tokens *sinh* và *viên*.

Cách tokenize theo word của SpaCy cũng có nhược điểm trong việc xử lý các từ hiếm hay từ chưa xuất hiện trong vocab, được trình bày trong mục 4.1.4, gây khó khăn cho việc dịch các thuật ngữ chưa xuất hiện trong vocab và các tên riêng. Nhóm sử dụng phương pháp tokenization BPE theo từng từ nhỏ (subword) nhằm tránh hiện tượng OOV.

4.3.1 Phân tích và đánh giá cải tiến

Các cải tiến về phương pháp tokenization và hàm kích hoạt SwiGLU không trực tiếp cải thiện các chỉ số như val loss hay perplexity nhưng cải thiện kết quả dịch, đặc biệt là các kết quả dịch với các lỗi đã đề cập trong mục 4.1.4. Cải thiện rõ rệt nhất được thể hiện ở cách dịch các tên riêng và thuật ngữ ít xuất hiện:

Ví dụ minh họa

- **Câu gốc (Source):**

  In Hindu mythology, Jatayu was the vulture god, and he risked his life in order to save the goddess Sita from the 10-headed demon Ravana.

- **Câu dịch của Mô hình (Model Output):**

  Trong thần thoại học Ấn Độ , Jatayu là thần thánh của nhà tù. và ông ấy liều mạng sống của mình để cứu những nữ thần viên Sita từ một người có tội ác Ravana .

- **Dịch chuẩn (Reference Translation):**

  Trong thần thoại Hindu , Jatayu là thần kền kền , Ông đã mạo hiểm mạng sống để cứu nữ thần Sita từ tay quỷ 10 đầu Ravana .

Đánh giá: Lỗi về việc dịch tên riêng và thuật ngữ đã được khắc phục nhờ cơ chế tokenization bằng BPE, giúp một phần xử lý các từ hiếm và vấn đề OOV, biểu hiện qua các tên riêng như *Hindu*, *Sita* và *Ravana* được giữ chính xác vai trò. Tuy nhiên, mô hình gặp vấn đề với việc xử lý nhập nhằng về ngữ nghĩa của các từ như *vulture - kền kền* và *demon - ác quỷ*. Vấn đề này bắt nguồn từ tập dữ liệu train IWSLT được xây dựng đơn thuần trên các bài TED Talks, ít xuất hiện chủ đề tôn giáo, tâm linh; dẫn đến vấn đề bao quát chuyên ngành kém (Low Domain Coverage). Mô hình học được kết nối giữa *demon* và những người xấu, cũng như *vulture* và hàm nghĩa tiêu cực nhưng không thể dịch chính xác do bản thân hai từ này không xuất hiện trong tập train (thiên kiến phân phối).

Ví dụ minh họa

- **Câu gốc (Source):**

  A total maverick from a remote province of Afghanistan , he insisted that his daughter , my mom , go to school , and for that he was disowned by his father.

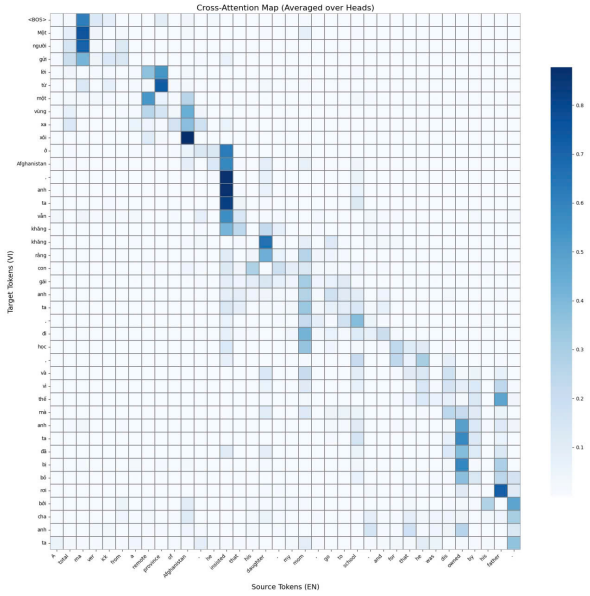
- **Câu dịch của Mô hình (Model Output):**

  Một người gửi lời từ một vùng xa xôi ở Afghanistan , anh ta vẫn khăng khăng rằng con gái anh ta , đi học , và vì thế mà anh ta đã bị bỏ rơi bởi cha anh ta .

- **Dịch chuẩn (Reference Translation):**

  Một người hoàn toàn bị ruồng bỏ từ một tỉnh xa ở Afghanistan , ông nài nỉ để con gái ông , mẹ của tôi được đến trường , và vì vậy nên ông bị cha của ông ấy từ bỏ .

Đánh giá: Bản dịch mới của mô hình được cải thiện có cách dịch tương đối tự nhiên và đúng về nghĩa hơn so với mô hình ban đầu, ngoại trừ từ hiếm *maverick*, được dịch là *người gửi lời*. Đồng thời, hiện tượng semantic underspecification cũng xuất hiện trong ví dụ này; mô hình đã bỏ qua việc



Hình 12: Ma trận attention của câu dịch ví dụ

dịch cụm *my mom*. Theo Hình 13 (trang sau), từ maverick được tách thành 3 tokens khác nhau *ma*, *ver* và *rick*, trong đó token *ma* có trọng số attention cao nhất tương quan với cụm *Một người gửi* được dịch; hai tokens *ver* và *rick* không tham gia hoặc có rất ít trong việc dịch sang cụm *người gửi lời*; do bản thân hai tokens này ít xuất hiện trong cùng ngữ cảnh với các tokens khác trên toàn tập dữ liệu. Từ *disowned* được tách chính xác thành hai tokens *dis* và *owned*, nhờ ngữ nghĩa của hai tokens *dis* - mang nghĩa phủ định và *owned* - nghĩa là sở hữu để dịch disowned thành *bỏ rơi* thay vì *từ chối* giống như khi dùng cơ chế tokenization của SpaCy.

5 Task 2: VLSP 2025 - Dịch máy Y tế

5.1 Mô tả bài toán

Nhiệm vụ tại VLSP 2025 yêu cầu dịch hai chiều Anh-Việt ($en \leftrightarrow vi$) trên dữ liệu y tế. Thách thức chính bao gồm:

- **Thiếu kiến thức nền:** Các mô hình nhỏ thường thiếu hiểu biết sâu về thế giới thực so với các mô hình lớn (VLSP Organizers, 2025).
- **Thuật ngữ chuyên ngành:** Văn bản y tế chứa nhiều từ viết tắt, tên thuốc, tên bệnh dễ gây ra lỗi Out-of-Vocabulary (OOV) (VLSP Organizers, 2025).
- **Ràng buộc tài nguyên:** Mô hình cơ sở (Base LLM) cố định thuộc họ Qwen 2.5 hoặc Qwen 3 với tối đa 3 tỷ tham số (3B parameters) (VLSP Organizers, 2025).

Bảng 3: Tham số kiến trúc mô hình tốt nhất

Thành phần	**Tham số**
Kiến trúc	Encoder–Decoder
Số lớp Encoder	6
Số lớp Decoder	6
Số chiều mô hình (d_{model})	512
Số chiều lớp FFN (d_{ff})	2048
Số đầu chú ý	8
Mã hóa vị trí	RoPE
Hàm kích hoạt trong FFN	SwiGLU
Dropout	0.1
Vocabulary	BPE (joint)

Bài toán VLSP đặt ra thách thức lớn đối với các mô hình Transformer được huấn luyện từ đầu, do bản chất bài toán dịch thuật trong lĩnh vực y tế cần sự chính xác cao về thuật ngữ và súc tích trong cách diễn đạt.

Mô hình Transformer có thể đạt được kết quả tốt trong các bài toán dịch thuật ưu tiên độ trôi chảy (fluency) và tự nhiên hơn độ chính xác về thuật ngữ, nhưng sẽ gặp một số khó khăn nhất định khi dịch các văn bản y khoa. Trong thực tiễn, phương pháp transfer learning và finetuning thường được ưu tiên đối với các bài toán yêu cầu tri thức chuyên ngành cao nhưng giới hạn về dữ liệu như VLSP.

5.2 Áp dụng kiến trúc Transformer đã xây dựng vào bài toán VLSP

5.2.1 Khái quát kiến trúc và tham số mô hình

Đối với phần một của task 2, nhóm chọn mô hình được cải tiến có BLEU score và độ ổn định tốt nhất từ task 1 để huấn luyện trên dữ liệu VLSP. Các tham số và kiến trúc của mô hình được mô tả trong Bảng 3.

Nhóm sử dụng mô hình Transformer theo kiến trúc encoder-decoder gồm 6 lớp mỗi bộ phận. Mỗi lớp sử dụng cơ chế multi-head self-attention, kèm theo một lớp FFN. Tại mỗi lớp FFN, hàm kích hoạt được sử dụng là SwiGLU và đầu ra được chuẩn hóa bằng cơ chế Pre-Layer Normalization và mã hóa vị trí được thực hiện bằng RoPE thay cho sử dụng sinusoidal truyền thống.

Các siêu tham số được giữ nguyên tương đối so với mục 4.1.2 ngoại trừ sự thay đổi về tham số lr.

Cơ chế tokenization được sử dụng là SentencePiece BPE. Đây là cơ chế tiêu chuẩn được sử dụng trong bài báo (Vaswani et al., 2017), được

sử dụng nhằm hạn chế tình trạng OOV và hiện tượng từ hiếm, các thuật ngữ chuyên ngành xuất hiện với tần suất ít trong dữ liệu huấn luyện; một vấn đề dễ xảy ra đối với phạm trù y tế.

5.2.2 Phân tích và đánh giá bản dịch của mô hình Transformer

Nhìn chung, mô hình được huấn luyện có hiệu năng khá ổn định trong việc dịch các văn bản y tế. Việc sử dụng tập dữ liệu VLSP với kích thước lớn hơn đáng kể so với tập dữ liệu IWSLT mang lại lợi ích rõ rệt. Kích thước dữ liệu lớn hơn đã góp phần giảm thiểu đáng kể hiện tượng từ hiếm (rare words) và tỷ lệ từ ngoài từ điển (Out-Of-Vocabulary - OOV).

Ví dụ bản dịch và các lỗi phổ biến

Lỗi về ngữ pháp và các thuật ngữ hiếm và khó phân tách

- **Câu gốc (Source):**

  The purpose of this study was to evaluate the effects of a mixture extract of C chrysantha and G pentaphyllum on weight loss and lowering lipid blood levels in obese Swiss mice.

- **Câu dịch của Mô hình (Model Output):**

  Mục đích của nghiên cứu này là đánh giá tác động của hỗn hợp cao phối hợp C. chrysantha và G. tuần hoàn trong việc giảm cân và hạ lipid máu trên chuột nhắt trắng bị béo phì.

- **Dịch chuẩn (Reference Translation):**

  Nghiên cứu được thực hiện nhằm đánh giá tác dụng giảm cân, hạ lipid máu của hỗn hợp dịch chiết lá Trà hoa vàng và Giảo cổ lam trên chuột nhắt trắng gây béo phì.

Bản dịch trên cho thấy, mô hình chưa thể học được sự liên hệ giữa cách viết tên khoa học *C. chrysantha* và *G pentaphyllum* và tên Tiếng Việt như *Trà hoa vàng* và *Giảo cổ lam*. Các token được tạo ra thông qua Byte Pair Encoding (BPE) từ các cụm từ có cấu trúc đặc biệt (như chrysantha và pentaphyllum) có phân phối xác suất (distribution) trong tập dữ liệu không đủ cao so với các từ vựng phổ thông, dẫn đến thiên kiến phân phối (distributional bias) và bản dịch không chính xác như *G. tuần hoàn*.

Ngoài ra bản dịch cũng dịch sai cụm *hỗn hợp cao phối hợp*. Đây là lỗi sai về từ vựng xảy ra do thiên kiến phân phối (distributional bias). Mô hình học được việc dịch *hỗn hợp* và *cao* (có thể trong các cụm như *hỗn hợp liều cao*) sẽ gây loss nhỏ hơn khi sinh token, dẫn đến nhầm lẫn khi dịch.

Tuy nhiên, mô hình có thể làm tốt nhiệm vụ dịch trong các trường hợp từ vựng được cấu tạo từ các thành tố (dạng composition). Đây là dạng từ vựng khá phổ biến trong lĩnh vực y tế, các tên bệnh như: cardiovascular disease (bệnh tim mạch), hepatitis (bệnh viêm gan) được cấu thành từ các hình vị (morpheme) tương đồng nhau, có nguồn gốc từ tiếng Hy Lạp hoặc La-tinh; hay các thuật ngữ viết tắt như: PPI (thuốc ức chế bơm proton), LDL-C (LDL - Cholesterol) có thể được phân tách thành các subwords một cách hiệu quả nhờ BPE nhờ các hình vị thường xuyên xuất hiện trong các từ vựng khác nhau và giữ nguyên ngữ nghĩa của nó (cardio-: liên quan đến tim, hepa-: liên quan đến gan).

Ví dụ minh họa

- **Câu gốc (Source):**

  Mice in each group was assessed for weight weekly and the levels of Total Cholesterol (CT), HDLCholesterol (HDL-C), LDL-Cholesterol (LDL-C) and Triglyceride (TC) was recorded at initial time (after obesity was induced for 8 weeks) and 1 hour after taking the extracted mixtures on the last day.

- **Câu dịch của Mô hình (Model Output):**

  Mỗi nhóm được đánh giá cân nặng hàng tuần và nồng độ Cholesterol toàn phần (CT), HDL cholesterol (HDL-C), LDL-C (LDL-C) và triglyceride (TC) được ghi nhận vào thời điểm ban đầu (sau khi gây béo phì trong 8 tuần) và 1 giờ sau khi uống hỗn hợp dịch chiết vào ngày cuối.

- **Dịch chuẩn (Reference Translation):**

  Trọng lượng chuột ở mỗi lô được đánh giá hàng tuần và hàm lượng Cholesterol toàn phần (CT), HDL-Cholesterol (HDL-C), LDL-Cholesterol (LDL-C) và Triglycerid

(TC) tại các thời điểm chưa uống thuốc (sau gây béo phì 8 tuần) và sau uống thuốc thử ngày cuối 1 giờ.

Trong ví dụ minh họa, bản dịch của mô hình có tính tự nhiên và dễ hiểu hơn so với bản dịch tham chiếu. Mô hình Transformer dịch giữ nguyên thứ tự đầu vào và cách dịch word-by-word của mô hình trong trường hợp này dễ hiểu và sát nghĩa gốc hơn, cụ thể ở cụm *và 1 giờ sau khi uống hỗn hợp dịch chiết vào ngày cuối.*. Bản dịch tham chiếu nên đặt bổ ngữ *ngày cuối* sau *1 giờ* để rõ hơn về ngữ nghĩa.

Khó khăn về các bài toán dịch yêu cầu tính tổng hợp cấu thành cao (weak compositionality)

Theo thử nghiệm, mô hình Transformer có xu hướng ánh xạ và dịch các cụm từ theo từng đơn vị từ (word-wise). Điều này dẫn đến các cụm từ được dịch có ý nghĩa một cách cục bộ (local) nhưng không có ý nghĩa kết nối với nhau trong một câu dài. Việc mô hình được huấn luyện bằng hàm loss Cross Entropy khuyến khích mô hình sinh các token tốt nhất trong một phạm vi nhỏ mà không quan tâm đến ngữ nghĩa trên toàn câu.

Ví dụ minh họa

- **Câu gốc (Source):**

  Conclusion: The proportion of proton pump inhibitors was not safe and reasonable and the proportion of prescription drugs with no instructions on how long to use proton pump inhibitors were low. The proportion of prescription interacting drugs accounted for a high proportion, clopidogrel was the most interactive drug commonly used with PPIs.

- **Câu dịch của Mô hình (Model Output):**

  Kết luận: Tỷ lệ sử dụng thuốc ức chế bơm proton không an toàn và hợp lý và tỷ lệ thuốc theo toa không có hướng dẫn sử dụng thuốc ức chế bơm proton còn thấp, tỷ lệ các thuốc có tác dụng tương tác thuốc chiếm tỷ lệ cao, clopidogrel là thuốc tương tác được sử dụng phổ biến nhất với PPI.

- **Dịch chuẩn (Reference Translation):**

  Kết luận: Tỷ lệ thuốc ức chế bơm proton chưa an toàn, chưa hợp lý, đơn thuốc không có hướng dẫn thời gian sử dụng thuốc ức chế bơm proton chiếm tỷ lệ thấp. Tỷ lệ đơn thuốc có tương tác chiếm tỷ lệ cao, clopidogrel là thuốc tương tác được sử dụng chung với PPI nhiều nhất.

Bản dịch cho thấy một số lỗi đặc trưng khi dịch của mô hình Transformer. Cụ thể:

Negation Scope : Lỗi thể hiện ở cụm *was not safe and reasonable*. Theo cách chúng ta phủ định trong ngôn ngữ nói, từ *not* sẽ áp dụng cho cả *safe* và *reasonable*; bản dịch nên là *không an toàn và không hợp lý*. Tuy nhiên, do mô hình Transformer decode theo từng tokens, tại mỗi bước token được sinh để giảm thiểu hàm loss Cross Entropy tại một thời điểm sinh token, dẫn đến mô hình thường dịch word-by-word và tránh sinh thêm dù việc sinh thêm từ sẽ cải thiện ngữ nghĩa của câu.

Semantic underspecification : Lỗi thể hiện ở các cụm dịch *with no instructions on how long to use - không có hướng dẫn sử dụng* thay vì *không có hướng dẫn thời gian sử dụng*. Hiện tượng mất mát thông tin này xảy ra do bổ ngữ *how long to* bị lược bớt. Mô hình Transformer có xu hướng lược bớt các bổ ngữ do bản thân các từ này không có xác suất xuất hiện lớn, và hàm loss Cross Entropy không có giá trị quá lớn nếu token được sinh ra không đầy đủ. Việc sinh tokens *hướng dẫn sử dụng* an toàn hơn và gây mất mát thấp hơn so với sinh cụm *hướng dẫn thời gian sử dụng* trong nhiều trường hợp; vì vậy, mô hình học cách sinh cụm có xác suất xuất hiện nhiều hơn như *hướng dẫn sử dụng* trong hầu hết các ngữ cảnh.

Terminology mistranslation Việc dịch sai thuật ngữ ở cụm dịch *prescription interacting drugs - thuốc có tác dụng tương tác thuốc* xảy ra do cơ chế attention hiểu từ prescription là *thuốc* trong trường hợp này, thay vì là *đơn thuốc* hay *toa thuốc*. Cụm từ *prescription interacting drugs* vì vậy được dịch thành *thuốc có tác dụng tương tác thuốc* do ảnh hưởng của token *thuốc* trong Tiếng Việt lớn hơn *đơn* hay *toa* trong ngữ cảnh này.

5.2.3 Nhận xét và đánh giá mô hình Transformer

Mô hình Transformer có khả năng dịch tương đối tốt với các bài toán dịch có liên quan đến kiến thức chuyên ngành, cụ thể là lĩnh vực y tế khi các cụm từ chuyên ngành được thể hiện tốt trong dữ liệu; nhưng sẽ gặp khó khăn với việc dịch cho chuẩn về

ngữ nghĩa hay với việc dịch bảo đảm sự đồng nhất trong ngữ nghĩa của một câu dài.

Điểm mạnh Một số điểm mạnh của mô hình Transformer có thể kể đến như:

- **Cách dịch trôi chảy và các cú pháp ổn định trong một câu** nhờ cơ chế attention có thể xử lý dữ liệu chuỗi dài và khả năng mô hình hóa ngôn ngữ của decoder
- **Dịch chính xác các cụm từ chuyên ngành thường xuyên xuất hiện** nhờ việc học theo phân phối của mô hình
- **Khả năng mở rộng khi bổ sung hoặc tăng cường dữ liệu**

Điểm yếu Tuy nhiên, mô hình Transformer cũng gặp phải một số khó khăn nhất định trong bài toán dịch

- **Biểu diễn thiếu ngữ nghĩa**: Mô hình loại bỏ những bổ ngữ cần thiết nhằm tối ưu hàm loss cũng như dịch một cụm từ dài, có ý nghĩa thành một từ ngắn, sai về ý nghĩa nhưng có xác suất đủ lớn để hàm loss không quá cao.
- **Hiện tượng mất mát thông tin**: được trình bày trong ví dụ minh họa thứ hai tại cụm dịch thiếu bổ ngữ *thời gian* do xu hướng của mô hình dễ bỏ qua các bổ ngữ có ảnh hưởng nhỏ trong nghĩa chung trong câu nhưng mang ý nghĩa lớn cục bộ.
- **Dịch theo thứ tự của chuỗi đầu vào và không đổi mới về cấu trúc câu** do việc dự đoán vào các cụm có xác suất lớn thường ít rủi ro hơn các cụm hiếm xuất hiện nhưng đúng trong ngữ cảnh đó; dẫn đến mô hình lệ thuộc vào các cụm có cấu trúc an toàn.

5.2.4 Kết quả huấn luyện

Mô hình Transformer dịch một chiều EN-VI đạt được kết quả khả quan với các chỉ số BLEU, TER và METEOR được báo cáo như trong Bảng 4

Bảng 4: Đánh giá mô hình Transformer trên tập test

Metric	BLEU	TER	METEOR
Score	35.2	46.9	0.66

Các chỉ số cho thấy hiệu năng ổn định của mô hình xét về độ chính xác và trôi chảy khi dịch, thể hiện qua điểm BLEU cao và điểm TER tương đối thấp. Điểm METEOR đạt mức 0.66, thể hiện tính công thức trong cách dịch và các cụm từ chuyên ngành xuất hiện lại nhiều lần trong lĩnh vực y tế.

Nhóm quyết định dừng việc phát triển Transformer tại bài toán dịch một chiều EN-VI, do đặc tính mô hình Transformer không phù hợp với bài toán dịch hai chiều. Về mặt lý thuyết, mô hình Transformer có thể được huấn luyện cho bài toán dịch hai chiều nhờ cơ chế attention linh hoạt và mục tiêu mô hình hóa xác suất phân phối của các chuỗi, không phụ thuộc vào chiều dịch.

Tuy nhiên, các lỗi đã được phân tích và đánh giá trong mục 5.2.2 cho thấy những khó khăn nhất định của mô hình Transformer cho bài toán dịch yêu cầu độ chính xác cao và ngữ nghĩa trong một câu phức tạp như VLSP. Yêu cầu về tính tổng hợp cấu thành cao của bài toán VLSP trình bày trong mục 5.2.2 là một khó khăn lớn khi thực hiện dịch hai chiều. Dữ liệu huấn luyện có thể không đối xứng (asymmetric), dẫn đến mô hình ưu tiên học theo hướng dịch có dữ liệu nhiều và sạch hơn, gây ra tình trạng *semantic underspecification* trầm trọng hơn ở chiều dịch có dữ liệu nhiễu và nhỏ hơn.

5.3 Transfer Learning và Fine-tuning Methods

5.4 Dữ liệu và tiền xử lý

Dữ liệu y tế chuyên biệt theo domain cụ thể sẽ gây khó khăn cho mô hình với các từ ngữ chuyên ngành các số liệu, chữ số có thể gây nhiễu và gây hallucation cho các mô hình pretrained.

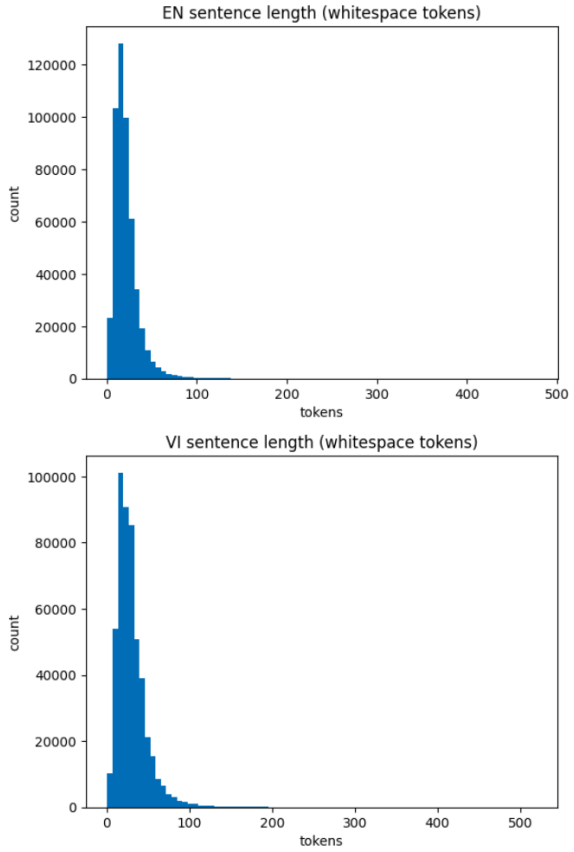
Phân tích thống kê sơ bộ. Nhóm thực hiện một số phân tích ban đầu nhằm hiểu rõ đặc điểm dữ liệu: **Độ dài câu:** Phần lớn các câu có độ dài phân phối trong khoảng 0-100 từ, 1 phần nhỏ khác có độ dài đột biến rõ rệt (Hình 13. Độ dài tối đa sau khi tokenize được được sử dụng là 128.

Tên riêng và thuật ngữ y khoa: Dữ liệu chứa nhiều thực thể đặc thù như tên bệnh, thuốc, xét nghiệm và cơ quan giải phẫu, đòi hỏi mô hình phải duy trì tính nhất quán và chính xác khi dịch. Tokenization bằng bộ tokenizer tương ứng với từng mô hình để đảm bảo tính nhất quán giữa huấn luyện và suy luận.

5.5 Tinh chỉnh Qwen2.5 với LoRA và Instruction Prompting

Nhóm sử dụng mô hình ngôn ngữ lớn Qwen/Qwen2.5-0.5B-Instruct [3] theo kiến

[3] https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct



Hình 13: Thống kê độ dài câu (word split) trong tập train

Pattern	Count_EN	Count_VI
Chữ viết hoa / Tên riêng (EN) – Acronym (VI)	1106116	236905
Chữ số	650740	670881
Đơn vị y khoa (mg, ml, mmHg, ...)	21056	21956
Phần trăm (%)	5	14

Hình 14: Thống kê số lượng ký tự chuyên ngành

trúc decoder-only làm nền tảng cho bài toán dịch máy y tế. Thay vì tinh chỉnh toàn bộ tham số, nhóm áp dụng kỹ thuật **LoRA (Low-Rank Adaptation)** nhằm giảm chi phí tính toán và bộ nhớ trong quá trình huấn luyện.

Thiết lập tinh chỉnh Mô hình Qwen2.5B-Instruct được tinh chỉnh trên tập con 150K cặp câu từ dữ liệu huấn luyện VLSP-2025 trong một epoch, sử dụng thư viện HuggingFace transformers. Quá trình huấn luyện được thực hiện trên hai GPU NVIDIA T4 với batch size 4 mỗi GPU và 8 bước gradient accumulation, tương ứng batch hiệu dụng 32 mẫu/GPU. Mô hình được tối ưu bằng AdamW với learning rate 2×10^{-4} với 500 warm-up steps. Để giảm chi phí bộ nhớ, nhóm sử dụng huấn luyện FP16 và gradient checkpointing. Việc đánh giá và lưu checkpoint được thực hiện định kỳ theo số bước trong quá trình huấn luyện.

Tinh chỉnh tham số hiệu quả với LoRA Nhóm áp dụng kỹ thuật Low-Rank Adaptation (LoRA) nhằm thích nghi mô hình với miền y tế mà không cần cập nhật toàn bộ tham số. Các adapter LoRA được cấu hình với hạng $r = 16$, hệ số $\alpha = 32$ và dropout 0.05. Tổng tham số sử dụng để finetune là 8,798,208 / 502,830,976 (1.7497%) params.

Thiết kế câu lệnh (instruction) Dữ liệu huấn luyện được chuyển đổi sang định dạng hội thoại để phục vụ quá trình tinh chỉnh theo hướng dẫn (instruction tuning). Mỗi mẫu huấn luyện được cấu trúc bao gồm một lệnh hệ thống (*system prompt*) nhằm thiết lập vai trò chuyên gia dịch thuật y khoa và một lệnh người dùng (*user prompt*) mô tả chi tiết tác vụ dịch thuật cụ thể.

Cấu trúc chi tiết của bộ khung câu lệnh (instruction template) và các ví dụ minh họa về định dạng dữ liệu đầu vào được trình bày cụ thể tại **Phụ lục A**.

Mặc dù kiến trúc decoder-only không được thiết kế chuyên biệt cho dịch máy, kết quả thực nghiệm cho thấy khi kết hợp instruction prompting và LoRA, mô hình Qwen có khả năng học tốt các mẫu dịch y khoa. Tuy nhiên, chất lượng dịch vẫn phụ thuộc đáng kể vào thiết kế prompt và phân phối độ dài câu trong dữ liệu huấn luyện.

Bảng 5: Kết quả mô hình Qwen2.5-0.5B-Intruct (public-test)

Metric	EN→VI		VI→EN	
	Base	+150K	Base	+150K
BLEU	12.75	30.12	9.09	11.23
TER	–	68.14	–	97.88
METEOR	–	54.72	–	36.34
ROUGE-L	–	56.31	–	36.06

5.6 Tinh chỉnh mô hình VietAI Translation

Mô hình envit5-translation [4] của VietAI là một hệ dịch máy Anh–Việt dựa trên kiến trúc encoder–decoder kiểu T5, đã được tiền huấn luyện và tinh chỉnh trên các tập dữ liệu song ngữ quy mô lớn, và được báo cáo đạt hiệu năng cạnh tranh trong cả hai chiều dịch Anh–Việt và Việt–Anh trong các nghiên cứu trước đây. Các kết quả này cho thấy mô hình có khả năng học tốt cấu trúc ngữ nghĩa và cú pháp của hai ngôn ngữ, đặc biệt phù hợp cho các tác vụ

[4] https://huggingface.co/VietAI/envit5-translation

dịch máy chuyên ngành khi được tinh chỉnh thêm với dữ liệu miền.

Trong thực nghiệm ban đầu của nhóm trên tập dữ liệu y khoa VLSP, mô hình VietAI Translation đã thể hiện hiệu năng ổn định. Kết quả so sánh giữa mô hình gốc (Base) và sau khi bổ sung dữ liệu được trình bày tại Bảng 6.

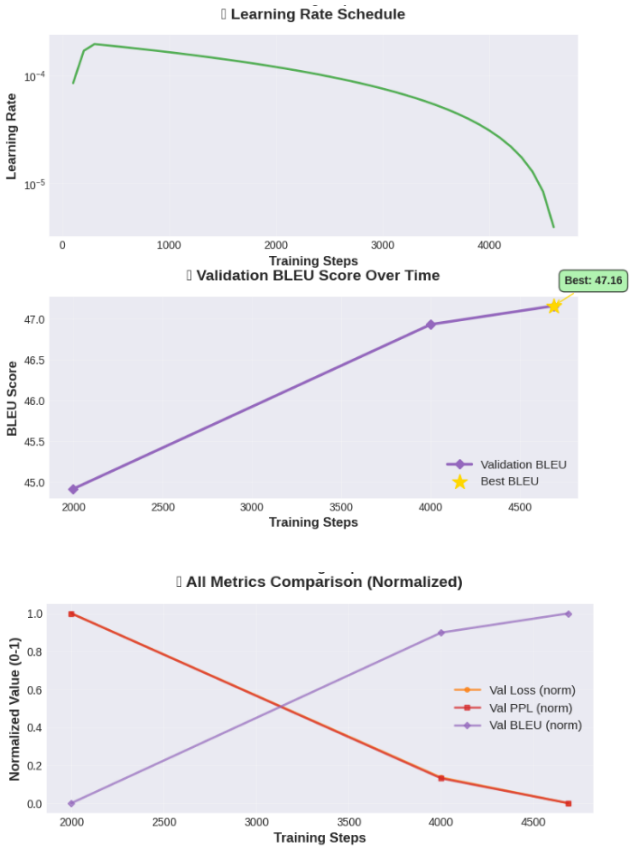
Các kết quả này phản ánh sự chênh lệch độ khó giữa hai chiều dịch, đồng thời khẳng định VietAI Translation là một baseline mạnh cho bài toán dịch máy y tế Anh–Việt trong khuôn khổ VLSP.

Dựa trên nền tảng đó, nhóm tiến hành tinh chỉnh mô hình envit5-translation theo thiết lập sequence-to-sequence chuẩn, không sử dụng instruction prompting. Để hỗ trợ dịch hai chiều, tập huấn luyện được nhân đôi theo hai hướng EN→VI và VI→EN, tạo thành 300,000 mẫu huấn luyện và 6,000 mẫu kiểm tra. Tổng số 275,102,976 tham số được training.

Mô hình có 275,102,976 tham số và toàn bộ tham số đều được cập nhật trong fine-tuning. Nhóm sử dụng bộ tối ưu AdamW với learning rate 2×10^{-4}, weight decay 0.01, $\beta = (0.9, 0.999)$ và $\epsilon = 10^{-8}$. Batch size được đặt là 32, số epoch là 1 và gradient accumulation là 2, dẫn đến tổng số bước cập nhật là 4,687; trong đó warm-up chiếm 5% tương ứng 234 bước. Learning rate được điều chỉnh bằng scheduler tuyến tính có warm-up. Độ dài chuỗi tối đa được giới hạn ở 128 token để đảm bảo ổn định bộ nhớ và tốc độ. Để tránh bùng nổ gradient, nhóm áp dụng gradient clipping với max_grad_norm=1.0.

5.6.1 Kết quả huấn luyện trên 150.000 mẫu đầu tiên

Sau khi hoàn thành giai đoạn huấn luyện với 150.000 mẫu dữ liệu đầu tiên, nhóm đã tiến hành đánh giá quá trình hội tụ của mô hình thông qua các chỉ số: Learning rate, BLEU score và Perplexity.



Hình 15: Diễn biến các chỉ số đánh giá trong quá trình tinh chỉnh mô hình VietAI với 150k mẫu.

Dựa trên Hình 15, ta có thể thấy chỉ số BLEU tăng trưởng đáng kể. Chỉ số Perplexity thấp cho thấy mô hình đã học được xác suất phân phối từ vựng chuyên ngành y tế một cách hiệu quả.

5.6.2 Phân tích định tính kết quả dịch thuật

Khả năng chuyển đổi thuật ngữ chuyên ngành: Mô hình thể hiện khả năng dịch thuật ngữ y sinh rất chính xác. Các thuật ngữ như *"proton pump inhibitors"* (thuốc ức chế bơm proton), *"obesity was induced"* (gây béo phì) hay các chỉ số hóa sinh (HDL-C, LDL-C) được giữ nguyên định dạng hoặc dịch đúng nghĩa chuyên môn.

Cấu trúc ngữ pháp và ngữ cảnh: Ở chiều EN→VI, mô hình biết cách chuyển đổi linh hoạt cụm từ *"mice in each group"* thành *"chuột trong mỗi lô"* (thuật ngữ thường dùng trong thí nghiệm thay vì dịch word-by-word là "nhóm"). Ở chiều VI→EN, mô hình đã tự động điều chỉnh sự hòa hợp giữa chủ ngữ và động từ (*"were evaluated"* thay vì dùng số ít như câu gốc tiếng Anh ở VD2), cho thấy khả năng làm mượt văn bản tốt.

Các lỗi tồn tại (Error Analysis):

Lỗi lặp từ và ngắt nghỉ: Trong Ví dụ 1, câu dịch

Bảng 6: So sánh kết quả mô hình VietAI Translation theo quy mô dữ liệu huấn luyện trên tập VLSP-2025 (public-test)

Metric	EN→VI				VI→EN			
	Base	**+150K**	**+300K**	**+500K**	**Base**	**+150K**	**+300K**	**+500K**
BLEU	40.75	45.66	47.02	**48.80**	24.09	34.50	35.02	**38.62**
TER	55.25	47.23	49.10	**45.11**	88.14	61.03	63.23	**56.99**
METEOR	65.14	69.90	70.77	**72.04**	59.00	64.22	**68.72**	66.30
ROUGE-L	66.74	70.98	71.08	**72.73**	55.13	62.70	63.70	**64.94**

Bảng 7: Ví dụ kết quả dịch thuật thực tế từ mô hình VietAI

STT	Câu gốc (Source)	Kết quả mô hình (VietAI Output)
1 (EN-VI)	Conclusion: The proportion of proton pump inhibitors was not safe and reasonable and the proportion of prescription drugs with no instructions on how long to use proton pump inhibitors were low. The proportion of prescription interacting drugs accounted for a high proportion, clopidogrel was the most interactive drug commonly used with PPIs.	Kết luận: Tỷ lệ thuốc ức chế bơm proton chưa an toàn và hợp lý và tỷ lệ thuốc kê đơn không có hướng dẫn sử dụng thuốc ức chế bơm proton còn thấp, tỷ lệ thuốc tương tác đơn chiếm tỷ lệ cao, clopidogrel là thuốc tương tác nhiều nhất được sử dụng với PPIs.
2 (EN-VI)	Mice in each group was assessed for weight weekly and the levels of Total Cholesterol (CT), HDL-Cholesterol (HDL-C), LDL-Cholesterol (LDL-C) and Triglyceride (TC) was recorded at initial time (after obesity was induced for 8 weeks) and 1 hour after taking the extracted mixtures on the last day.	Chuột trong mỗi lô được đánh giá cân nặng hàng tuần và ghi nhận các chỉ số Cholesterol toàn phần (CT), HDL-Cholesterol (HDL-C), LDL-Cholesterol (LDL-C) và Triglyceride (TC) ở thời điểm ban đầu (sau khi gây béo phì 8 tuần) và 1 giờ sau khi uống hỗn hợp cao chiết vào ngày cuối cùng.
3 (VI-EN)	Chuột trong mỗi lô được đánh giá cân nặng hàng tuần và ghi nhận các chỉ số Cholesterol toàn phần (CT), HDL-Cholesterol (HDL-C), LDL-Cholesterol (LDL-C) và Triglyceride (TC) ở thời điểm ban đầu (sau khi gây béo phì 8 tuần) và 1 giờ sau khi uống hỗn hợp cao chiết vào ngày cuối cùng.	Mice in each group were evaluated weekly weight and recorded total cholesterol (CT), HDL-Cholesterol (HDL-C), LDL-C and Triglyceride (TC) at baseline (after 8 weeks of obesity) and 1 hour after drinking the extract mixture on the final day.

tiếng Việt có cấu trúc hơi dài và thiếu dấu chấm phẩy giữa các mệnh đề độc lập.

Từ vựng: Ở Ví dụ 3, các cụm từ chuyển ngành khi dịch từ tiếng Việt sang Anh bị thiếu từ và các từ dịch chưa sát nghĩa (HDL-Cholesterol (HDL-C) dịch ra thành ((HDL-C).

5.7 Đánh giá mô hình (VLSP Medical MT Task)

Hiệu năng của các mô hình được đánh giá định lượng thông qua SacreBleu trên cả hai chiều dịch Anh–Việt (EN–VI) và Việt–Anh (VI–EN). Kết quả thực nghiệm đối với ba kiến trúc mô hình tiêu biểu được trình bày tại Bảng 8.

Bảng 8: So sánh chỉ số BLEU giữa các kiến trúc mô hình trên tập dữ liệu y khoa VLSP

Mô hình	Tham số	EN→VI	VI→EN
Transformer-base	65M	35.2	–
Qwen2.5-1.5B	0.5B	30.12	11.23
VietAI-enviT5	275M	**48.80**	**38.62**

Dựa trên kết quả thực nghiệm, mô hình VietAI-enviT5 mặc dù có số lượng tham số nhỏ hơn đáng kể so với dòng Qwen nhưng lại cho kết quả BLEU cao nhất ở cả hai chiều dịch. Điều này cho thấy lợi thế của việc sử dụng các mô hình đã được tiền huấn luyện chuyên biệt cho cặp ngôn ngữ Anh–Việt trong các tác vụ dịch máy đặc thù.

5.8 Tổng kết

Hai hướng tiếp cận được trình bày trong nghiên cứu này mang tính bổ sung cho nhau. Mô hình VietAI Translation cung cấp một baseline mạnh, chuyên biệt cho dịch máy Anh–Việt trong miền y tế. Trong khi đó, Qwen2.5-0.5B kết hợp LoRA và instruction prompting cho thấy tiềm năng lớn trong việc tận dụng các mô hình ngôn ngữ lớn với chi phí tính toán hạn chế. Kết quả này gợi mở hướng phát triển các hệ thống dịch máy chuyên ngành linh hoạt, phù hợp với bối cảnh tài nguyên hạn chế và yêu cầu thực tiễn của VLSP 2025.

References

2025. Văn bản kĩ thuật - viettel ai race (mô tả dữ liệu và thách thức đa ngôn ngữ/ký hiệu). Tài liệu PDF do người dùng cung cấp trong dự án.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *EAMT*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, and Ting Liu. 2020. Codebert: A pre-trained model for programming and natural languages. *arXiv preprint arXiv:2002.08155*.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1).

Alistair E. W. Johnson, Tom J. Pollard, Li Shen, Lucas-weyh Lea Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, David Anthony, Leo A. Celi, and 1 others. 2021. Mimic-iv, version 2.2. *PhysioNet*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP: System Demonstrations*.

Zhenzhong Lan, Ming Chen, Steven Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biolbert: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Myle Ott, Sergey Edunov, Alexei Baevski, and 1 others. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL-HLT: Demonstrations*.

Loc Phan, Hieu Tran Sun, Heng Li, Samarth Gupta, Yi Ma, Christof Wood, Genevieve Feild, Ramesh Nallapati, and Bing Xiang. 2021. Cotext: Multi-generated context expansion for unsupervised code translation. *arXiv preprint arXiv:2108.06215*.

Long Phan, Tai Dang, Hieu Tran, Trieu H. Trinh, Vy Phan, Lam D. Chau, and Minh-Thang Luong. 2022. Enriching biomedical knowledge for low-resource language through large-scale translation. *arXiv preprint*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Noam Shazeer. 2020. Glu variants improve transformer. arXiv preprint arXiv:2002.05202v1.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864v5.

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

VLSP Organizers. 2025. VLSP 2025 Challenge on Medical Domain MT with Limited-Pretraining Models. Association for Vietnamese Language and Speech Processing.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On layer normalization in the transformer architecture. arXiv preprint arXiv:2002.04745v2.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426.

A Qwen Prompt Instruction

Bảng 9: Cấu trúc thiết kế Instruction Tuning cho mô hình Translation

System Role: *You are a professional medical translator specialized in clinical and biomedical text. Follow these rules:* *(1) Translate faithfully without adding or omitting information.* *(2) Preserve numbers, units, dosage, and formatting.* *(3) Use standard medical terminology and keep abbreviations as in the source.* *(4) Do not add explanations or comments.* *(5) Output only the translation.*
Query Format: *Translate from English to Vietnamese.* *Source (English): [Source sentence]* *Translation (Vietnamese): [Target sentence]*