Xây dựng và tối ưu hệ thống dịch máy Anh - Việt trên miền dữ liệu chuyên ngành y khoa

Nguyễn Hoàng Tú
Trường Đại học Công nghệ - ĐHQGHN
Viện Trí Tuệ Nhân tạo
23020428@vnu.edu.vn

Nguyễn Khánh Tùng
Trường Đại học Công nghệ - ĐHQGHN
Viện Trí Tuệ Nhân Tạo
23020434@vnu.edu.vn

Tóm tắt

Báo cáo này trình bày tổng quan quá trình xây dựng và tối ưu hiệu suất của hệ thống dịch máy từ tiếng Anh sang tiếng Việt trên tập dữ liệu văn bản chuyên ngành y khoa. Nghiên cứu được triển khai theo ba hướng tiếp cận: Xây dựng thủ công và huấn luyện từ đầu mô hình Transformer cơ bản trên tập dữ liệu y khoa đã được chuẩn bị; hướng đi thứ hai là cải tiến kiến trúc Transformer gốc và huấn luyện từ đầu mô hình trên cùng tập dữ liệu; và hướng đi cuối cùng là áp dụng phương pháp transfer learning bằng cách sử dụng mô hình ngôn ngữ lớn **Helsinki-NLP/opus-mt-en-vi** đã được huấn luyện trước, sau đó tinh chỉnh với tập dữ liệu y khoa. Tập dữ liệu huấn luyện bao gồm 500,000 cặp câu song ngữ Anh – Việt trong lĩnh vực y khoa, với nhiều thuật ngữ chuyên ngành đặc thù. Hiệu quả dịch thuật được đánh giá thông qua chỉ số BLEU và kiểm tra trên các câu văn thực tế. Kết quả cho thấy phương pháp tinh chỉnh mô hình pretrained trên tập dữ liệu y khoa mang lại chất lượng dịch tốt hơn so với hai phương pháp huấn luyện trực tiếp mô hình Transformer từ đầu trên cùng dữ liệu, với cả hai trường hợp là Transformer gốc và Transformer đã cải tiến về kiến trúc.

1 Thông tin chung

- Giảng viên hướng dẫn: **TS. Trần Hồng Việt**

- Thành viên nhóm:

Tên	Mã sinh viên	Lớp
Nguyễn Hoàng Tú	23020428	K68-A-AI2
Nguyễn Khánh Tùng	23020434	K68-A-AI2

2 Giới thiệu

2.1 Đặt vấn đề

Trong bối cảnh toàn cầu hóa, dịch máy nơ-ron (Neural Machine Translation - NMT) đã trở thành một công cụ thiết yếu, đóng vai trò cầu nối trong việc phá vỡ rào cản ngôn ngữ, thúc đẩy giao tiếp và trao đổi thông tin trên toàn thế giới. Đặc biệt, trong lĩnh vực y khoa, nơi mà sự chính xác và kịp thời của thông tin có thể ảnh hưởng trực tiếp đến công tác nghiên cứu, chẩn đoán và điều trị, vai trò của dịch máy lại càng trở nên quan trọng. Việc chuyển ngữ nhanh chóng các tài liệu, công trình nghiên cứu, hay tóm tắt y văn giúp các chuyên gia y tế trên toàn cầu tiếp cận với những kiến thức mới nhất, góp phần nâng cao chất lượng chăm sóc sức khỏe.

Tuy nhiên, dịch thuật chuyên ngành y khoa là một trong những bài toán thách thức nhất đối với các hệ thống Machine Translation. Lĩnh vực này có những đặc thù riêng biệt, bao gồm:

- **Thuật ngữ phức tạp**: Hệ thống thuật ngữ y khoa rất phong phú, phần lớn có nguồn gốc Latin - Hy Lạp và đòi hỏi độ chính xác tuyệt đối. Chỉ một sai sót nhỏ trong dịch thuật có thể làm thay đổi hoàn toàn ý nghĩa khoa học của văn bản.
- **Cấu trúc câu dài và phức tạp**: Các văn bản y khoa thường sử dụng câu bị động, nhiều mệnh đề phụ lồng ghép, gây khó khăn cho mô hình trong việc phân tích và nắm bắt chính xác quan hệ ngữ nghĩa.
- **Sự đa nghĩa phụ thuộc ngữ cảnh**: Nhiều thuật ngữ có thể mang ý nghĩa khác nhau tùy vào bối cảnh y khoa cụ thể, đòi hỏi mô hình phải có khả năng suy luận ngữ cảnh sâu.

Dù sở hữu năng lực dịch thuật mạnh mẽ, các hệ thống dịch máy đa dụng vẫn thường gặp hạn chế trước những thách thức đặc thù của y khoa, dẫn đến kết quả dịch thiếu chính xác và không đảm bảo tính tin cậy.

2.2 Mục tiêu

Đứng trước những thách thức từ thực tiễn, chúng em thực hiện đề tài này với các mục tiêu chính như sau :

1. Xây dựng và huấn luyện một hệ thống dịch máy nơ-ron (Neural Machine Translation) với nền tảng chính là kiến trúc Transformer tiên tiến để dịch các văn bản chuyên ngành y khoa từ tiếng Anh sang tiếng Việt.

2. Triển khai, so sánh và đánh giá 3 phương pháp huấn luyện mô hình dịch máy khác nhau cho hệ thống :

- **Phương pháp 1**: Xây dựng mô hình Transformer nguyên gốc từ đầu bằng PyTorch, huấn luyện trực tiếp trên tập dữ liệu chuyên ngành được cung cấp nhằm đánh giá khả năng học của mô hình từ nền tảng cơ bản.
- **Phương pháp 2**: Sử dụng mô hình Re-Transformer là một phiên bản Transformer đã được cải tiến về mặt kiến trúc để huấn luyện trực tiếp từ đầu trên tập dữ liệu y khoa đã được cung cấp.
- **Phương pháp 3**: Tinh chỉnh một mô hình dịch máy đã được huấn luyện trước (**Helsinki-NLP/opus-mt-en-vi**) bằng kỹ thuật QLoRA, nhằm chuyên môn hóa và tối ưu hóa hiệu suất của mô hình trên miền dữ liệu y khoa.

3. Đánh giá hiệu năng dịch thuật của ba mô hình xây dựng từ ba phương pháp trên thông qua cả phương pháp định lượng và định tính : Sử dụng chỉ số BLEU để đánh giá định lượng và đánh giá định tính qua việc phân tích lỗi ngay trên các bản dịch từ mô hình đã huấn luyện.

4. Đưa ra kết luận về những ưu điểm và hạn chế của từng phương pháp.

3 Cơ sở lý thuyết

3.1 Kiến trúc Transformer

Transformer là một kiến trúc neural network được giới thiệu trong bài báo "*Attention is All You Need*"(Vaswani et al., 2017), hiện đóng vai trò nền tảng cho hầu hết các mô hình ngôn ngữ lớn (LLM) cũng như nhiều ứng dụng trong xử lý ngôn ngữ tự nhiên (NLP). Điểm nổi bật của Transformer nằm ở việc thay thế cơ chế xử lý tuần tự của RNN bằng cơ chế attention, cho phép mô hình xử lý toàn bộ chuỗi song song, từ đó tăng tốc quá trình huấn luyện và cải thiện khả năng nắm bắt các quan hệ dài hạn trong câu.

Mô hình Transformer có kiến trúc tinh vi và giàu thành phần hơn so với các kiến trúc Seq2Seq truyền thống dựa trên RNN. Tuy nhiên, hai bộ phận trung tâm cấu thành vẫn là Encoder và Decoder, được kết hợp với một số thành phần quan trọng khác, bao gồm:

1. **Embedding Layer**

- Sau khi các câu văn đã được mã hóa thành chuỗi các token, Embedding Layer có chức năng chuyển đổi các token đầu vào thành các vector liên tục trong không gian nhiều chiều.
- Các token đại diện cho các từ từ vựng được chuyển thành các vector số học để máy tính học được mối quan hệ ngữ nghĩa giữa các từ vựng.

2. **Positional Encoding**

- Do Transformer không xử lý dữ liệu theo thứ tự tuần tự như RNN, mô hình cần cơ chế để nhận biết vị trí của từng token trong chuỗi.
- Positional Encoding là thành phần tạo ra các vector vị trí và cộng trực tiếp vào vector embedding của token, giúp mô hình nắm bắt thông tin thứ tự giữa các từ trong câu nhằm học được khả năng dịch tốt hơn.
- Bài báo gốc sử dụng hàm sin và cos với các tần số khác nhau để đảm bảo mỗi vị trí có một biểu diễn duy nhất, đồng thời giúp mô hình học được quan hệ cả ở khoảng cách gần và xa giữa các token.

3. **Multi-Head Self-Attention**

- Là cơ chế attention được nhân rộng thành nhiều "đầu"(head) độc lập, giúp mô hình học được nhiều loại quan hệ khác nhau giữa các token.
- Trong Encoder, self-attention cho phép mỗi token quan sát toàn bộ chuỗi đầu vào.
- Trong Decoder, self-attention được kết hợp với causal mask để đảm bảo mỗi vị trí chỉ nhìn thấy các token trước đó.

4. **Feed-Forward Network (FFN)**

- Gồm hai lớp tuyến tính (linear) và một hàm kích hoạt phi tuyến (thường dùng ReLU hoặc GELU).
- Áp dụng độc lập cho từng vị trí trong chuỗi, giúp tăng khả năng biểu diễn phi tuyến.

5. **Add & Norm (Residual Connection + Layer Normalization)**

- Kết nối tắt (residual connection) giúp giảm hiện tượng mất mát thông tin và hỗ trợ lan truyền gradient.
- Layer Normalization giúp ổn định và tăng tốc quá trình huấn luyện.

6. **Encoder - Decoder**

- **Encoder**: Bao gồm nhiều lớp giống nhau xếp chồng lên nhau, mỗi lớp bao gồm cơ chế Multi-Head Self-Attention và Feed-Forward Network. Encoder nhận vào chuỗi token nguồn đã được embedding và positional encoding, sau đó tạo ra các biểu diễn (representation) giàu thông tin về ngữ nghĩa và mối quan hệ giữa các token. Nhờ cơ chế self-attention, mỗi token có thể "chú ý" tới tất cả các token khác trong câu, giúp mô hình nắm bắt các quan hệ gần và xa hiệu quả hơn so với các kiến trúc tuần tự như RNN.
- **Decoder**: Cũng gồm nhiều lớp xếp chồng, nhưng ngoài self-attention và feed-forward, nó còn có Encoder - Decoder Attention. Thành phần này cho phép decoder truy xuất thông tin từ đầu ra của encoder, kết hợp với thông tin về các token đã được sinh ra trước đó. Trong quá trình huấn luyện, **decoder** còn sử dụng causal mask để đảm bảo rằng mỗi token chỉ dựa vào các token trước đó, tránh "nhìn trộm" thông tin tương lai. Output cuối cùng của decoder được đưa qua một lớp linear và softmax để dự đoán token tiếp theo trong câu dịch.

7. **Output Layer**: Lớp tuyến tính kết hợp với Softmax để dự đoán xác suất của từng token tiếp theo trong từ vựng.

Bộ tham số được sử dụng trong mô hình Transformer:

- **d_model**: Kích thước embedding của token và đầu ra của mỗi lớp encoder/decoder. Quyết định không gian biểu diễn của mô hình. Giá trị trong bài báo gốc là 512.

- **n_heads**: Số đầu trong cơ chế Multi-Head Attention, giúp mô hình học được nhiều loại quan hệ giữa các token song song. Giá trị gốc là 8.
- **d_ff**: Kích thước ẩn của Feed-Forward Network trong mỗi lớp, thường lớn hơn d_model để tăng khả năng biểu diễn phi tuyến. Giá trị gốc là 2048.
- **num_layers**: Số lớp encoder và decoder xếp chồng lên nhau, ảnh hưởng trực tiếp đến khả năng học biểu diễn phức tạp. Giá trị gốc là 6.
- **dropout_rate**: Tỷ lệ dropout áp dụng trong attention và feed-forward, dùng để giảm overfitting. Giá trị gốc là 0.1.
- **max_position_embeddings**: Số vị trí tối đa mà positional encoding có thể biểu diễn, quyết định chiều dài câu tối đa mà mô hình có thể xử lý. Giá trị gốc là 512.
- **vocab_size**: Kích thước từ vựng của nguồn và đích, ảnh hưởng đến lớp output projection cuối cùng. Ví dụ giá trị có thể khác nhau tùy tokenizer.

3.2 Transformer trong bài toán dịch máy

Kiến trúc Transformer khi được ứng dụng cho dịch máy hoạt động như một hệ thống hoàn chỉnh, tiếp nhận dữ liệu văn bản thô và chuyển đổi nó thành bản dịch thông qua một quy trình gồm nhiều bước. Nền tảng của hệ thống này là khả năng xử lý song song toàn bộ câu và nắm bắt các mối quan hệ phức tạp giữa các từ thông qua cơ chế Self-Attention với các bước như sau:

Bước 1: Chuẩn bị dữ liệu

Trước khi được đưa vào mô hình, câu văn bản thô phải trải qua một số bước biến đổi:

- **Làm sạch văn bản**: Thực hiện các bước chuẩn hóa cơ bản để giảm nhiễu mà không làm mất thông tin quan trọng.
- **Tokenization**: Tách câu thành các token, có thể là từ (word) hoặc bán từ (subword).
- **Xây dựng Từ vựng và Số hóa**: Tạo ra bộ từ vựng từ dữ liệu huấn luyện, trong đó mỗi token duy nhất được gán với một ID. Các token đặc biệt như <bos>, <eos>, <pad>, <unk> cũng được thêm vào.
- **Padding**: Các câu trong được đệm bằng token <pad> để có cùng độ dài.

Bước 2: Xây dựng quá trình dịch trong mô hình

- Sau khi được xử lí, dữ liệu được đưa vào kiến trúc Transformer. Encoder sẽ đọc và xử lý song song toàn bộ câu nguồn, tạo ra một ma trận chứa đựng thông tin ngữ nghĩa của cả câu. Quá trình dịch sau đó diễn ra ở Decoder.
- Khi huấn luyện, Decoder được cung cấp toàn bộ câu đích đúng ngay từ đầu. Nhờ có Masked Self-Attention, tại mỗi vị trí, nó học cách dự đoán từ tiếp theo dựa trên các từ đúng đứng trước đó. Điều này cho phép quá trình học diễn ra song song và hiệu quả.
- Khi suy luận, mô hình giải mã theo cơ chế tự hồi quy: bắt đầu với token <bos>, sử dụng bộ nhớ để dự đoán từ đầu tiên, sau đó đưa từ vừa tạo vào làm đầu vào cho bước tiếp theo. Quá trình này tiếp tục cho đến khi mô hình sinh ra token kết thúc <eos>.

Bước 3: Thực hiện kĩ thuật giải mã cho bước suy luận (Inference)

Do mô hình giải mã theo cơ chế tự hồi quy, nên sau mỗi bước tạo token mới, Decoder sẽ xuất ra phân phối xác suất trên toàn bộ từ vựng cho token tiếp theo. Từ phân phối này, token được chọn theo chiến lược tìm kiếm, chuyển thành ID, rồi detokenize thành dạng văn bản và đưa lại làm đầu vào cho bước kế tiếp. Quá trình lặp lại cho đến khi sinh token <eos>. Hai phương pháp phổ biến thường được dùng là:

- **Greedy search**: Ở mỗi bước giải mã, mô hình đơn giản chọn token có xác suất cao nhất trong phân phối đầu ra. Cách này có ưu điểm là nhanh và dễ triển khai, nhưng do luôn chọn lựa phương án cục bộ tốt nhất nên dễ bỏ lỡ các chuỗi từ có chất lượng cao hơn về tổng thể, dẫn đến câu dịch có thể kém tự nhiên hoặc thiếu chính xác.

- **Beam Search**: Duy trì k giả thuyết tốt nhất tại mỗi bước. Mỗi giả thuyết sẽ được mở rộng bằng các token mới, sau đó chọn ra k kết quả có tổng xác suất cao nhất để tiếp tục. Phương pháp này thường giúp tìm được câu dịch chất lượng hơn, nhưng đổi lại tốc độ chậm và tốn nhiều bộ nhớ hơn.

3.3 Kiến trúc Re-Transformer

Về cơ bản, Re-Transformer là một mô hình được cải tiến từ mô hình Transformer gốc với vài sự tinh chỉnh nhẹ trong kiến trúc, vậy nên hai mô hình này không có quá nhiều điểm khác biệt. Tuy nhiên, mô hình Re-Transformer được cho là có tiềm năng thể hiện kết quả trên bài toán dịch máy tốt hơn so với mô hình Transformer gốc, nhưng lại tính toán và cần ít thời gian huấn luyện hơn.

Điểm khác biệt giữa Re-Transformer và Transformer nguyên bản:

- Mô hình Re-Transformer kết hợp 2 layer Self-Attention liên tiếp rồi mới áp dụng 1 layer Feed Forward, nghĩa là trong khối encoder với 6 encoder layer, thay vì sử dụng 6 Feed-Forward Network như Transformer gốc, Re-Transformer chỉ dùng 3 FFN.
- Mô hình Re-Transformer vẫn sử dụng 6 encoder layer trong khối encoder, nhưng chỉ dùng 2 hoặc 4 decoder layer trong khối decoder, trong khi ở Transformer gốc sử dụng số encoder layer và decoder layer bằng nhau. Điều này giúp tốc độ huấn luyện nhanh hơn mà không làm giảm BLEU score.

3.4 Kỹ thuật PEFT và LoRA

Vấn đề của fine-tuning toàn phần: Khi fine-tune toàn bộ một mô hình lớn, tất cả trọng số đều được cập nhật. Điều này gây ra một số thách thức:

- Yêu cầu VRAM lớn: Cần lưu trữ gradient và trạng thái của optimizer cho hàng trăm triệu đến hàng tỷ tham số, đòi hỏi GPU bộ nhớ rất cao, vượt khả năng phần cứng phổ thông.
- Tốn dung lượng lưu trữ: Mỗi bản fine-tune phải lưu toàn bộ mô hình (hàng trăm MB hoặc vài GB), gây khó khăn khi triển khai nhiều phiên bản cho các tác vụ khác nhau.
- Nguy cơ Catastrophic Forgetting: Cập nhật toàn bộ trọng số trên dữ liệu chuyên ngành có thể làm mô hình mất kiến thức tổng quát đã học từ trước, giảm khả năng khái quát hóa.

Giải pháp: LoRA (Low-Rank Adaptation) là một kỹ thuật thuộc họ PEFT (Parameter-Efficient Fine-Tuning) được thiết kế để giải quyết các vấn đề trên một cách hiệu quả.

- **Ý tưởng**: Quan sát thực nghiệm cho thấy trong quá trình fine-tuning, thay đổi của ma trận trọng số ΔW thường có hạng nội tại thấp (low intrinsic rank). Nói cách khác, một ma trận thay đổi lớn $d \times k$ có thể được xấp xỉ hiệu quả bằng tích của hai ma trận nhỏ hơn nhiều.
- **Cơ chế hoạt động**:
  - Giữ nguyên (freeze) trọng số gốc W của mô hình.
  - Thay vì học trực tiếp $\Delta W \in \mathbb{R}^{d \times k}$, LoRA chỉ huấn luyện hai ma trận $A \in \mathbb{R}^{r \times k}$ và $B \in \mathbb{R}^{d \times r}$.
  - Khi đó ma trận thay đổi được xấp xỉ theo $\Delta W \approx BA$ và đầu ra của lớp sau là $h = W.x + \Delta W.x = W.x + B.A.x$
- Trong quá trình huấn luyện, **chỉ A và B được cập nhật**. Vì r rất nhỏ, số tham số huấn luyện chỉ chiếm khoảng 1% tổng số tham số mô hình.

Lợi ích chính:

- **Tiết kiệm tài nguyên**: Giảm đáng kể yêu cầu về VRAM, cho phép fine-tune các model lớn trên các GPU phổ thông.
- **Huấn luyện nhanh hơn**: Ít tham số cần cập nhật hơn đồng nghĩa với việc quá trình huấn luyện nhanh hơn.
- **Lưu trữ hiệu quả**: Thay vì lưu toàn bộ model, bạn chỉ cần lưu các ma trận LoRA nhỏ (chỉ vài MB), giúp dễ dàng quản lý và chuyển đổi giữa các tác vụ.

- **Tránh Catastrophic Forgetting**: Vì 99% trọng số gốc được giữ nguyên, mô hình không bị mất đi kiến thức nền tảng.

Các siêu tham số quan trọng:

- r (rank): Quyết định "năng lực"của các ma trận LoRA. Tham số r càng cao, số lượng tham số huấn luyện càng nhiều, model có khả năng học các chi tiết phức tạp hơn nhưng cũng tốn nhiều tài nguyên hơn.
- `lora_alpha`: Một hệ số tỉ lệ (scaling factor) cho các trọng số LoRA. Một quy tắc phổ biến là đặt `lora_alpha` $= 2 \times r$.
- `target_modules`: Danh sách các lớp trong mô hình gốc mà bạn muốn áp dụng LoRA (thường là các ma trận chiếu q, v trong các khối attention).

4 Thực nghiệm

4.1 Mô tả bài toán

Bài toán được đặt ra là xây dựng một hệ thống dịch máy có khả năng chuyển ngữ các văn bản chuyên ngành y khoa từ tiếng Anh sang tiếng Việt với độ chính xác cao và tính tự nhiên trong diễn đạt. Để giải quyết bài toán này, nhóm sử dụng bộ dữ liệu song ngữ Anh–Việt chuyên ngành y khoa được cung cấp:

- Tập dữ liệu Y khoa (Medical Corpus): Bao gồm 500,000 cặp câu cho tập train và 3000 cặp câu cho tập test được trích xuất từ các tài liệu y khoa.
- Đặc điểm của bộ dữ liệu này là chứa một lượng lớn thuật ngữ chuyên ngành và các cấu trúc câu phức tạp, là thử thách chính cho các mô hình dịch máy.

4.2 Thí nghiệm

Trong bài toán xây dựng hệ thống dịch máy văn bản y khoa, chúng em thiết lập hai phương pháp triển khai và huấn luyện trên môi trường Kaggle:

Phương pháp 1: Xây dựng mô hình Transformer thủ công từ và huấn luyện trực tiếp từ đầu trên dataset y khoa.

- **Mô hình**: Sử dụng kiến trúc Transformer nguyên gốc từ bài báo “Attention is all you need”, cài đặt từ đầu và huấn luyện trực tiếp trên bộ dữ liệu được lấy từ hai tệp `train.en.txt` và `train.vi.txt` gồm 500,000 cặp câu song ngữ Anh - Việt.

- **Tiền xử lý dữ liệu**: Để đảm bảo chất lượng đầu vào cho mô hình dịch, dữ liệu song ngữ được chuẩn hóa thông qua hai hàm tiền xử lý riêng cho tiếng Anh và tiếng Việt:

(1) Tiếng Anh:

- Chuyển toàn bộ văn bản sang chữ thường để giảm sự đa dạng không cần thiết do chữ hoa/chữ thường.
- Chuẩn hóa khoảng trắng, loại bỏ khoảng trắng dư thừa ở đầu/cuối câu và giữa các từ.
- Tách dấu hai chấm nếu dính liền từ hai phía để đảm bảo tính thống nhất trong phân tách câu.
- Loại bỏ dấu ngoặc kép không cần thiết, nhưng giữ lại dấu nháy đơn bên trong từ để bảo toàn ngữ nghĩa.
- Chèn khoảng trắng giữa chữ số và đơn vị đo lường trong y khoa.
- Thêm khoảng trắng quanh các toán tử so sánh để dễ dàng tách token.
- Tách số và ký hiệu phần trăm để mô hình xử lý chính xác.

(2) Tiếng Việt:

- Chuyển sang chữ thường và chuẩn hóa khoảng trắng tương tự như tiếng Anh.
- Tách dấu hai chấm, loại bỏ dấu câu không cần thiết (bao gồm dấu nháy đơn và kép).

- Chèn khoảng trắng giữa chữ số và đơn vị, chuẩn hóa định dạng số và dấu phẩy.
- Chuẩn hóa dấu gạch ngang, giữ nguyên các ký hiệu y học và định dạng đặc biệt.
- Thêm khoảng trắng quanh các toán tử so sánh và tách số với ký hiệu phần trăm.

- **Tách từ và xây dựng từ điển**: Sau khi thực hiện tiền xử lý dữ liệu đối với các cặp câu Anh - Việt, ta mã hóa các câu này thành chuỗi các token để mô hình Transformer có thể xử lý. Ở bước này, ta sử dụng Byte-Level Byte Pair Encoding (ByteLevel BPE) - một biến thể của BPE giúp xử lý tốt cả các ký tự đặc biệt và khoảng trắng, đồng thời giảm lỗi tách từ đối với tiếng Việt và các thuật ngữ y khoa. Quy trình bao gồm việc huấn luyện 2 tokenizer riêng biệt cho tiếng Anh và tiếng Việt, trong đó:

  - **Tokenizer tiếng Anh** được huấn luyện trên toàn bộ tập huấn luyện tiếng Anh (`train.en.txt`) với kích thước từ vựng là 32000, loại bỏ các token xuất hiện ít hơn 2 lần, và bổ sung bốn token đặc biệt: <s> (bắt đầu câu), <pad> (padding), </s> (kết thúc câu) và <unk> (token không xác định).
  - **Tokenizer tiếng Việt** được huấn luyện độc lập trên tập huấn luyện tiếng Việt (`train.vi.txt`) với các tham số tương tự.

- **Kỹ thuật trong huấn luyện mô hình:** Bên cạnh việc sử dụng kiến trúc Transformer từ bài báo giới thiệu nguyên gốc, chúng em tích hợp thêm một số kỹ thuật bổ trợ nhằm giúp mô hình cải thiện khả năng học và hội tụ tốt hơn. Các thành phần và kỹ thuật đáng chú ý được sử dụng thêm ở đây gồm:

  - **Tạo batch động với Dataset tùy chỉnh và hàm Collate**: Bên cạnh việc sử dụng kiến trúc Transformer, hệ thống áp dụng kỹ thuật dynamic padding với Dataset tùy chỉnh và hàm `collate_fn` nhằm xử lý dữ liệu có độ dài câu không đồng nhất. Cụ thể, lớp TranslationDataset được xây dựng để quản lý và trả về tensor ID token của câu nguồn và câu đích, trong khi `collate_fn` đảm nhiệm việc ghép batch và thực hiện padding động bằng `pad_sequence` với token <pad> tới chiều dài lớn nhất trong batch. Việc tích hợp `collate_fn` vào DataLoader giúp tối ưu hiệu suất tính toán, giảm lãng phí tài nguyên do padding thừa, đồng thời đảm bảo dữ liệu đầu vào phù hợp với cơ chế attention của Transformer.
  - **Mixed Precision Training**: Trong quá trình huấn luyện, hệ thống áp dụng kỹ thuật Mixed Precision Training bằng cách sử dụng `torch.amp.autocast` để thực hiện các phép tính ở độ chính xác hỗn hợp (FP16 và FP32). Cách tiếp cận này giúp giảm dung lượng bộ nhớ GPU và tăng tốc độ huấn luyện nhờ khai thác hiệu quả phần cứng Tensor Cores trên GPU hiện đại. Để đảm bảo tính ổn định số học khi dùng FP16, hệ thống kết hợp Gradient Scaling thông qua `torch.amp.GradScaler`, giúp tránh hiện tượng gradient underflow và tự động điều chỉnh hệ số scale trong quá trình huấn luyện. Kỹ thuật này vừa tối ưu hiệu suất, vừa giữ được độ chính xác của mô hình.
  - **Weight Typing**: Trong quá trình huấn luyện mô hình Transformer, chúng em sử dụng thêm một kỹ thuật là Weight Tying (ràng buộc trọng số) giữa lớp embedding đầu vào của decoder và lớp tuyến tính đầu ra (projection layer). Ở đây, trọng số của lớp chiếu từ vector ẩn sang phân phối xác suất trên từ vựng được gán bằng trọng số của lớp embedding của decoder. Cách làm này giúp giảm số lượng tham số, tăng tính nhất quán giữa không gian embedding và output, đồng thời cải thiện khả năng tổng quát hóa của mô hình, từ đó góp phần nâng cao chất lượng dịch.
  - **Label Smoothing**: Kỹ thuật làm mượt nhãn Label Smoothing được áp dụng với hệ số 0.1 nhằm cải thiện khả năng tổng quát hóa của mô hình và giảm hiện tượng overfitting. Thay vì gán xác suất 1.0 cho nhãn đúng và 0.0 cho các nhãn còn lại, Label Smoothing phân bổ một phần nhỏ xác suất (0.1) cho các nhãn sai, khiến mô hình bớt "quá tự tin"vào một dự đoán duy nhất. Cách tiếp cận này giúp mô hình học được phân phối xác suất mềm hơn, tăng độ ổn định trong quá trình huấn luyện và cải thiện chất lượng dự đoán, đặc biệt trong các bài toán dịch máy với Transformer.
  - **Gradient Clipping**: Để tăng độ ổn định trong quá trình huấn luyện, chúng em áp dụng kỹ thuật Gradient Clipping với ngưỡng 1.0. Kỹ thuật này đảm bảo rằng norm của gradient không vượt quá một giá trị nhất định, từ đó tránh hiện tượng gradient exploding, giúp mô hình học hiệu quả hơn và hội tụ ổn định hơn.

- **Warmup Inverse Square Root Learning Rate Scheduler**: Warmup Inverse Square Root Learning Rate Scheduler là một chiến lược điều chỉnh tốc độ học theo chiến lược được đề xuất trong bài báo "Attention Is All You Need". Trong giai đoạn đầu huấn luyện (warmup), tốc độ học tăng tuyến tính theo số bước cập nhật, giúp mô hình ổn định và tránh hiện tượng gradient thay đổi đột ngột. Sau khi vượt qua số bước warmup được định trước (4000 bước), tốc độ học giảm dần theo quy luật $1/\sqrt{step}$ và được nhân với hệ số chuẩn hóa $1/\sqrt{d\_model}$ để phù hợp với kích thước mô hình. Cơ chế này đặc biệt phù hợp với Transformer, vốn nhạy cảm với tốc độ học ban đầu, và góp phần cải thiện khả năng hội tụ ổn định của mô hình.
- **Optimizer**: Hệ thống sử dụng Adam optimizer với bộ tham số được khuyến nghị trong bài báo "Attention is all you need", với β_1 = 0.9, β_2 = 0.98, $\epsilon = 10^{-9}$. Adam là thuật toán tối ưu thích nghi, kết hợp giữa momentum và RMSProp, giúp điều chỉnh tốc độ học riêng cho từng tham số dựa trên trung bình có trọng số của gradient và bình phương gradient. Việc lựa chọn β_2 = 0.98 thay vì giá trị mặc định 0.999 giúp phản ứng nhanh hơn với các biến động gradient trong các mô hình Seq2Seq, trong khi giá trị ϵ rất nhỏ đảm bảo tính ổn định số học khi chia cho các giá trị gần 0. Cấu hình này giúp mô hình hội tụ nhanh, ổn định và đạt hiệu quả cao hơn so với Adam mặc định.
- **Pre-Layer Normalization**: Trong mô hình Transformer ban đầu, vị trí của Layer Normalization được đặt sau mỗi tầng attention và feed-forward, hay còn gọi là Post-Layer Normalization. Tuy nhiên, các nghiên cứu sau này đã chỉ ra rằng việc đưa LayerNorm lên trước các khối attention và feed-forward - hay Pre-Layer Normalization - giúp ổn định quá trình huấn luyện, đặc biệt trong các mô hình sâu hoặc khi sử dụng batch nhỏ. Do đó, chúng em thí nghiệm huấn luyện Transformer với Pre-Layer Normalization để giúp mô hình hội tụ ổn định hơn.

- **Kỹ thuật Inference**:

- Sau khi huấn luyện, mô hình sử dụng beam search song song theo batch để sinh câu dịch, thay cho phương pháp greedy decoding nhằm cải thiện chất lượng dịch. Trong quá trình này, mỗi câu nguồn được encoder mã hóa thành các biểu diễn (memory), sau đó decoder sinh từng token một cách tuần tự. Để đảm bảo tính causal, causal mask được áp dụng, giúp mỗi token chỉ dựa vào các token đã sinh trước đó, tránh "nhìn trộm" thông tin tương lai.
- Trong quá trình beam search, mô hình duy trì beam size = 4, nghĩa là cùng lúc sinh 4 nhánh giải mã cho mỗi câu. Tại mỗi bước, xác suất log của các token tiếp theo được tính và cộng dồn vào total score, sau đó áp dụng length penalty (α = 0.6) để cân bằng giữa các câu ngắn và dài, tránh ưu tiên quá mức các câu ngắn. Quá trình lặp lại cho tới khi tất cả nhánh đạt token kết thúc <eos> hoặc đạt độ dài tối đa (max_len).
- Để tối ưu tốc độ, kỹ thuật này được vector hóa theo batch, cho phép cùng lúc xử lý nhiều câu (ví dụ batch size = 32), với việc nhân memory và mask theo beam size để tính toán song song trên GPU. Kết quả cuối cùng là chọn ra câu có tổng score tối đa sau khi áp dụng length penalty từ các nhánh beam, đảm bảo chất lượng dịch tối ưu cho mỗi câu trong batch.

Phương pháp 2: Xây dựng mô hình Re-Transformer từ đầu và huấn luyện trực tiếp trên dataset y khoa được cung cấp

- Ở phương pháp 2, các bước làm đều tương tự phương pháp 1, chỉ có 1 điểm khác biệt là ở phương pháp 2, kiến trúc Transformer được tinh chỉnh lại để trở thành Re-Transformer.

Phương pháp 3: Fine-tuning model Helsinki-NLP/opus-mt-en-vi

- Mô hình **Helsinki-NLP/opus-mt-en-vi** là một phần của dự án OPUS-MT, cung cấp các mô hình dịch máy mã nguồn mở cho nhiều cặp ngôn ngữ. Với khoảng 72 triệu tham số, đây là một mô hình nền tảng mạnh mẽ, ứng cử viên lý tưởng cho việc fine-tuning để thích ứng với các lĩnh vực chuyên ngành.

- **Xử lý dữ liệu**: Sử dụng tokenizer SentencePiece có sẵn đi kèm với model, đã được tối ưu cho cặp ngôn ngữ Anh - Việt. 450,000 câu y khoa được sử dụng để fine-tune.

- **Kỹ thuật**: Sử dụng kỹ thuật QLoRA (Quantization + LoRA) trên nền tảng transformers của Hugging Face để fine-tune hiệu quả. Cấu hình LoRA bao gồm rank (r) = 32, `lora_alpha` = 64, và áp dụng LoRA lên tất cả các lớp chiếu trong khối attention.

- **Chiến lược huấn luyện**: Do thời gian huấn luyện vượt quá 12 tiếng, quá trình được thực hiện qua nhiều session Kaggle, với việc huấn luyện được tiếp tục (resume) từ checkpoint cuối cùng của session trước đó. Các tham số được thiết lập trong Seq2SeqTrainingArguments bao gồm:

  - Learning rate = 10^{-4}
  - Weight_decay = 0.01
  - label_smoothing_factor = 0.1
  - Sử dụng thuật toán Beam Search với generation_num_beams = 5 để cải thiện chất lượng dịch khi đánh giá.

5 Kết quả & Thảo luận

5.1 Đánh giá định lượng

Hiệu năng của các mô hình được đo lường trên tập test gồm 3,000 câu y khoa. Chỉ số BLEU được sử dụng làm thước đo chính. Kết quả được tổng hợp trong bảng dưới đây:

Bảng 1: Kết quả đánh giá chỉ số BLEU trên tập test của từng mô hình

Model	BLEU score
Transformer from scratch (> 50 epoch trained)	57.44
Re-Transformer from scratch (40 epoch trained)	57.39
OpusMT (base)	23.87
OpusMT (finetuned)	60.56

5.2 Đánh giá định tính

Phương pháp 1:

Với hướng đi là huấn luyện Transformer từ đầu trên tập dữ liệu mới như tập dữ liệu y khoa, mô hình đã đạt được BLEU score là 57.44 sau khi được huấn luyện trong khoảng hơn 50 epoch, cùng với những bản dịch tương đối sát với thực tế. Tuy nhiên, mô hình vấp phải một số không ít nhược điểm như thời gian huấn luyện dài, khả năng dịch thuật vẫn còn một số vấn đề như một số câu dịch sai khá nhiều so với câu tiếng Việt gốc, chưa thể xử lý tốt với một số kiểu diễn đạt khó hoặc từ mới gặp lần đầu.

Xét một trường hợp mô hình **dịch đúng**:

- Câu gốc (tiếng Anh): "a cross-sectional descriptive study was implemented to assess fluor contaminated teeth among children aged 12 years of thai ethnic group in con cuong district, nghe an province."

- Câu dịch: "nghiên cứu mô tả cắt ngang nhằm đánh giá tình trạng răng nhiễm fluor ở trẻ 12 tuổi dân tộc thái tại huyện con cuông, tỉnh nghệ an."

- Câu gốc (tiếng Việt): "nghiên cứu mô tả cắt ngang nhằm đánh giá tình trạng răng nhiễm fluor ở trẻ 12 tuổi dân tộc thái tại huyện con cuông, tỉnh nghệ an."

Xét một trường hợp mô hình **dịch sai**:

- Câu gốc (tiếng Anh): "group 1: non-obese mice."

- Câu dịch: "nhóm 1: chuột nhắt trắng không béo phì."

- Câu gốc (tiếng Việt): "lô 1: chuột không gây béo phì."

Phương pháp 2: Với mô hình Re-Transformer, chất lượng dịch của mô hình đã tương đối tốt. Về cơ bản, mô hình đã chứng kiến nhiều cải thiện đáng kể trong cách diễn đạt khi dịch câu.

Xét một trường hợp mô hình **dịch đúng các chuỗi từ dài và phức tạp**:

- Câu gốc (tiếng Anh): "subjects and methods: cross-sectional descriptive study on 4532 outpatient prescriptions of patients who were prescribed drugs by doctors from july 1 st, 2020 to july 30 th, 2020 at clinics of the examination department - military hospital 120."

- Câu dịch: "đối tượng và phương pháp: nghiên cứu mô tả cắt ngang trên 4532 đơn thuốc ngoại trú của bệnh nhân được bác sĩ kê đơn thuốc từ ngày 01/7/2020 đến ngày 30/7/2020 tại phòng khám của khoa khám bệnh - bệnh viện quân y 120."

- Câu gốc (tiếng Việt): "đối tượng và phương pháp: nghiên cứu mô tả cắt ngang trên 4532 đơn thuốc khám bệnh ngoại trú của bệnh nhân được bác sĩ kê đơn thuốc từ ngày 01/7/2020 đến ngày 30/7/2020 tại các phòng khám thuộc khoa khám bệnh - bệnh viện quân y 120."

Xét một ví dụ model **dịch sai ý nghĩa câu**:

- Câu gốc (tiếng Anh): "knowledge, practices in public health service utilization among health insurance card's holders and influencing factors in vientiane, lao."

- Câu dịch: "kiến thức, thực hành sử dụng thẻ bảo hiểm y tế của người dân và một số yếu tố liên quan tại huyện viêng chăn, lào."

- Câu gốc (tiếng Việt): "thực trạng kiến thức và thực hành của người có thẻ bảo hiểm y tế trong sử dụng dịch vụ khám chữa bệnh ở các cơ sở y tế công và một số yếu tố ảnh hưởng tại tỉnh viêng chăn, chdcnd lào, năm 2017."

Nhận xét: model vẫn còn gặp khó khăn để hiểu được một số cụm từ thông dụng lẫn từ chuyên ngành. Tuy nhiều bản dịch cho chất lượng ổn, song vẫn còn nhiều bản dịch vẫn bị sai, khác xa ý nghĩa so với câu gốc, điều đó cho thấy mô hình vẫn chưa nắm bắt quá tốt toàn bộ ngữ cảnh và văn phong trong câu tiếng Việt, vẫn gặp phải thực trạng dịch nhiều theo word-by-word và chưa ứng phó tốt với một số từ chưa gặp.

Phương pháp 3: Với mô hình baseline (Opus-MT chưa fine-tune), chất lượng dịch của mô hình gốc rất thấp. Nó mắc phải nhiều lỗi nghiêm trọng như dịch sai hoàn toàn thuật ngữ, tạo ra các từ vô nghĩa ("Phesente"), và dịch word-by-word một cách máy móc, dẫn đến các câu văn lủng củng, khó hiểu.

Với mô hình sau khi fine-tune đã được cải thiện đáng kể, nhưng vẫn còn những đặc điểm quan trọng về cách mô hình học và các hạn chế còn tồn tại, dựa trên kết quả thực tế từ tập kiểm tra.

Xét một trường hợp mô hình **dịch đúng các chuỗi từ dài và phức tạp**:

- Câu gốc (tiếng Anh): "To investigate the reasonableness of using proton pump inhibitors (PPI) at 120 Military Hospital in Tien Giang province."

- Câu dịch: "Nghiên cứu tính hợp lý của việc sử dụng chất ức chế bơm proton (PPI) tại Bệnh viện Quân đội 120 tỉnh Tiền Giang."

- Câu gốc (tiếng Việt): "Khảo sát tính hợp lý trong sử dụng thuốc ức chế bơm proton (PPI) tại Bệnh viện Quân y 120 tỉnh Tiền Giang."

Điều này cho thấy mô hình đã học được cách **ghi nhớ** các cụm từ chuyên ngành quan trọng và thường xuyên lặp lại. Đây là một khả năng hữu ích, giúp đảm bảo tính nhất quán và chính xác cho các thuật ngữ phổ biến.

Xét một ví dụ model **dịch sai cụm từ chuyên ngành**:

- Câu gốc (tiếng Anh): "Research for clinical characteristics and plasma dopamine concentrations in schizophrenic patients"

- Câu dịch: "Nghiên cứu đặc điểm lâm sàng và nồng độ nồng độ plasma ở bệnh nhân tâm thần phân liệt"

- Câu gốc (tiếng Việt): "Nghiên cứu đặc điểm lâm sàng và nồng độ dopamin huyết tương ở bệnh nhân tâm thần phân liệt."

Và một ví dụ model **dịch nguyên văn theo câu gốc**:

- Câu gốc (tiếng Anh): "The Dean's index of community fluor contaminated teeth in 12-year-old children was 0.13. This was a community without fluor contaminated teeth according to the 2013 classification of the World Health Organization."

- Câu dịch: "Chỉ số răng bị nhiễm khuẩn cộng đồng của Dean ở trẻ 12 tuổi là 0,13 tuổi. Đây là một cộng đồng không nhiễm khuẩn răng theo phân loại của Tổ chức Y tế Thế giới năm 2013."

- Câu tiếng Việt: "Chỉ số răng nhiễm fluor trong cộng đồng theo chỉ số Dean ở trẻ 12 tuổi là 0,13 theo phân loại của Tổ chức Y tế Thế giới năm 2013 thuộc cộng đồng không bị nhiễm fluor răng."

Nhận xét: model gặp khó khăn với các cụm từ về y khoa chưa gặp bao giờ. Tuy không có trường hợp bị chệch hướng dẫn đến dịch sai hoàn toàn câu văn, model vẫn thi thoảng dịch sai một số cụm từ dễ và chưa nắm bắt được toàn bộ văn phong của câu tiếng Việt mà thường dịch word-by-word.

5.3 Nhận xét

Nhận xét chung về kết quả:

- **Hiệu quả của Fine-tuning**: Kết quả thực nghiệm khẳng định rằng fine-tuning một mô hình lớn đã được huấn luyện trước (Phương pháp 3) là chiến lược hiệu quả nhất. Nó tận dụng được kiến thức ngôn ngữ sâu rộng của mô hình gốc và chỉ cần một lượng tài nguyên tương đối nhỏ để chuyên môn hóa cho lĩnh vực mới.
- **Tầm quan trọng của Pre-training**: So sánh giữa các phương pháp 1 và 2 với phương pháp 3 cho thấy tầm quan trọng của giai đoạn pre-training đối với các mô hình xây dựng từ đầu. Việc học trên một bộ dữ liệu lớn và đa dạng trước giúp model xây dựng một nền tảng ngôn ngữ vững chắc, là tiền đề quan trọng để học tốt hơn trên dữ liệu chuyên ngành.
- **Hạn chế của BLEU**: Mặc dù điểm BLEU tăng tương quan với chất lượng dịch, nó không phản ánh được toàn bộ câu chuyện. Các lỗi sai nghiêm trọng về mặt y khoa đôi khi chỉ ảnh hưởng một chút đến điểm BLEU. Do đó, việc phân tích định tính thủ công là không thể thiếu để đánh giá đúng mức độ tin cậy của một hệ thống dịch máy chuyên ngành.

Đề xuất giải pháp:

- **Lỗi quan sát được**: Mô hình vẫn dịch sai hoặc không dịch được các thuật ngữ hóa dược rất cụ thể và hiếm gặp.
- **Giải pháp đề xuất**: Một trong những cách hiệu quả nhất để khắc phục triệt để vấn đề này là Bảng thuật ngữ (Glossary). Đây là một danh sách các cặp thuật ngữ Anh-Việt đã được định nghĩa trước. Trong quá trình dịch, hệ thống có thể được lập trình để ưu tiên hoặc bắt buộc phải sử dụng bản dịch từ bảng thuật ngữ mỗi khi nó gặp một từ khóa có trong danh sách. Kỹ thuật này đảm bảo các thuật ngữ quan trọng, không thể dịch sai, bù đắp cho những "lỗ hổng" kiến thức của mô hình.

5.4 Kết luận và hướng phát triển

Kết luận: Đề tài đã xây dựng và so sánh thành công ba phương pháp dịch máy Anh-Việt cho chuyên ngành y khoa. Kết quả cho thấy phương pháp fine-tuning một mô hình lớn có sẵn bằng kỹ thuật QLoRA (Phương pháp số 3) mang lại hiệu quả vượt trội nhất, tạo ra các bản dịch vừa chính xác về mặt thuật ngữ, vừa tự nhiên về mặt văn phong.

Hướng phát triển: Trong tương lai, nhóm có thể tiếp tục phát triển bài toán theo các hướng sau đây:

- **Sử dụng model nền tảng lớn hơn**: Thử nghiệm fine-tune các model lớn hơn như NLLB-200-1.3B có thể mang lại chất lượng dịch tốt hơn nữa.
- **Tăng cường dữ liệu bằng Back-Translation**: Sử dụng một model dịch Việt-Anh đủ tốt để tạo ra thêm dữ liệu huấn luyện "giả lập"từ các văn bản y khoa tiếng Việt gốc.
- **Tích hợp Bảng thuật ngữ (Glossary)**: Xây dựng bảng thuật ngữ bắt model phải dịch đúng các thuật ngữ y khoa quan trọng đã được định nghĩa trước.

6 Phụ lục

6.1 Mã nguồn

- **Github**: https://github.com/darkjeanne/nlp_asm_group1/tree/main

6.2 Tài liệu tham khảo

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need.

2. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs.

3. Tiedemann, J., & Thottingal, S. (2020). OPUS-MT – Building open translation services for the World.

4. Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation.

5. Huey-Ing Liu & Wei-Lin Chen (2021). Re-Transformer: A Self-Attention Based Model for Machine Translation