

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



**NGUYỄN MINH HOÀNG CHUƠNG - 52000744  
TRẦN VĂN DUY - 52000751**

**ĐÁNH GIÁ ĐỘ HIỆU QUẢ GIỮA MÔ  
HÌNH DỊCH MÁY ĐA NGỮ VÀ  
DỊCH MÁY SONG NGỮ TRÊN CẤP  
NGÔN NGỮ HẠN CHẾ TÀI NGUYÊN**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN**

**KHOA HỌC MÁY TÍNH**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



**NGUYỄN MINH HOÀNG CHƯƠNG - 52000744  
TRẦN VĂN DUY - 52000751**

**ĐÁNH GIÁ HIỆU QUẢ GIỮA MÔ HÌNH  
DỊCH MÁY ĐA NGỮ VÀ DỊCH MÁY  
SONG NGỮ TRÊN CẤP NGÔN NGỮ  
HẠN CHẾ TÀI NGUYÊN**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN**

**KHOA HỌC MÁY TÍNH**

Người hướng dẫn  
**ThS. Dung Cẩm Quang**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

## LỜI CẢM ƠN

Chúng tôi xin chân thành cảm ơn và biết ơn thời gian mà thầy ThS. Dung Cẩm Quang đã dành cho Dự án Công nghệ thông tin của chúng tôi. Với sự chỉ dẫn của thầy đã giúp nhóm có thêm nhiều kiến thức, tài liệu tham khảo và hướng giải quyết bài toán dễ dàng hơn. Nhờ có sự góp ý của thầy trong quá trình làm Dự án, nhóm chúng tôi có cơ hội học hỏi và khám phá thêm nhiều kiến thức mới. Chúc thầy luôn dồi dào sức khỏe và thành công trong công việc.

*TP. Hồ Chí Minh, ngày ... tháng ... năm 20..*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

## **CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Chúng tôi xin cam đoan đây là công trình nghiên cứu của riêng chúng tôi và được sự hướng dẫn khoa học của ThS. Dung Cẩm Quang. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình.** Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày ... tháng ... năm 20..*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

# **ĐÁNH GIÁ ĐỘ HIỆU QUẢ GIỮA MÔ HÌNH DỊCH MÁY ĐA NGỮ VÀ DỊCH MÁY SONG NGỮ TRÊN CẶP NGÔN NGỮ HẠN CHẾ TÀI NGUYÊN**

## **TÓM TẮT**

Trong dự án này, chúng tôi đã tiến hành đánh giá và so sánh hiệu quả giữa mô hình dịch máy đa ngữ (bao gồm tiếng Việt, tiếng Anh và tiếng Tây Ban Nha) và mô hình dịch máy song ngữ (tiếng Việt, tiếng Tây Ban Nha), trong đó cặp ngôn ngữ Việt - Tây Ban Nha có nguồn tài nguyên hạn chế nhưng lại có dữ liệu dồi dào với ngôn ngữ trung gian là tiếng Anh.

Cả hai mô hình dịch đều sử dụng mô hình mT5, một kiến trúc dựa trên mô hình Transformer, cho phép đánh giá chính xác hiệu suất và khả năng của mỗi phương pháp. Kết quả của nghiên cứu cho thấy cải thiện nhỏ trong điểm BLEU khi sử dụng mô hình đa ngữ so với mô hình song ngữ. Điểm BLEU đạt được cho mô hình đa ngữ từ tiếng Tây Ban Nha sang tiếng Việt là 20.96 so với điểm BLEU là 20.87 của mô hình song ngữ. Và điểm BLEU đạt được cho mô hình đa ngữ từ tiếng Việt sang tiếng Tây Ban Nha là 15.83 so với mô hình song ngữ là 14.09. Điều này cho thấy rằng bổ sung một ngôn ngữ trung gian với tài nguyên dồi dào có thể giúp cải thiện chất lượng dịch cho các cặp ngôn ngữ hạn chế tài nguyên.

Kết quả trên nhờ sự hỗ trợ của ngôn ngữ trung gian tài nguyên dồi dào, giúp tăng cường hiệu quả dịch máy cho các cặp ngôn ngữ hạn chế tài nguyên. Điều này không chỉ mở ra hướng mới cho việc nghiên cứu và phát triển mô hình dịch máy mà còn cho thấy tiềm năng của việc sử dụng các ngôn ngữ giàu tài nguyên để hỗ trợ dịch cho các ngôn ngữ hạn chế tài nguyên.

# **EVALUATING THE EFFECTIVENESS BETWEEN MULTILINGUAL AND BILINGUAL MACHINE TRANSLATION MODEL ON LOW-RESOURCE CORPUS**

## **ABSTRACT**

In this project, we conducted evaluations and comparisons of the effectiveness between multilingual translation models (including Vietnamese, English, and Spanish) and bilingual translation models (Vietnamese, Spanish), where the Vietnamese - Spanish language pair has low resources but abundant data with English as an intermediary language.

Both translation models used the mT5 model, a powerful Transformer architecture, allowing for accurate assessment of the performance and capabilities of each method. The study's results showed a slight improvement in BLEU scores when using the multilingual model compared to the bilingual model. The BLEU score achieved for the multilingual model from Spanish to Vietnamese was 20.96, compared to the bilingual model's BLEU score of 20.87. And the BLEU score achieved for the multilingual model from Vietnamese to Spanish was 15.83, compared to the bilingual model at 14.09. This indicates that supplementing with a resource-rich intermediary language can help improve translation quality for language pairs with low-resources.

These results, thanks to the support of a resource-rich intermediary language, help enhance the translation efficiency for language pairs with low-resources. This not only opens new directions for the research and development of translation models but also shows the potential of using resource-rich languages to support translation for languages with low-resources.

## MỤC LỤC

<b>DANH MỤC HÌNH VẼ .....</b>	<b>vii</b>
<b>DANH MỤC BẢNG BIỂU .....</b>	<b>viii</b>
<b>DANH MỤC CÁC CHỮ VIẾT TẮT.....</b>	<b>ix</b>
<b>CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI.....</b>	<b>1</b>
1.1 Giới thiệu đề tài.....	1
1.2 Mục tiêu đề tài.....	1
1.3 Phương pháp nghiên cứu.....	2
1.4 Ý nghĩa khoa học .....	2
1.5 Ý nghĩa thực tiễn .....	3
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....</b>	<b>4</b>
2.1 Khảo sát bài báo khoa học cùng lĩnh vực .....	4
2.2 Mô hình Transformer .....	6
2.2.1 Giới thiệu Transformer .....	6
2.2.2 Kiến trúc mô hình.....	6
2.2.3 Cơ chế Attention.....	9
2.2.4 Position-wise Feed-Forward Networks .....	13
2.2.5 Embeddings and Softmax và Positional Encoding .....	14
2.3 Mô hình T5 và Mô hình mT5.....	15
2.3.1 Mô hình T5.....	15
2.3.2 Mô hình mT5.....	17
2.4 Chỉ số đánh giá mô hình tự động BLEU.....	18
2.5 Lý thuyết về các kỹ thuật xử lý dữ liệu.....	22

2.5.1 Data Cleaning .....	22
2.5.2 Tokenization .....	23
2.5.3 Language Tags .....	25
2.5.4 Encoding with padding .....	26
2.5.5 Optimizer .....	27
<b>CHƯƠNG 3. THỰC NGHIỆM .....</b>	<b>29</b>
3.1 Giới thiệu tập dữ liệu .....	29
3.1.1 Tập dữ liệu tiếng Việt – tiếng Tây Ban Nha .....	29
3.1.2 Tập dữ liệu tiếng Anh – tiếng Việt .....	31
3.1.3 Tập dữ liệu tiếng Anh – tiếng Tây Ban Nha .....	33
3.2 Xử lý dữ liệu và cài đặt chạy thực nghiệm .....	35
3.2.1 Xử lý dữ liệu .....	35
3.2.2 Cài đặt và chạy thực nghiệm .....	35
3.3 Phân tích kết quả .....	37
3.4 Demo website dịch máy đa ngôn ngữ .....	39
<b>CHƯƠNG 4. KẾT LUẬN .....</b>	<b>43</b>
4.1 Kết quả đạt được .....	43
4.2 Hạn chế của phương pháp giải quyết bài toán .....	43
4.3 Hướng phát triển trong tương lai .....	43
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>44</b>



## DANH MỤC HÌNH VẼ

Hình 2.1 Hình The Transformer - model architecture (Vaswani et al. 2017).....	8
Hình 2.2 Scaled Dot-Product Attention & Multi-Head Attention (Vaswani et al. 2017).....	10
Hình 2.3 A diagram of our text-to-text framework (Raffel et al. 2023) .....	15
Hình 3.1 Hình ảnh dữ liệu tập train tiếng Việt và tiếng Tây Ban Nha .....	29
Hình 3.2 Hình ảnh dữ liệu tập test tiếng Việt và tiếng Tây Ban Nha .....	30
Hình 3.3 Hình ảnh dữ liệu tập validation tiếng Việt và tiếng Tây Ban Nha.....	30
Hình 3.4 Hình ảnh dữ liệu tập train tiếng Anh và tiếng Việt.....	31
Hình 3.5 Hình ảnh dữ liệu tập test tiếng Anh và tiếng Việt.....	32
Hình 3.6 Hình ảnh dữ liệu tập validation tiếng Anh và tiếng Việt .....	32
Hình 3.7 Hình ảnh dữ liệu tập train tiếng Anh và tiếng Tây Ban Nha .....	33
Hình 3.8 Hình ảnh dữ liệu tập test tiếng Anh và tiếng Tây Ban Nha .....	34
Hình 3.9 Hình ảnh dữ liệu tập validation tiếng Anh và tiếng Tây Ban Nha.....	34
Hình 3.10 So sánh BLEU Score giữa mô hình đa ngữ và mô hình song ngữ .....	38
Hình 3.11 Giao diện trang chủ .....	39
Hình 3.12 Hình ảnh demo hướng dẫn dịch từ tiếng Anh sang tiếng Việt.....	39
Hình 3.13 Hình ảnh demo hướng dẫn dịch từ tiếng Anh sang tiếng Tây Ban Nha ..	40
Hình 3.14 Hình ảnh demo hướng dẫn dịch từ tiếng Việt sang tiếng Tây Ban Nha ..	40
Hình 3.15 Hình ảnh demo hướng dẫn dịch từ tiếng Việt sang tiếng Anh.....	41
Hình 3.16 Hình ảnh demo hướng dẫn dịch từ tiếng Tây Ban Nha sang tiếng Việt ..	41
Hình 3.17 Hình ảnh demo hướng dẫn dịch từ tiếng Tây Ban Nha sang tiếng Anh ..	42
Hình 3.18 Hình ảnh demo trường hợp input trùng với output .....	42

## **DANH MỤC BẢNG BIỂU**

Bảng 3.1 Bảng thống kê dữ liệu Việt – Tây Ban Nha .....	29
Bảng 3.2 Bảng thống kê dữ liệu tiếng Anh – tiếng Việt .....	31
Bảng 3.3 Bảng thống kê dữ liệu tiếng Anh – tiếng Tây Ban Nha .....	33
Bảng 3.4 Các tham số huấn luyện của mô hình song ngữ mT5.....	36
Bảng 3.5 Các tham số huấn luyện của mô hình đa ngữ mT5 .....	37
Bảng 3.6 Kết quả mô hình song ngữ.....	37
Bảng 3.7 Kết quả mô hình đa ngữ.....	37

## **DANH MỤC CÁC CHỮ VIẾT TẮT**

BLEU	Bilingual Evaluation Understudy
LRL	Low-Resource Language
NLP	Natural Language Processing
NMT	Multilingual neural machine translation

# CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

## 1.1 Giới thiệu đề tài

Trong bối cảnh toàn cầu hóa ngày nay, nhu cầu giao tiếp và trao đổi thông tin qua các ngôn ngữ khác nhau ngày càng trở nên quan trọng. Bài toán dịch máy dựa trên mạng neural (Bentivogli et al. 2016) đã mở ra một kỷ nguyên mới trong việc giải quyết nhu cầu này, với khả năng cung cấp dịch thuật tự động chính xác và linh hoạt hơn so với các phương pháp truyền thống. Tuy nhiên, dịch máy hiệu quả giữa các ngôn ngữ không phải lúc nào cũng dễ dàng, đặc biệt là giữa các ngôn ngữ có ít dữ liệu song ngữ, hay còn gọi là các ngôn ngữ hạn chế tài nguyên.

Trọng tâm của đề tài này là khám phá và phát triển phương pháp trong việc dịch máy giữa tiếng Việt và tiếng Tây Ban Nha, hai ngôn ngữ này có giới hạn về lượng dữ liệu song ngữ có sẵn. Đối mặt với thách thức này, chúng tôi đề xuất việc áp dụng tiếng Anh làm ngôn ngữ trung gian để tận dụng kho dữ liệu dồi dào giữa tiếng Anh đối với hai ngôn ngữ trên. Sử dụng mô hình dịch máy neural mT5 mà không phụ thuộc vào trọng số đã được huấn luyện trước, cho phép tinh chỉnh mô hình một cách hiệu quả dựa trên bộ dữ liệu đa ngữ, hướng đến việc cải thiện đáng kể chất lượng dịch cho cặp ngôn ngữ Việt - Tây Ban Nha.

## 1.2 Mục tiêu đề tài

Đề tài này tập trung vào việc phát triển mô hình dịch máy song ngữ giữa tiếng Việt và tiếng Tây Ban Nha dựa trên bộ dữ liệu hạn chế. Đồng thời, nghiên cứu cũng tiến hành huấn luyện một mô hình đa ngữ, kết hợp tiếng Việt, tiếng Tây Ban Nha và tiếng Anh, để khám phá liệu việc sử dụng một ngôn ngữ trung gian với tài nguyên dồi dào có thể cải thiện chất lượng dịch cho hai ngôn ngữ còn lại hay không. Cả hai mô hình dịch máy trên đều được huấn luyện bằng mô hình mT5 (Britz et al. 2017) không sử dụng trọng số đã được huấn luyện trước. Cuối cùng, nghiên cứu so sánh và đánh giá hiệu quả của mô hình song ngữ so với mô hình đa ngữ, nhằm xác định liệu sự hỗ trợ từ một ngôn ngữ trung gian có thể tạo ra sự cải thiện đáng kể trong chất lượng dịch đối với các ngôn ngữ có hạn chế về tài nguyên.

### 1.3 Phương pháp nghiên cứu

Chúng tôi tiến hành thu thập bộ dữ liệu song ngữ giữa tiếng Việt và tiếng Tây Ban Nha, cũng như dữ liệu song ngữ giữa tiếng Anh với tiếng Việt và tiếng Anh với Tiếng Tây Ban Nha. Dữ liệu này sau đó được tiền xử lý, bao gồm làm sạch và chuẩn hóa, để tối ưu hóa cho quá trình huấn luyện.

Mô hình mT5, không sử dụng trọng số đã được huấn luyện trước, được tinh chỉnh riêng lẻ cho mỗi bộ dữ liệu. Đối với phần dịch song ngữ, mô hình được huấn luyện trực tiếp trên dữ liệu giữa tiếng Việt và tiếng Tây Ban Nha. Trong khi đó, mô hình đa ngữ được huấn luyện để xử lý dữ liệu từ cả ba ngôn ngữ, sử dụng tiếng Anh như một cầu nối để cải thiện chất lượng dịch giữa hai ngôn ngữ còn lại.

Cuối cùng, chúng tôi đánh giá và so sánh hiệu suất giữa mô hình song ngữ và đa ngữ thông qua các chỉ số đánh giá chất lượng dịch chuẩn như BLEU (Bi-Lingual Evaluation Understudy) (Papineni, Roukos, Ward, Zhu, et al. 2002). Phân tích kết quả giúp xác định liệu việc sử dụng ngôn ngữ trung gian có thể cải thiện đáng kể chất lượng dịch trong trường hợp ngôn ngữ hạn chế tài nguyên.

### 1.4 Ý nghĩa khoa học

Đề tài nghiên cứu này đem lại ý nghĩa khoa học rất lớn, bởi vì nó mở rộng hiểu biết về khả năng áp dụng NMT trong bối cảnh của các ngôn ngữ hạn chế tài nguyên như tiếng Việt và tiếng Tây Ban Nha. Nghiên cứu cung cấp cái nhìn sâu sắc về việc sử dụng mô hình mT5 không tải trọng số đã được huấn luyện trước, đề xuất một cách tiếp cận mới cho việc cải thiện chất lượng dịch thông qua sử dụng ngôn ngữ trung gian là tiếng Anh. Điều này không chỉ có ý nghĩa trong việc nâng cao chất lượng dịch giữa các ngôn ngữ ít dữ liệu, mà còn góp phần vào lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) bằng cách khám phá tiềm năng của các mô hình đa ngữ. Kết quả của nghiên cứu có thể mở đường cho các phương pháp mới trong việc xử lý và dịch các ngôn ngữ ít được nghiên cứu, đồng thời cung cấp cái nhìn mới mẻ về việc tận dụng nguồn lực của ngôn ngữ trung gian trong công nghệ dịch máy, qua đó đóng góp vào việc xóa bỏ rào cản ngôn ngữ trên toàn cầu. Phát hiện này khẳng định tầm quan trọng của

việc áp dụng công nghệ AI và machine learning trong lĩnh vực dịch thuật, đặc biệt là trong việc xử lý các ngôn ngữ có ít dữ liệu. Hơn nữa, nó mở ra hướng nghiên cứu mới về khả năng tối ưu hóa quy trình dịch bằng cách kết hợp ngôn ngữ trung gian, qua đó cải thiện độ chính xác và tự nhiên của bản dịch. Cuối cùng, nghiên cứu này cũng đặt nền móng cho việc phát triển các mô hình dịch máy hiệu quả hơn, góp phần vào sự phát triển của cộng đồng NLP toàn cầu.

### **1.5 Ý nghĩa thực tiễn**

Ý nghĩa thực tiễn của đề tài nghiên cứu này đề xuất giải pháp tiên tiến cho việc cải thiện giao tiếp và hiểu biết lẫn nhau giữa những người sử dụng tiếng Việt và tiếng Tây Ban Nha. Qua việc áp dụng mô hình mT5 không dựa trên trọng số đã huấn luyện trước và sử dụng tiếng Anh làm ngôn ngữ trung gian, nghiên cứu không chỉ mở ra cơ hội để cải thiện độ chính xác của dịch máy giữa các ngôn ngữ có hạn chế về dữ liệu, mà còn hỗ trợ cộng đồng người dùng trong việc tiếp cận thông tin và tài nguyên trên toàn cầu một cách dễ dàng hơn. Điều này đặc biệt quan trọng trong thời đại số hóa, nơi mà việc trao đổi thông tin nhanh chóng và chính xác giữa các ngôn ngữ và văn hóa khác nhau trở nên thiết yếu. Ngoài ra, việc phát triển mô hình dịch máy đa ngữ còn giúp tăng cường khả năng hiểu biết và hợp tác quốc tế, thúc đẩy giao lưu văn hóa và kinh tế. Kết quả của nghiên cứu có thể được ứng dụng trong nhiều lĩnh vực, từ giáo dục, du lịch đến thương mại và ngoại giao, qua đó đóng góp vào việc xây dựng một cộng đồng toàn cầu gắn kết hơn. Điều này không chỉ làm phong phú thêm kho tàng văn hóa toàn cầu, mà còn mở ra cánh cửa cho những phát triển mới trong công nghệ thông tin và truyền thông, tạo điều kiện cho việc phát triển bền vững và tích cực trên phạm vi toàn cầu. Đi sâu vào ứng dụng thực tiễn, nghiên cứu này còn làm sáng tỏ cách công nghệ mới có thể giảm bớt những rào cản ngôn ngữ, qua đó khuyến khích sự hiểu biết và tôn trọng lẫn nhau giữa các nền văn hóa. Đặc biệt, trong bối cảnh toàn cầu hóa, khả năng truyền đạt và hiểu thông tin một cách chính xác và nhanh chóng giữa các ngôn ngữ khác nhau trở nên cực kỳ quan trọng.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1 Khảo sát bài báo khoa học cùng lĩnh vực

#### Giới thiệu về bài báo

Tiêu đề bài báo: “Low-resource neural machine translation: A benchmark for five african languages”. (Lakew, Negri, and Turchi 2020)

Tác giả: Surafel M. Lakew, Matteo Negri, và Marco Turchi.

Mục tiêu chính và phạm vi nghiên cứu của bài báo:

Trong nghiên cứu này, các tác giả có mục đích chính là đánh giá hiệu suất của các mô hình NMT giữa tiếng Anh và năm ngôn ngữ ít tài nguyên (Low-Resource Languages - LRL) của châu Phi, bao gồm Swahili, Amharic, Tigrigna, Oromo và Somali (SATOS). Các mô hình NMT được đánh giá không chỉ trong bối cảnh dịch từ tiếng Anh sang các ngôn ngữ SATOS ( $En \rightarrow LRL$ ) mà còn từ các ngôn ngữ SATOS sang tiếng Anh ( $LRL \rightarrow En$ ). Các phương pháp đánh giá bao gồm mô hình NMT cơ bản (Standard neural machine translation) (Bahdanau 2016), mô hình bán giám sát (Semi-Supervised neural machine translation) (Sennrich, Haddow, and Birch 2016), học chuyển giao (Transfer learning) (Zhuang et al. 2020), và mô hình đa ngôn ngữ (Multilingual neural machine translation) (Freitag and Firat 2020). Những phương pháp này được so sánh dựa trên hiệu suất dịch ngôn ngữ, được đo lường bằng điểm số BLEU, để xác định cách tiếp cận nào cho kết quả tốt nhất.

#### Dữ liệu dùng để nghiên cứu trong bài báo

- Tiếp cận dữ liệu:

Nghiên cứu này sử dụng các nguồn dữ liệu đa dạng từ bộ sưu tập Opus (Tiedemann, n.d.), bao gồm dữ liệu từ JW300 (Agic and Vulic 2019), Kinh Thánh, Tanzil và TED Talks (Cettolo, Girardi, and Federico 2012), nhằm thu thập thông tin song ngữ giữa tiếng Anh và năm ngôn ngữ LRL của châu Phi. Việc lựa chọn nguồn dữ liệu này là để đảm bảo rằng đánh giá hiệu suất NMT phản ánh một phạm vi rộng lớn của ngữ cảnh và sử dụng ngôn ngữ.

- Tiền xử lý dữ liệu:

Dữ liệu thu thập được chia thành ba tập: huấn luyện, phát triển và kiểm tra. Quá trình tiền xử lý bao gồm việc chọn ngẫu nhiên các đoạn dữ liệu để tạo ra tập phát triển và kiểm tra, cùng với việc lọc và chuẩn hóa dữ liệu để loại bỏ các đoạn trùng lặp. Mô hình SentencePiece được sử dụng để chia nhỏ dữ liệu thành subwords (Kudo and Richardson 2018), giúp cải thiện quá trình xử lý ngôn ngữ và dịch máy.

### **Các loại mô hình và đánh giá được sử dụng trong bài báo**

Các loại mô hình NMT khác nhau đã được huấn luyện để đánh giá các phương pháp tiếp cận LRLs:

- Standard neural machine translation: Mười mô hình cặp ngôn ngữ đơn lẻ được huấn luyện, mỗi cặp dành riêng cho từng ngôn ngữ SATOS ↔ En.
- Semi-Supervised neural machine translation: Các mô hình được huấn luyện sử dụng dữ liệu song ngữ gốc và dữ liệu tổng hợp qua back-translation (Bertoldi and Federico 2009).
- Transfer learning: Áp dụng việc học chuyển giao từ dữ liệu đa ngôn ngữ lớn để cải thiện hiệu suất mô hình cho từng cặp ngôn ngữ.
- Multilingual neural machine translation: Huấn luyện một mô hình đa ngôn ngữ duy nhất sử dụng tất cả dữ liệu SATOS ↔ En.

Các mô hình này được đánh giá trên nhiều bộ dữ liệu thử nghiệm, trong đó điểm số BLEU (Papineni, Roukos, Ward, and Zhu 2002) được sử dụng như là chỉ số đánh giá chính để đo lường chất lượng dịch thuật.

### **Kết quả và đánh giá**

Nghiên cứu đã chỉ ra rằng phương pháp multilingual neural machine translation cung cấp sự cải thiện đáng kể nhất, tăng trung bình đến 5 điểm BLEU, làm nổi bật sự hiệu quả của việc sử dụng dữ liệu ngôn ngữ đa dạng trong huấn luyện NMT. Điều này cho thấy tiềm năng của việc áp dụng các phương pháp học máy tiên tiến cho việc xử lý dịch thuật cho các ngôn ngữ LRL. Mở ra cánh cửa mới trong việc xóa bỏ rào cản ngôn ngữ, đặc biệt là đối với những ngôn ngữ ít được nghiên cứu, tăng cường sự kết nối và hiểu biết lẫn nhau giữa các nền văn hóa khác nhau.



## 2.2 Mô hình Transformer

### 2.2.1 Giới thiệu Transformer

Mô hình Transformer là một dạng kiến trúc mạng neural được sử dụng rộng rãi trong lĩnh vực xử lý ngôn ngữ tự nhiên và nhiều ứng dụng khác. Nó ra đời từ bài báo nghiên cứu "Attention is All You Need" (Vaswani et al. 2017) và đã trở thành một trong những phương pháp tiên tiến nhất trong lĩnh vực này.

Điểm nổi bật của Transformers là sự kết hợp giữa cơ chế attention và kiến trúc mạng neural không chứa các lớp kết nối tuần tự như LSTM (Ghojogh and Ghodsi 2023) hoặc GRU (Zhang, Xiong, and Su 2019). Thay vào đó, nó sử dụng các lớp self-attention để xác định mức độ quan trọng của từng phần tử trong chuỗi đầu vào đối với từng phần tử khác, sau đó áp dụng các phép biến đổi tương ứng.

Mô hình Transformer có hai thành phần chính: Encoder và Decoder. Encoder chịu trách nhiệm mã hóa thông tin đầu vào thành các biểu diễn ngữ cảnh, trong khi decoder tạo ra chuỗi đầu ra từ các biểu diễn này. Mỗi thành phần này lại chứa nhiều lớp của các module attention và mạng neural feedforward.

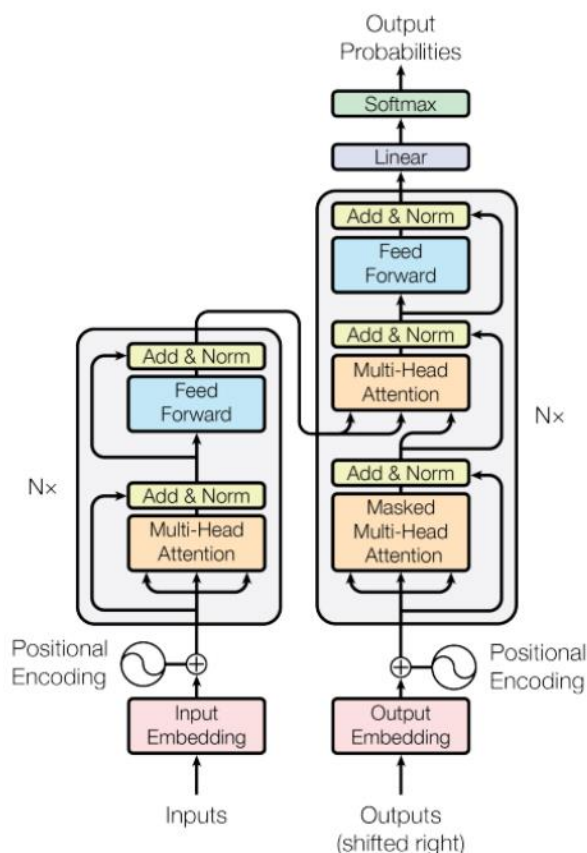
Với khả năng tự chú ý vào các phần tử quan trọng của dữ liệu đầu vào, mô hình Transformers thường cho hiệu suất tốt và có thể huấn luyện trên tập dữ liệu lớn mà không gặp vấn đề phụ thuộc vào thứ tự từ. Điều này đã làm cho chúng trở thành lựa chọn hàng đầu cho nhiều ứng dụng xử lý ngôn ngữ tự nhiên, bao gồm dịch máy, tóm tắt văn bản, sinh văn bản, và nhiều ứng dụng khác.

### 2.2.2 Kiến trúc mô hình

Mô hình Transformer có kiến trúc mã hóa - giải mã (Encoder-Decoder architecture) (Cho et al. 2014). Đây là cấu trúc cơ bản của mô hình, bao gồm hai phần chính là encoder và decoder. Encoder có nhiệm vụ chuyển đổi đầu vào, thường là dữ liệu dạng chuỗi như văn bản, thành dạng biểu diễn liên tục, còn được biết đến là vector đặc trưng. Mỗi phần tử trong chuỗi đầu vào được mã hóa thành một vector đặc trưng phức tạp, đại diện cho thông tin cần thiết mà decoder sẽ sử dụng để sinh ra chuỗi đầu ra.

Decoder có nhiệm vụ sinh ra chuỗi đầu ra từ vector đặc trưng mà nó nhận được từ encoder. Quá trình này diễn ra từng bước, mỗi lần sinh ra một phần tử của chuỗi đầu ra (từ hoặc ký tự). Điều đặc biệt ở đây là decoder hoạt động một cách tự hồi quy (auto-regressive) (Paschalidou et al. 2021), tức là dựa vào không chỉ thông tin từ vector đặc trưng nhận được từ encoder mà còn dựa vào những phần tử đã được sinh ra trước đó trong chuỗi đầu ra. Mỗi khi sinh ra một phần tử mới, decoder xem xét cả ngữ cảnh từ chuỗi đầu vào qua vector đặc trưng và ngữ cảnh từ phần đã được tạo ra của chuỗi đầu ra để dự đoán phần tử tiếp theo một cách chính xác nhất. Cách tiếp cận này giúp tối ưu hóa quá trình tạo ra chuỗi đầu ra, đảm bảo tính mạch lạc và ngữ cảnh phù hợp, tăng khả năng dự đoán từ tiếp theo dựa trên toàn bộ thông tin có sẵn từ cả đầu vào và quá trình tạo ra đầu ra đến thời điểm hiện tại.

Kiến trúc Transformer, như được minh họa trong Hình 2.1, không chỉ kế thừa cách tiếp cận mã hóa-giải mã truyền thống mà còn mang đến một cải tiến đáng chú ý đó là việc áp dụng kỹ thuật self-attention. Điều này cho phép mỗi phần tử trong chuỗi đầu vào có khả năng chú ý đến toàn bộ các phần tử khác trong cùng một chuỗi, từ đó nắm bắt được các mối quan hệ phức tạp giữa chúng. Đặc biệt, Transformer tích hợp các point-wise, fully connected layers vào cả hai phần encoder và decoder, nâng cao khả năng xử lý thông tin. Giúp Transformer hiệu quả hơn trong việc hiểu và xử lý ngữ cảnh.



Hình 2.1 Hình The Transformer - model architecture (Vaswani et al. 2017)

### Các lớp Encoder và Decoder

#### - Encoder

Encoder trong kiến trúc Transformer được cấu tạo từ 6 lớp giống hệt nhau, mỗi lớp gồm hai sub-layers. Sub-layer đầu tiên áp dụng kỹ thuật multi-head self-attention, cho phép mô hình nắm bắt được mối quan hệ giữa các phần tử trong cùng một chuỗi. Tiếp theo, sub-layer thứ hai là một position-wise fully connected feed-forward network, chịu trách nhiệm xử lý thông tin được tinh chỉnh bởi self-attention. Mỗi sub-layer được tăng cường bởi một residual connection (He et al. 2015), được áp dụng trước layer normalization (Ba, Kiros, and Hinton 2016). Điều này đồng nghĩa với việc output của mỗi sub-layer được tính theo công thức  $\text{LayerNorm}(x + \text{Sublayer}(x))$ , với  $x$  là input của sub-layer, và  $\text{Sublayer}(x)$  là kết quả xử lý của chính sub-layer đó. Cấu trúc này giúp cho việc huấn luyện mô hình trở nên ổn định và hiệu quả hơn. Để đảm bảo tương thích với residual connection, mỗi sub-layer trong mô

hình, kể cả embedding layer, đều tạo ra output với kích thước  $d_{\text{model}} = 512$ . Điều này giúp tăng cường khả năng tích hợp và xử lý thông tin hiệu quả.

- Decoder:

Trong kiến trúc Transformer, decoder được thiết kế để tinh chỉnh và sinh ra chuỗi đầu ra từ thông tin đã được mã hóa bởi encoder. Đặc biệt, decoder bao gồm 6 lớp giống hệt nhau, mỗi lớp chứa ba sub-layers chính: hai sub-layers tương tự như trong encoder và một sub-layer thêm vào dành cho multi-head self-attention lên output của encoder. Điều này cho phép decoder không chỉ xử lý thông tin nội bộ trong chuỗi đầu ra mà còn tích hợp thông tin từ chuỗi đầu vào đã được encoder xử lý.

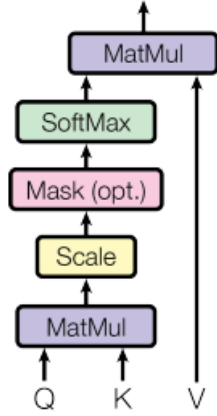
Sub-layer thứ ba trong decoder đặc biệt quan trọng vì nó thực hiện multi-head self-attention trên toàn bộ chuỗi output của encoder, giúp mỗi từ trong chuỗi đầu ra có thể chú ý và học hỏi từ toàn bộ chuỗi đầu vào. Điều này tăng cường khả năng của decoder trong việc nắm bắt các mối quan hệ phức tạp giữa chuỗi đầu vào và đầu ra.

Một trong những cải tiến quan trọng nhất trong decoder là cơ chế masking được áp dụng trong multi-head self-attention, ngăn không cho các vị trí trong chuỗi đầu ra chú ý đến các vị trí sau nó. Cơ chế này đảm bảo rằng tại mỗi bước sinh từ, mô hình chỉ sử dụng thông tin từ các từ trước đó và từ chuỗi đầu vào, từ đó duy trì tính logic và ngữ cảnh trong quá trình sinh chuỗi đầu ra.

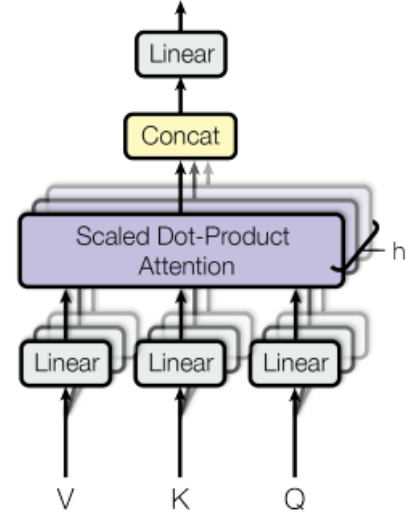
### 2.2.3 Cơ chế Attention

Chức năng của cơ chế chú ý (attention mechanism) được mô tả như một hàm ánh xạ giữa một query và một tập hợp các key-value pairs để tạo ra một đầu ra, nơi mà query, các keys, values và đầu ra đều là các vector. Kết quả cuối cùng dựa vào quá trình cộng các values lại với nhau, nhưng mỗi value trước khi được cộng sẽ được nhân với một số nhất định gọi là trọng số. Trọng số này không giống nhau giữa các values mà được xác định dựa trên mức độ liên quan giữa query và key tương ứng với value đó. Càng liên quan nhiều thì trọng số càng lớn, giúp cho value đó có ảnh hưởng nhiều hơn đến kết quả cuối cùng. Hình 2.2 biểu diễn hai cấu trúc quan trọng trong cơ chế Attention là Scaled Dot-Product Attention và Multi-Head Attention.

Scaled Dot-Product Attention



Multi-Head Attention



Hình 2.2 Scaled Dot-Product Attention & Multi-Head Attention (Vaswani et al. 2017)

### Scaled Dot-Product Attention

Đầu vào của cơ chế chú ý bao gồm các queries (Q) và keys (K) với kích thước chiều là  $d_k$ , và values (V) với kích thước chiều là  $d_v$ .

Để tính toán trọng số chú ý, mô hình thực hiện tích vô hướng (dot product) giữa mỗi query với tất cả các key, sau đó chia cho  $\sqrt{d_k}$  để đánh tỉ lệ kết quả và áp dụng hàm softmax. Việc chia tỉ lệ này là cần thiết để khi  $d_k$  lớn, không làm cho value của tích vô hướng quá lớn, dẫn đến vấn đề với hàm softmax do gradient có thể trở nên rất nhỏ.

Công thức (2.1) là của cơ chế Attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V \quad (2.1)$$

Trong đó  $Q$ ,  $K$  và  $V$  đại diện cho ma trận của query, key và value tương ứng.  $QK^T$  là tích vô hướng giữa  $Q$  và chuyển vị của  $K$ , được chia tỉ lệ trước khi áp dụng softmax để xác định trọng số chú ý.

Khi so sánh dot-product attention với additive attention (Bahdanau, Cho, and Bengio 2016), dot-product attention không khác gì với additive attention ngoại trừ

việc áp dụng tỉ lệ  $\frac{1}{\sqrt{d_k}}$ . Additive attention sử dụng một feedforward neural network với một hidden layer để tính toán hàm tương thích mặc dù cả hai có độ phức tạp lý thuyết tương tự, nhưng cơ chế dot-product attention thực hiện nhanh hơn và hiệu quả không gian hơn trong thực tế. Lí do cần đến tỉ lệ là vì đối với các values nhỏ của  $d_k$ , hai cơ chế này hoạt động tương tự nhau. Tuy nhiên, khi  $d_k$  lớn (Britz et al. 2017), additive attention lại hiệu quả hơn nếu không có tỉ lệ. Nguyên nhân là vì các tích vô hướng có thể tăng lớn, đẩy hàm softmax vào vùng có độ dốc rất nhỏ, làm giảm khả năng học của mô hình. Để khắc phục điều này, các tích vô hướng được chia tỉ lệ bằng  $\frac{1}{\sqrt{d_k}}$ , giúp ổn định các gradient và làm cho quá trình học trở nên dễ dàng hơn.

### Multi-Head Attention

Trong mô hình Transformer, nếu sử dụng một đầu chú ý duy nhất (single head attention), thì toàn bộ dữ liệu sẽ được xử lý bởi một đầu chú ý duy nhất đó. Điều này có thể hạn chế vì một đầu chú ý duy nhất chỉ có thể tập trung vào một loại mối quan hệ cụ thể trong dữ liệu.

Multi-Head Attention sẽ giải quyết vấn đề này bằng cách tạo ra  $h$  phiên bản khác nhau của hàm chú ý, mỗi phiên bản có một bộ trọng số học được riêng biệt, cho phép mô hình tập trung vào nhiều loại mối quan hệ khác nhau đồng thời.

Mỗi đầu chú ý thực hiện hàm chú ý một cách độc lập:

Đầu tiên, mỗi đầu chú ý chiếu query ( $Q$ ), key ( $K$ ), và value ( $V$ ) vào không gian có kích thước thấp hơn sử dụng các ma trận trọng số riêng biệt  $W_i^Q$ ,  $W_i^K$  và  $W_i^V$ .

Tiếp theo, cơ chế chú ý được áp dụng lên mỗi phiên bản đã được chuyển đổi này. Kết quả là một loạt các vector đầu ra, mỗi cái được tạo ra từ một đầu chú ý riêng biệt.

Các vector đầu ra từ mỗi đầu chú ý sau đó được nối lại với nhau (concatenation) để tạo thành một vector duy nhất. Vector này sau đó được nhân một lần nữa qua ma trận trọng số  $W^O$  để thu được đầu ra cuối cùng của lớp Multi-Head Attention.

Công thức toán học của Multi-Head Attention được định nghĩa trong công thức (2.2):

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2.2)$$

trong đó mỗi  $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

- Sử dụng nhiều đầu chú ý cho phép mô hình nhìn vào thông tin từ nhiều góc độ khác nhau, giúp nắm bắt được sự phong phú trong dữ liệu mà một đầu chú ý đơn lẻ có thể sẽ bỏ lỡ.

### **Cách thức mà mô hình Transformer sử dụng cơ chế Multi-head Attention**

- Trong các lớp Encoder - Decoder Attention:

Queries trong lớp này đến từ decoder, giúp xác định phần nào của input cần được chú ý để tạo ra từng token mới. Keys và Values lại đến từ encoder, mang thông tin về toàn bộ chuỗi input. Cơ chế này cho phép mỗi bước trong decoder nhìn thấy và chú ý đến toàn bộ chuỗi input, từ đó lựa chọn thông tin liên quan nhất khi dự đoán mỗi token mới.

- Trong Encoder:

Self-attention cho phép mỗi từ (hoặc token) trong input chú ý đến tất cả các từ khác trong cùng một chuỗi. Điều này giúp mô hình hiểu được mối quan hệ giữa các từ, bất kể khoảng cách về vị trí giữa chúng, như quan hệ ngữ pháp hay ngữ nghĩa.

- Trong Decoder:

Self-attention hoạt động giống như trong encoder nhưng có một hạn chế chỉ cho phép mỗi token chú ý đến các token đã được giải mã, tức là những token trước đó trong chuỗi đầu ra. Để làm được điều này, mô hình sẽ ẩn thông tin của các token chưa được xử lý (những token sau đó trong chuỗi đầu ra) bằng cách đặt value là  $-\infty$  cho chúng trước khi thực hiện softmax để tính toán trọng số attention. Cách này ngăn cản chúng ảnh hưởng đến kết quả attention và đảm bảo rằng mỗi dự đoán chỉ dựa trên thông tin đã biết.

### 2.2.4 Position-wise Feed-Forward Networks

Mỗi lớp trong bộ mã hóa và giải mã của Transformer chứa một mạng feed forward , được áp dụng riêng biệt cho mỗi vị trí trong chuỗi đầu vào hoặc đầu ra. Điều này đảm bảo rằng mỗi vị trí được xử lý theo cách giống hệt nhau nhưng với thông tin cụ thể của vị trí đó.

Mạng này bao gồm hai phép biến đổi tuyến tính. Đầu tiên, đầu vào  $x$  được nhân với ma trận trọng số  $W_1$  và cộng với vector độ lệch (bias)  $b_1$  . Sau đó, hàm kích hoạt ReLU được áp dụng, và cuối cùng, kết quả lại được nhân với ma trận trọng số thứ hai  $W_2$  và cộng với độ lệch  $b_2$  để tạo ra đầu ra cuối cùng của mạng feed forward.

Hàm ReLU được biểu diễn bằng công thức  $\max(0, z)$ , nơi  $z$  là đầu vào cho hàm kích hoạt, và kết quả là đầu ra của hàm ReLU không bao giờ âm.

Công thức toán học cho mạng feed forward được cho công thức (2.3):

$$\text{FFN}(x) = \max(0, xW_1 + b_1) W_2 + b_2 \quad (2.3)$$

Trong đó  $W_1$  và  $W_2$  là trọng số của hai phép biến đổi tuyến tính, và  $b_1$  và  $b_2$  là các bias. Hàm  $\max(0, z)$  là hàm ReLU, loại bỏ giá trị âm bằng cách thay thế chúng bằng 0.

Mặc dù các biến đổi tuyến tính giống nhau cho tất cả các vị trí trong chuỗi, nhưng mô hình sử dụng các tham số khác nhau cho mỗi lớp trong mạng. Điều này có nghĩa là mỗi lớp feed forward học một hàm biểu diễn tuyến tính khác nhau.

Input và output của mạng feed forward có cùng kích thước, là  $d_{model}$ , thường là 512. Tuy nhiên, hidden layer của mạng feed forward có kích thước lớn hơn là  $d_{ff}$ , thường là 2048. Điều này tạo ra một không gian biểu diễn tạm thời có độ lớn lớn hơn, cho phép mạng có đủ không gian để học các biểu diễn phức tạp trước khi thu hẹp lại về kích thước ban đầu.

Mạng feed forward đóng vai trò cung cấp khả năng biểu diễn phi tuyến tính và phức tạp hơn, điều mà chỉ sử dụng cơ chế chú ý không thể đạt được. Mỗi vị trí trong chuỗi đầu vào được biến đổi một cách độc lập nhưng theo cùng một quy trình, cho phép mô hình nắm bắt được các mối quan hệ phức tạp không chỉ dựa trên ngữ cảnh xung quanh.



### 2.2.5 Embeddings and Softmax và Positional Encoding

#### Embeddings

- Giống như các mô hình chuyển đổi chuỗi khác, mô hình Transformer sử dụng các embedding đã học để chuyển đổi token đầu vào và token đầu ra thành các vector có kích thước chiều là  $d_{model}$ . Embedding giúp chuyển đổi token, là những đơn vị thông tin cơ bản như từ hoặc ký tự, thành một không gian vector liên tục mà mô hình có thể xử lý được. Trong các lớp embedding, trọng số được nhân với  $\sqrt{d_{model}}$ , điều này là một phần của việc chuẩn hóa trọng số để tối ưu hóa quá trình học.

#### Softmax

Hàm softmax được sử dụng để chuyển đổi đầu ra từ bộ giải mã thành các xác suất của token tiếp theo dự đoán.

Vì mô hình Transformer không sử dụng cơ chế hồi quy (recurrence) hay tích chập (convolution), nên nó cần một cách để nhận biết vị trí tương đối hoặc tuyệt đối của từng token trong chuỗi đầu vào. Để thực hiện điều này, mô hình thêm positional encoding vào input embedding ở phần dưới cùng của encoder và decoder. Positional encoding có cùng kích thước chiều  $d_{model}$  với embedding, cho phép chúng có thể được cộng lại một cách trực tiếp.

Các positional encoding được tạo ra bằng cách sử dụng các hàm sin và cos với các tần số khác nhau công thức (2.4) và (2.5):

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (2.4)$$

$$PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (2.5)$$

trong đó  $pos$  là vị trí của từng token trong chuỗi, và  $i$  là chỉ số chiều. Điều này tạo ra một dạng sóng hình sin và cos cho mỗi chiều trong không gian biểu diễn của embedding, với các bước sóng tạo thành một cấp số nhân từ  $2\pi$  đến  $10000.2\pi$ . Việc sử dụng hàm sin và cos để tạo ra positional encoding giúp mô hình Transformer dễ dàng nhận biết được vị trí tương đối giữa các từ trong câu. Điều này có nghĩa là, mô

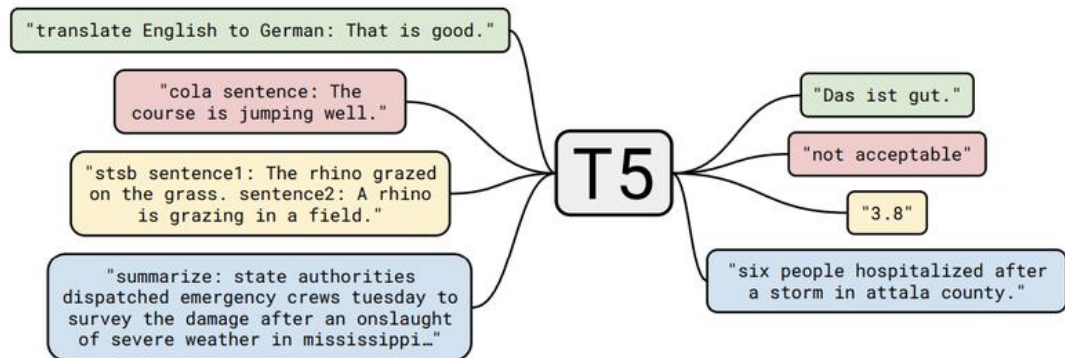
hình có thể hiểu được khoảng cách giữa các từ chỉ bằng cách xem xét sự thay đổi trong các giá trị của positional encoding, mà không cần phụ thuộc vào việc các từ nằm ở đâu trong câu. Cách làm này giúp Transformer xử lý và hiểu thông tin về vị trí của từ một cách linh hoạt và chính xác hơn.

## 2.3 Mô hình T5 và Mô hình mT5

### 2.3.1 Mô hình T5

#### Giới thiệu về T5

Mô hình T5 được phát triển với mục tiêu chính là tận dụng sức mạnh của học chuyển giao trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Bằng cách tiền huấn luyện một mô hình trên một nhiệm vụ giàu dữ liệu, mô hình này có thể phát triển khả năng và kiến thức đa năng, sau đó có thể được chuyển giao cho các nhiệm vụ khác nhau. Một cách độc đáo, T5 xử lý mọi vấn đề xử lý văn bản như một vấn đề "văn bản-sang-văn bản", cho phép cùng một mô hình và kỹ thuật được áp dụng trên một loạt các nhiệm vụ NLP khác nhau mà không cần tinh chỉnh đáng kể.



Hình 2.3 A diagram of our text-to-text framework (Raffel et al. 2023)

Hình 2.3 cho thấy sơ đồ mô tả cách mô hình T5 dựa trên Transformer có thể xử lý nhiều loại nhiệm vụ khác nhau bằng cách định dạng chúng dưới dạng các câu input và các câu đầu ra output theo những yêu cầu cụ thể như:

Dịch từ tiếng Anh sang tiếng Đức: câu "That is good." Bản dịch: "Das ist gut."

Câu "The course is jumping well." Kiểm tra chất lượng của bản dịch với bộ dữ liệu kiểm tra chất lượng câu (CoLA) và mô hình đánh giá nó là "not acceptable."

Đánh giá câu: "The rhino grazed on the grass." và "A rhino is grazing in a field." Đánh giá sự tương tự của hai câu

Tóm tắt: câu "state authorities dispatched emergency crews Tuesday to survey the damage after an onslaught of severe weather in Mississippi..." và mô hình tạo ra một bản tóm tắt "six people hospitalized after a storm in attala county."

### **Kiến trúc mô hình transformer**

T5 dựa trên kiến trúc Transformer (Vaswani et al. 2017), được biết đến với hiệu quả trong dịch máy và đã được mở rộng ứng dụng trong nhiều nhiệm vụ NLP khác. Kiến trúc này gồm hai phần chính: Encoder và Decoder, với cơ chế tự chú ý là khối xây dựng chính. Tự chú ý cho phép mô hình xem xét mối quan hệ giữa các từ trong cùng một câu hoặc đoạn văn, cung cấp một cách hiệu quả để xử lý thông tin ngữ nghĩa.

### **Tập dữ liệu Colossal Clean Crawled Corpus (C4)**

Để tiền huấn luyện mô hình T5, nhóm nghiên cứu đã phát triển "Colossal Clean Crawled Corpus" (C4), một tập hợp dữ liệu văn bản lớn được sử dụng làm dữ liệu không dán nhãn. C4 được tạo ra từ Common Crawl và đã qua một quá trình lọc chặt chẽ để đảm bảo chất lượng và tính sạch sẽ của văn bản, làm cơ sở cho việc học không giám sát.

### **Phương pháp xử lý các nhiệm vụ**

T5 tiếp cận các nhiệm vụ NLP bằng cách chuyển tất cả chúng thành định dạng "văn bản-sang-văn bản". Cách tiếp cận này không chỉ đơn giản hóa quy trình xử lý nhiều loại nhiệm vụ khác nhau mà còn cho phép mô hình sử dụng một mục tiêu huấn luyện nhất quán. Ví dụ, mô hình có thể được yêu cầu dịch một câu từ tiếng Anh sang tiếng Đức hoặc phân loại một câu dựa trên cảm xúc, tất cả đều dùng cùng một phương pháp.

### **Định dạng đầu vào và đầu ra**

Mỗi nhiệm vụ được xác định bằng một tiền tố cụ thể cho nhiệm vụ, được thêm vào trước chuỗi đầu vào. Việc này chỉ định rõ nhiệm vụ mà mô hình cần thực hiện và cho phép mô hình tinh chỉnh trên các nhiệm vụ cụ thể bằng cách sử dụng đầu vào và đầu ra tương ứng.

### **2.3.2 Mô hình mT5**

#### **Giới thiệu mT5**

Trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), việc sử dụng học chuyển giao đã trở nên phổ biến, với các mô hình được tiền huấn luyện trên dữ liệu lớn trước khi được tinh chỉnh cho các nhiệm vụ cụ thể. mT5 được giới thiệu như một biến thể đa ngôn ngữ của mô hình T5, mục tiêu là mở rộng khả năng của mô hình cho hơn 100 ngôn ngữ, giúp giải quyết vấn đề về sự thiếu hụt mô hình tiền huấn luyện cho các ngôn ngữ khác ngoài tiếng Anh.

#### **Nền tảng về T5 và C4**

T5, viết tắt của "Text-to-Text Transfer Transformer", sử dụng một định dạng "văn bản-đến-văn bản" thống nhất cho tất cả các nhiệm vụ NLP, từ dịch máy đến phân loại cảm xúc. T5 được tiền huấn luyện trên tập dữ liệu C4, một tập hợp lớn dữ liệu văn bản tiếng Anh từ web, và được thiết kế để có thể mở rộng quy mô, với các phiên bản từ 60 triệu đến 11 tỷ tham số (Raffel et al. 2023).

#### **mC4 và mT5**

##### **- mC4**

Khác biệt với C4, mC4 bao gồm dữ liệu trong 101 ngôn ngữ từ Common Crawl, được xử lý bằng cách sử dụng heuristics để đảm bảo chất lượng và loại bỏ nội dung không mong muốn. Việc lựa chọn dữ liệu từ mỗi ngôn ngữ được cân nhắc kỹ lưỡng, với mục tiêu tạo ra một bộ dữ liệu cân đối giữa các ngôn ngữ có nguồn lực cao và thấp.

##### **- mT5**

Mục tiêu của mT5 là kế thừa và mở rộng khả năng của T5 sang đa ngôn ngữ, giữ nguyên các lợi ích của T5 như định dạng văn bản-sang-văn bản linh hoạt và kiến

trúc dựa trên Transformer. mT5 được tiền huấn luyện trên mC4 và được thiết kế để xử lý một loạt các nhiệm vụ NLP trong hơn 100 ngôn ngữ, giúp giảm thiểu sự thiên vị về ngôn ngữ và mở rộng khả năng sử dụng mô hình tiền huấn luyện trong cộng đồng NLP toàn cầu.

Một vấn đề cụ thể với các mô hình đa ngôn ngữ tiền huấn luyện là việc chúng thường xuyên dịch sai một phần dự đoán của mình sang ngôn ngữ sai. Để giải quyết vấn đề này, mT5 sử dụng một quy trình đơn giản trong đó trộn dữ liệu tiền huấn luyện không được gán nhãn vào quá trình điều chỉnh mịn, giúp giảm thiểu đáng kể vấn đề này.

## 2.4 Chỉ số đánh giá mô hình tự động BLEU

### Chỉ số BLEU

BLEU là một chỉ số gán một điểm số cho một translation để chúng tôi biết mức độ tốt hơn so với các bản dịch thông thường. Ví dụ:

<div data-bbox="395 1182 667 1249" style="border: 1px solid black; border-radius: 10px; padding: 5px; background-color: #e6f2ff;">Tengo veinticinco años</div> <div data-bbox="454 1317 593 1384" style="text-align: center;">Source text (Spanish)</div>	<div data-bbox="722 1182 1042 1249" style="border: 1px solid black; border-radius: 10px; padding: 5px; background-color: #e6ffe6;">I have twenty five years</div> <div data-bbox="715 1328 1053 1361" style="text-align: center;">Machine generated traslation</div>	<div data-bbox="1114 1104 1433 1171" style="border: 1px solid black; border-radius: 10px; padding: 5px; background-color: #ffe6e6;">I am twenty five years old</div> <div data-bbox="1114 1238 1433 1305" style="border: 1px solid black; border-radius: 10px; padding: 5px; background-color: #ffe6e6;">I am twenty five</div> <div data-bbox="1153 1328 1409 1395" style="text-align: center;">Human translation (reference translation)</div>
---	--	--

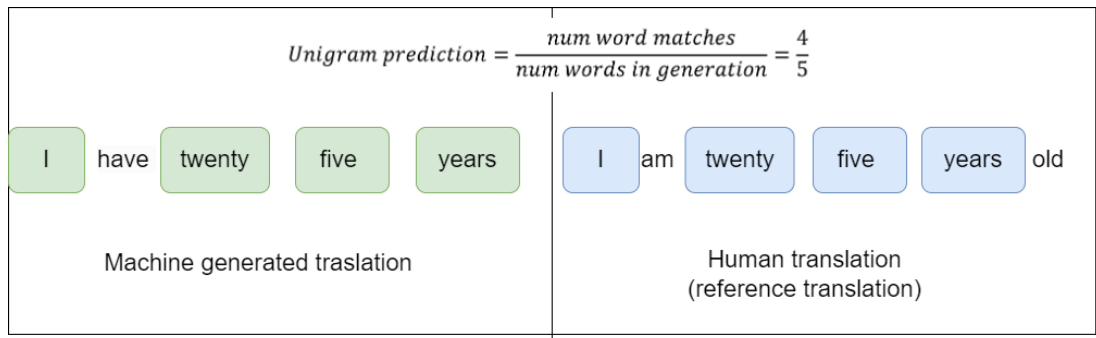
Trong ví dụ trên đề tài có mẫu tiếng Tây Ba Nha được dịch sang Tiếng Anh. Nếu so sánh bản dịch với một số bản dịch do con người dịch có thể thấy mô hình khá tốt nhưng còn mắc lỗi phổ biến. Từ “tengo” trong tiếng Tây Ban Nha có nghĩa là “have” trong tiếng Anh nhưng trong ngữ cảnh dịch như trên thì không phù hợp và nó không được tự nhiên.

Vì thế cần tính chất lượng của các bản dịch tự động từ mô hình machine translation. BLEU là thực hiện so sánh n-grams của bản dịch được tạo từ máy và n-grams của tập dữ liệu.

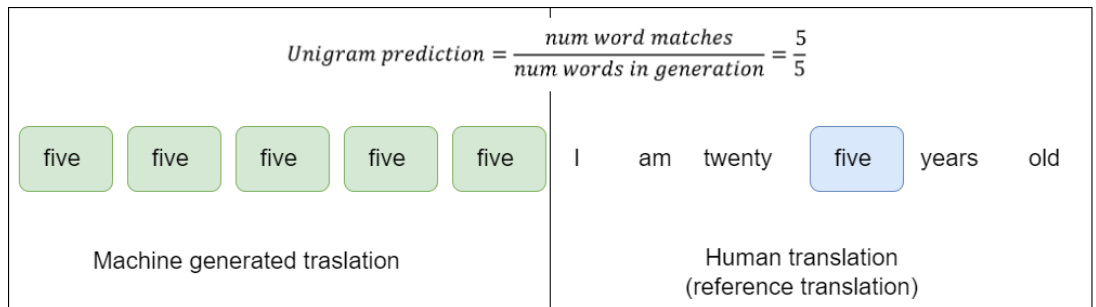
### Điểm Unigram

Unigram tương ứng với từng từ riêng lẻ. Cách tính là đếm số từ phù hợp với bản machine translate và reference translation. Sau đó chuẩn hóa số bằng cách chia số từ cho tổng số.

$$\text{Unigram prediction} = \frac{\text{num word matches}}{\text{num words in generation}}$$



Đếm số từ phù hợp với bản machine translate và reference translation. Sau đó chuẩn hóa số bằng cách chia số từ cho tổng số từ trong dịch máy. Ở ví dụ trên, chúng tôi tìm được Unigram là  $\frac{4}{5} = 80\%$ . Nhận xét thấy độ chính xác từ 0.8 đến 1 nghĩa là bản dịch này tốt. Nhưng các mô hình dịch đôi khi bị lặp lại cùng một từ nhiều lần.



Ở câu trên nếu chỉ đếm số lượng từ trùng khớp thì nhận được kết quả thực sự cao mặc dù bản dịch không đúng với bản tham chiếu. Nếu tính bản dịch của machine generated translation thì chúng tôi có kết quả là hoàn hảo (=1). Để xử lý vấn đề này, BLEU đã sử dụng modified precision để giảm bớt số lần đếm số từ. Từ đó chúng tôi có  $\text{Unigram prediction} = \frac{1}{5}$ .

Một vấn đề khác là unigram không tính đến thứ tự các từ trong câu.

$\text{Unigram prediction} = \frac{\text{num word matches}}{\text{num words in generation}} = \frac{4}{5} ?$	
<div>years five twenty have I</div> <div>Machine generated traslation</div>	<div>I am twenty five years old</div> <div>Human translation (reference translation)</div>

Giải pháp thực nghiệm để cải thiện và điều chỉnh độ chính xác của chỉ số BLEU là N-gram precision bằng cách chia văn bản thành văn bản con có độ dài N.

### Kỹ thuật N-gram precision

Một câu nguồn (source) qua mô hình có thể tạo ra nhiều câu đích (target). Chúng tôi gọi câu đích là một sentence. Đầu tiên tính toán n-gram tương ứng từng sentence, sau đó cộng số lượng n-gram đã cắt bớt cho câu machine translation để tính điểm chính xác cho từng thay đổi trong câu target.

Tính xác suất xuất hiện của một câu trong bài báo N-gram Language Models (“N-Gram Language Models.Pdf,” n.d.) có công thức (2.6):

$$\begin{aligned}
 P(X_1 \dots X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_{1:2}) \dots P(X_n|X_{1:n-1}) \\
 &= \prod_{k=1}^n P(X_k|X_{1:k-1})
 \end{aligned} \tag{2.6}$$

Áp dụng chain rule chúng tôi được công thức (2.7):

$$\begin{aligned}
 P(w_{1:n}) &= P(w_1)P(w_2|w_1)P(w_3|w_{1:2}) \dots P(w_n|w_{1:n-1}) \\
 &= \prod_{k=1}^n P(w_k|w_{1:k-1}) = \prod_{k=1}^n \frac{\text{count}(w_1 \dots w_k)}{\text{count}(w_1 \dots w_{k-1})}
 \end{aligned} \tag{2.7}$$

### Công thức tính điểm BLEU:

Đầu tiên tính Geometric Average Precision Score (điểm trung bình hình học) của tập dữ liệu sau đó nhân kết quả hệ số brevity penalty. Trước tiên tính trung bình hình học của các chỉ số n-gram precisions ( $p_n$ ), sử dụng n-gram cho đến độ dài thứ N và giá trị  $\text{weight}(w_n)$ .

Tính Geometric Average Precision Score bằng công thức (2.8):

$$\text{Geometric Average Precision}(N) \quad (2.8)$$

$$\begin{aligned} &= \exp \left( \sum_{n=1}^N w_n \cdot \log p_n \right) \\ &= \prod_{n=1}^N p_n^{w_n} = (p_1)^{\frac{1}{4}} \cdot (p_2)^{\frac{1}{4}} \cdot (p_3)^{\frac{1}{4}} \cdot (p_4)^{\frac{1}{4}} \end{aligned}$$

Tính Brevity Penalty bằng công thức (2.9) và tính BLEU bằng công thức (2.10) trong bài (Papineni, Roukos, Ward, Zhu, et al. 2002):

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases} \quad (2.9)$$

$$BLEU = BP \cdot \exp \left( \sum_{n=1}^N w_n \cdot \log p_n \right) \quad (2.10)$$

Công thức được tính rõ ràng hơn khi dùng hàm log công thức (2.11):

$$\log BLEU = \min \left( 1 - \frac{r}{c}, 0 \right) + \sum_{n=1}^N w_n \cdot \log p_n \quad (2.11)$$

Trong đó:  $c$  là predicted length = số lượng từ có trong machine translation sentence,  $r$  là target length = số lượng từ có trong reference translation sentence.

Đánh giá BLEU, chỉ số BLEU dao động từ 0 đến 1. Ít bản dịch nào đạt được điểm 1 trừ khi chúng giống nhau với một bản dịch target. Vì lý do này, ngay cả một dịch giả cũng không nhất thiết phải được 1 điểm. Nếu càng có nhiều bản dịch target trên mỗi câu thì điểm số càng cao.

#### **Ưu điểm của BLEU:**

- Dễ tính toán và được nguyên cứu rộng rãi trong nguyên cứu để so sánh các mô hình khác trên điểm chuẩn.
- BLEU được đánh giá hiệu quả trong việc đánh giá chất lượng học máy, đặc biệt là các ngôn ngữ có nguồn tài nguyên hạn chế.



**Nhược điểm:**

- Nó không kết hợp ngữ nghĩa của câu, khó khăn khi gặp phải ngữ pháp không phải tiếng Anh.
- BLEU có thể nhạy cảm với các thay đổi nhỏ trong khi dịch, dẫn đến kết quả không đúng với dự đoán. Khó so sánh khi tokenizer khác nhau.

**2.5 Lý thuyết về các kỹ thuật xử lý dữ liệu****2.5.1 Data Cleaning**

Là quá trình chuẩn bị và làm sạch dữ liệu ngôn ngữ tự nhiên giúp cải thiện hiệu suất của các mô hình NLP. Data cleaning là bước quan trọng trong quy trình xử lý dữ liệu, giúp loại bỏ nhiễu và dữ liệu không liên quan, làm cho dữ liệu đầu vào trở nên chính xác và dễ quản lý hơn. Một số ví dụ của data cleaning trong NLP:

**Loại bỏ nhiễu (noise removal)**

Nhiễu là bất kỳ thông tin không liên quan hoặc không mong muốn trong dữ liệu, bao gồm các ký tự đặc biệt (ví dụ: dấu ngoặc kép, dấu ngắt dòng, ký tự Unicode, ...), số (ví dụ: "2023-11-16" trong một đoạn văn bản về thời tiết, ...) , khoảng trắng thừa, v.v. Loại bỏ nhiễu giúp chú trọng vào thông tin có giá trị của dữ liệu đầu vào.

**Chuẩn hóa văn bản (text normalization)**

Chuẩn hóa là quá trình biến đổi văn bản thành một dạng đồng nhất. Công việc này bao gồm việc chuyển đổi tất cả chữ cái về dạng chữ thường (lowercasing) ví dụ: từ "VIỆT NAM" thành "việt nam", loại bỏ dấu câu bỏ đi các dấu như dấu chấm, dấu phẩy, dấu chấm hỏi,... Ví dụ: "Xin chào! Bạn tên gì?" trở thành "Xin chào Bạn tên gì", và áp dụng lemmatization hoặc stemming (Khyani and B S 2021) để đưa từ về dạng gốc của chúng, ví dụ: "đi", "địa", "duy chuyển" đều được chuyển về "đi".

**Loại bỏ stop words**

Stop words, như "và", "là", "trong", không chứa nhiều thông tin ngữ nghĩa quan trọng và thường được loại bỏ trong quá trình xử lý ngôn ngữ tự nhiên. Việc loại bỏ chúng giúp giảm kích thước dữ liệu và giảm tải không gian lưu trữ. Giúp tăng tỷ lệ thông tin có giá trị so với thông tin không cần thiết, giúp các thuật toán NLP hoạt

động hiệu quả hơn bằng cách tập trung vào các từ có ý nghĩa chính. Nói cách khác, loại bỏ stop words không chỉ giúp làm sạch dữ liệu mà còn nâng cao khả năng của mô hình trong việc phân tích và hiểu văn bản, qua đó cải thiện đáng kể chất lượng và tốc độ xử lý.

### **Phát hiện và sửa lỗi chính tả**

Phát hiện và sửa lỗi chính tả là bước quan trọng để cải thiện chất lượng dữ liệu và hiệu suất của mô hình. Lỗi chính tả có thể làm giảm giá trị của dữ liệu, vì vậy việc áp dụng công cụ tự động để tìm và khắc phục những lỗi này là cần thiết. Điều này không chỉ giúp làm sạch dữ liệu mà còn tăng cường độ chính xác và độ tin cậy của thông tin, làm cho các thuật toán NLP hoạt động hiệu quả hơn.

### **Phát hiện và loại bỏ trùng lặp**

Trong quá trình xử lý dữ liệu văn bản, việc phát hiện và loại bỏ các phần trùng lặp là bước quan trọng để đảm bảo tính chính xác và hiệu quả của phân tích. Dữ liệu văn bản thường chứa các đoạn hoặc mục có nội dung giống nhau, nếu không được loại bỏ, có thể dẫn đến việc phân tích bị lệch lạc, ảnh hưởng đến kết quả của các thuật toán học máy và gây lãng phí không gian lưu trữ. Việc loại bỏ trùng lặp không chỉ giúp cải thiện độ chính xác trong việc phân tích dữ liệu mà còn tăng hiệu quả cho các mô hình học máy bằng cách giảm thiểu sự phức tạp và tăng tốc độ xử lý dữ liệu.

## **2.5.2 Tokenization**

Tokenization là bước phân chia văn bản đầu vào thành các đơn vị nhỏ hơn như từ, cụm từ hoặc ký tự, làm cho việc xử lý bởi mô hình học sâu trở nên thuận tiện hơn. Trong quá trình xử lý ngôn ngữ tự nhiên (NLP) và cụ thể là trong dịch máy (NMT), tokenization đóng vai trò quan trọng bởi nó ảnh hưởng đến cách mô hình học và dự đoán dữ liệu. Quá trình này giúp định dạng lại văn bản một cách hợp lý, từ đó tối ưu hóa hiệu suất và cải thiện khả năng hiểu và xử lý ngôn ngữ của mô hình.

### **Những loại tokenization:**

- Thuật toán tokenization dựa trên từ (word-level tokenization algorithm), tách câu thành các từ riêng lẻ, ví dụ như ["Let", "us", "learn", "tokenization."]. Phương pháp này, mặc dù đơn giản và dễ hiểu, có thể không phải là lựa chọn tối ưu cho các

ngôn ngữ có cấu trúc ngữ pháp phức tạp hoặc với không gian từ vựng rộng lớn. Điều này là do mỗi từ được xem xét độc lập, không tính đến ngữ cảnh hoặc cấu trúc ngữ pháp, có thể khiến mô hình cần sử dụng nhiều tài nguyên hơn để học và hiểu ngôn ngữ. Dù ưu điểm rõ ràng là sự đơn giản và dễ áp dụng, nhưng nhược điểm chính là khả năng tiếp cận hạn chế với ngôn ngữ có cấu trúc phức tạp, khi mà không gian từ vựng lớn cũng làm tăng yêu cầu về tài nguyên cho việc học.

- Tokenization theo cụm từ (Phrase-level Tokenization) chia văn bản thành các cụm từ mang ý nghĩa, điều này giúp thu hẹp không gian từ vựng và tăng cường khả năng của mô hình trong việc hiểu ngữ cảnh. Tuy nhiên, thách thức lớn trong phương pháp này là việc xác định chính xác ranh giới của các cụm từ, vì điều này yêu cầu sự hiểu biết sâu sắc về ngữ pháp và ngữ nghĩa của ngôn ngữ. Điều này có thể làm tăng độ phức tạp cho mô hình học sâu, vì nó phải được thiết kế để có thể nhận diện và xử lý các cụm từ này một cách chính xác.

- Tokenization dựa trên ký tự (Character-based tokenization algorithm) là một phương pháp phân đoạn văn bản, nơi nó tách mỗi câu thành các ký tự riêng lẻ, tức là mỗi chữ cái được xem như một đơn vị độc lập. Kỹ thuật này rất phù hợp với ngôn ngữ có số lượng ký tự giới hạn và giúp mô hình học sâu dễ dàng nắm bắt và học các quy tắc ngữ âm cũng như cấu trúc ngôn ngữ cơ bản. Đặc biệt, đối với các ngôn ngữ như tiếng Thái, nơi mà cấu trúc từ và ngữ âm có thể phức tạp và không theo quy tắc linh hoạt như trong tiếng Anh, việc sử dụng character-based tokenization giúp giảm bớt trở ngại trong việc xác định ranh giới từ, đồng thời cung cấp một phương pháp đơn giản nhưng hiệu quả để mô hình có thể học ngôn ngữ. Phương pháp này cũng giúp làm giảm kích thước của không gian từ vựng mà mô hình cần xử lý, từ đó tối ưu hóa quá trình học và cải thiện hiệu suất của mô hình trong việc hiểu và tạo ra văn bản.

- Subword Tokenization (Yang,2024) là một kỹ thuật trong NLP kết hợp ưu điểm của việc phân loại token dựa trên từ và ký tự. Cách làm này phân chia các từ thành các đơn vị nhỏ hơn, thậm chí là các thành phần cơ bản của từ, giúp mô hình dễ dàng xử lý được bộ từ vựng đa dạng và phong phú hơn. Điều này đặc biệt quan trọng

trong việc giảm thiểu vấn đề khi gặp các từ mới hoặc hiếm gặp mà mô hình chưa từng thấy trong quá trình training.

BPE (Byte Pair Encoding) (Berglund and van der Merwe 2023), SentencePiece (Kudo, Richardson, and Kudo and Richardson 2018), và WordPiece là ba phương pháp Subword Tokenization phổ biến, mỗi phương pháp có cách tiếp cận riêng biệt nhưng chung mục đích: tối ưu hóa việc xử lý và hiểu từ vựng.

BPE là một phương pháp tokenization hiệu quả, hoạt động bằng cách tìm và kết hợp các cặp byte (thường là ký tự hoặc nhóm ký tự) xuất hiện cạnh nhau thường xuyên trong một tập dữ liệu. Quá trình này được lặp lại nhiều lần, tạo ra các token mới từ sự kết hợp của các byte. Bằng cách này, BPE có thể giảm kích thước của từ vựng mà không làm mất thông tin quan trọng, giúp mô hình dễ dàng học từ các mẫu dữ liệu phức tạp.

SentencePiece làm đơn giản hóa quy trình tokenization hơn nữa bằng cách áp dụng trực tiếp lên chuỗi byte, không qua bước chuyển đổi sang ký tự. Phương pháp này giúp SentencePiece không bị ảnh hưởng bởi sự đa dạng ngôn ngữ, cho phép nó hỗ trợ tốt cho việc xử lý ngôn ngữ tự nhiên trên quy mô lớn và đa ngôn ngữ.

WordPiece, tương tự như BPE, nhưng được tinh chỉnh để hoạt động tốt hơn với các từ vựng có kích thước lớn. Điều này làm cho WordPiece trở nên lý tưởng cho các ứng dụng như dịch máy, nơi bộ từ vựng lớn và đa dạng là yếu tố quan trọng. WordPiece giúp mô hình dịch máy không chỉ hiểu mà còn tái tạo được ngôn ngữ mục tiêu một cách chính xác, qua đó cải thiện chất lượng dịch.

### ***2.5.3 Language Tags***

Language Tags (Wu et al. 2021) là một phương pháp hiệu quả giúp nâng cao hiệu suất của các mô hình dịch máy (NMT). Việc áp dụng Language Tags, hay còn được gọi là việc gắn thẻ ngôn ngữ, là kỹ thuật đơn giản mà mang lại hiệu quả đáng kể. Cụ thể, một thẻ đặc biệt chỉ rõ ngôn ngữ nguồn hoặc ngôn ngữ đích sẽ được thêm vào đầu hoặc cuối của câu đầu vào trong quá trình training và prediction của mô hình NMT. Việc này giúp mô hình dễ dàng nhận diện và phân biệt ngôn ngữ đang được xử lý, từ đó tối ưu hóa quá trình dịch thuật. Kỹ thuật này không chỉ cải thiện chất

lượng dịch bằng cách cung cấp thông tin ngữ cảnh rõ ràng hơn cho mô hình mà còn giúp giảm thiểu những nhầm lẫn giữa các ngôn ngữ có cấu trúc tương tự. Nhờ đó, Language Tags trở thành một công cụ quý giá trong việc phát triển và cải tiến mô hình NMT, giúp chúng trở nên linh hoạt và chính xác hơn.

Ví dụ: cho một câu tiếng Anh "Hello, how are you?", chúng tôi có thể thêm thẻ "<en>" để chỉ rõ đây là một câu tiếng Anh.

#### ***2.5.4 Encoding with padding***

Là quá trình chuẩn bị dữ liệu văn bản để có thể được mô hình học sâu xử lý một cách hiệu quả. Dưới đây là một giải thích chi tiết hơn về quá trình này:

##### **Tokenization và Vectorization**

Vectorization: Mỗi token sau đó được ánh xạ sang một vector số thông qua quá trình vectorization. Có thể sử dụng các phương pháp như one-hot encoding hoặc word embeddings để biểu diễn mỗi token dưới dạng vector.

Tokenization là việc chia nhỏ văn bản thành các đơn vị nhỏ hơn gọi là tokens, có thể là từ, cụm từ, ký tự hoặc subwords. Bước này giúp đơn giản hóa văn bản và chuẩn bị cho việc xử lý tiếp theo.

Vectorization chuyển đổi các tokens này thành vectors số, làm cho chúng có thể được xử lý bởi mô hình học sâu. Hai phương pháp phổ biến là one-hot encoding, nơi mỗi token được biểu diễn bởi một vector với chỉ một giá trị 1 và phần còn lại là 0, word embeddings một kỹ thuật nâng cao hơn biểu diễn mỗi token bằng một vector dày đặc, mang thông tin ngữ nghĩa và cú pháp.

Trong khi one-hot encoding cung cấp một biểu diễn đơn giản nhưng ít hiệu quả về không gian và không nắm bắt được mối quan hệ giữa các từ, word embeddings cho phép biểu diễn phong phú hơn, giúp mô hình hiểu được ngữ cảnh và ngữ nghĩa của từ, từ đó cải thiện hiệu suất của mô hình học sâu.

##### **Padding**

Khi làm việc với các mô hình học sâu, thường gặp phải vấn đề là các câu trong văn bản có độ dài không đồng nhất. Tuy nhiên, để mô hình có thể xử lý dữ liệu một cách hiệu quả, đầu vào cần phải có kích thước đồng nhất. Để giải quyết vấn đề này,

kỹ thuật padding được áp dụng, nghĩa là thêm các giá trị đặc biệt (thường là 0) vào cuối hoặc đầu của các vector đại diện cho token, nhằm mục đích làm cho tất cả chuỗi đạt được độ dài nhất định và đồng nhất.

Việc áp dụng padding không chỉ giúp quá trình xử lý dữ liệu theo batch trở nên thuận tiện và hiệu quả hơn mà còn đảm bảo rằng mô hình có thể được huấn luyện một cách mượt mà với dữ liệu có cấu trúc đồng nhất. Đồng thời, để mô hình có thể phân biệt được thông tin thực từ phần padding không chứa thông tin, kỹ thuật masking thường được sử dụng song song. Masking cho phép mô hình bỏ qua phần padding, tập trung vào xử lý nội dung thực sự có ý nghĩa, từ đó cải thiện đáng kể hiệu suất và chất lượng của quá trình học.

### **Chuyển đổi thành Tensor**

Sau khi tất cả các chuỗi đã được đồng nhất về độ dài thông qua padding, chúng được chuyển đổi thành tensors - dạng dữ liệu cơ bản được sử dụng trong các thư viện học sâu như TensorFlow hay PyTorch. Mỗi tensor chứa dữ liệu đầu vào được chuẩn bị sẵn sàng cho việc đào tạo hoặc dự đoán bởi mô hình NMT.

### **2.5.5 Optimizer**

Trong học máy và xử lý ngôn ngữ tự nhiên (NLP), chức năng của một optimizer là rất quan trọng. Optimizer có nhiệm vụ tự động cập nhật và điều chỉnh các tham số của mô hình dựa trên gradient của hàm mất mát (loss function). Điều này giúp mô hình hướng tới việc tìm kiếm bộ tham số tối ưu giúp giảm thiểu hàm mất mát, qua đó cải thiện hiệu suất làm việc.

Một phần quan trọng của optimizer là điều chỉnh tốc độ học (learning rate), đây là tham số kiểm soát tốc độ mà các tham số của mô hình được cập nhật. Quá trình này cần được thực hiện một cách cẩn thận để đảm bảo sự ổn định và hiệu quả của quá trình tối ưu hóa. Nếu tốc độ học quá cao, mô hình có thể bỏ qua điểm tối ưu; ngược lại, nếu tốc độ học quá thấp, quá trình tìm kiếm bộ tham số tối ưu sẽ trở nên chậm chạp, dẫn đến lãng phí thời gian và tài nguyên.

Nhìn chung, optimizer là một công cụ không thể thiếu trong việc tối ưu hóa các mô hình học sâu, giúp chúng ta nhanh chóng tìm ra bộ tham số tối ưu, từ đó đạt được hiệu suất tốt nhất trong các tác vụ cụ thể.

### **Các loại Optimizer phổ biến:**

Momentum (Duda 2019) là một kỹ thuật tăng tốc cho Stochastic Gradient Descent (SGD) (Turinici 2023), giúp cải thiện quá trình tối ưu hóa bằng cách tích lũy gradient từ các bước cập nhật trước đó. Qua đó, Momentum xác định hướng đi cho cập nhật hiện tại, tạo đà cho quá trình học, giống như việc đẩy một vật lên dốc và cho nó tiếp tục di chuyển bằng động lượng của chính nó. Điều này không chỉ giúp mô hình nhanh chóng vượt qua các điểm local minima (Kawaguchi 2021) mà còn giúp giảm tốc độ cập nhật khi tiếp cận global minimum

RMSprop (Ruder 2017) là một phát triển từ thuật toán Adagrad (Chakrabarti and Chopra 2021), nhằm khắc phục điểm yếu của việc giảm tốc độ học (learning rate) quá nhanh. RMSprop đạt được điều này thông qua việc áp dụng moving average của bình phương gradients, một cách tiếp cận giúp làm cho tốc độ học được điều chỉnh một cách linh hoạt và hiệu quả hơn giúp ngăn chặn tốc độ học giảm xuống quá thấp.

Adam (Adaptive Moment Estimation) (Kingma and Ba 2017) là một thuật toán tối ưu hóa tiên tiến, kết hợp hai ý tưởng chính từ Momentum và RMSprop để cải thiện quá trình training của các mô hình học máy. Bằng cách tính toán moving average của gradients và bình phương gradients, Adam không chỉ giữ được hướng và tốc độ của quá trình tối ưu mà còn điều chỉnh tốc độ học một cách linh hoạt cho mỗi tham số, giúp quá trình học trở nên hiệu quả và nhanh chóng hơn. Một trong những ưu điểm nổi bật của Adam là khả năng hội tụ nhanh hơn so với cả Momentum và RMSprop, nhờ vào cơ chế điều chỉnh tốc độ học tinh vi dựa trên gradient. Adam trở thành một lựa chọn dễ dàng sử dụng cho nhiều bài toán khác nhau. Ngoài ra, Adam còn có khả năng thích nghi tốt với các gradient có độ lớn biến thiên và hiệu quả trong việc xử lý dữ liệu thừa thớt, làm cho nó trở thành một công cụ tối ưu hóa mạnh mẽ và linh hoạt cho nhiều tác vụ NLP và học máy khác.

## CHƯƠNG 3. THỰC NGHIỆM

### 3.1 Giới thiệu tập dữ liệu

#### 3.1.1 Tập dữ liệu tiếng Việt – tiếng Tây Ban Nha

Dữ liệu dưới được lấy từ OPUS, dữ liệu bao gồm 325,542 cặp câu Việt – Tây Ban Nha được chia thành tập dữ liệu Bảng 3.1 như sau:

Bảng 3.1 Bảng thống kê dữ liệu Việt – Tây Ban Nha

Tập dữ liệu	Số lượng cặp câu
<b>Train</b>	325,542 cặp câu
<b>Test</b>	2000 cặp câu
<b>Validation</b>	2000 cặp câu

Hình 3.1, Hình 3.2 và Hình 3.3 bên dưới thể hiện dữ liệu tập của dữ liệu song ngữ giữa tiếng Việt và tiếng Tây Ban Nha.

	vi	es
211307	Một nước có thể đóng cửa biên giới, nhưng chắc...	Una nación podría cerrar sus puertas pero eso ...
310824	Gần đây tôi đã thử làm điều này và bản vẽ ngân...	Intenté este ejercicio hace poco, hablando sob...
96424	Một trong số họ nói với tôi, "Ron, chúng tôi s...	Como uno de ellos me dijo: "Ron, no iremos a n...
159617	Thế là, họ đã chọn những nhà chuyên môn dám th...	Así, eligieron a profesionales que confesaron ...
295023	Vậy, hay cố gắng, và hy vọng bạn có 1 chuyến d...	Entonces, persigamos el objetivo, les deseo un...
...	...	...
187451	Chúng tôi đã phân tích đoạn ghi âm đoạn nói ch...	Analizamos la voz grabada de 34 jóvenes con al...
189978	Đương nhiên.	Por supuesto.
42989	Chúng tôi cũng sử dụng những thứ khá thú vị-- ...	También estamos usando algo bastante interesan...
186422	Trong một trò chơi, chúng tôi yêu cầu trẻ đoán...	En este juego pedimos a los niños que adivinen...
83946	Bây giờ điều gì sẽ xảy ra nếu các so sánh hai ...	Ahora bien, ¿qué sucede cuando comparamos esta...

321542 rows × 2 columns

Hình 3.1 Hình ảnh dữ liệu tập train tiếng Việt và tiếng Tây Ban Nha



	vi	es
314583	Thực ra,	De hecho,
262774	Và chúng tôi nhận ra rằng có một khoảng cách l...	Y nos dimos cuenta del gran abismo que existía...
93096	Thỏa thuận lớn nhất đó là chúng ta nên tham vộ...	El mayor acuerdo fue que deberíamos ser ambici...
282879	Và khi không được làm người có ích, họ sẽ sớm ...	Y cuando a la gente no se le permite ser útil,...
92547	Chuyện bắt đầu từ lúc Jonny còn bé, khi em ấy ...	Todo comenzó cuando Jonny era pequeño, y empez...
...	...	...
98796	Thêm một ít nữa.	Agrego un poco más.
67018	Nhưng ai biết được điều này? Giờ bạn hãy cố tư...	Pero ¿quién sabe eso? Imagínense el sentimient...
237132	Có những origami khác trong không gian.	Hay otro objeto de origami en el espacio.
197857	Hãy chọn cách nhìn thấu chúng.	Opta por ver a través de ellos.
266089	Chưa hết, y học ngày nay vẫn tiếp tục khái niệ...	Y, sin embargo, la medicina hoy continúa conce...

2000 rows × 2 columns

Hình 3.2 Hình ảnh dữ liệu tập test tiếng Việt và tiếng Tây Ban Nha

	vi	es
314583	Thực ra,	De hecho,
262774	Và chúng tôi nhận ra rằng có một khoảng cách l...	Y nos dimos cuenta del gran abismo que existía...
93096	Thỏa thuận lớn nhất đó là chúng ta nên tham vộ...	El mayor acuerdo fue que deberíamos ser ambici...
282879	Và khi không được làm người có ích, họ sẽ sớm ...	Y cuando a la gente no se le permite ser útil,...
92547	Chuyện bắt đầu từ lúc Jonny còn bé, khi em ấy ...	Todo comenzó cuando Jonny era pequeño, y empez...
...	...	...
98796	Thêm một ít nữa.	Agrego un poco más.
67018	Nhưng ai biết được điều này? Giờ bạn hãy cố tư...	Pero ¿quién sabe eso? Imagínense el sentimient...
237132	Có những origami khác trong không gian.	Hay otro objeto de origami en el espacio.
197857	Hãy chọn cách nhìn thấu chúng.	Opta por ver a través de ellos.
266089	Chưa hết, y học ngày nay vẫn tiếp tục khái niệ...	Y, sin embargo, la medicina hoy continúa conce...

2000 rows × 2 columns

Hình 3.3 Hình ảnh dữ liệu tập validation tiếng Việt và tiếng Tây Ban Nha

### 3.1.2 Tập dữ liệu tiếng Anh – tiếng Việt

Dữ liệu dưới được lấy từ dataset “opus100” trên hugging face, dữ liệu bao gồm 1,004,000 cặp câu Anh - Việt được chia thành tập dữ liệu Bảng 3.2:

Bảng 3.2 Bảng thống kê dữ liệu tiếng Anh – tiếng Việt

Tập dữ liệu	Số lượng cặp câu
<b>Train</b>	1,000,000 cặp câu
<b>Test</b>	2000 cặp câu
<b>Validation</b>	2000 cặp câu

Hình 3.4, Hình 3.5 và Hình 3.6 bên dưới thể hiện dữ liệu tập của dữ liệu song ngữ giữa tiếng Anh và tiếng Việt.

	en	vi
0	What is it?	Cái gì đó?
1	I thought we would go to the children's home.	Con nghĩ chúng ta nên đến mái ấm.
2	Is there something you want to tell your husband?	Có điều gì cô muốn nói với chồng mình không?
3	Your master wants to hunt us, burn us, eat our...	Thầy của người muốn săn chúng ta, thiêu chúng ...
4	Or too weak to see this through?	Hay là kẻ yếu đuối?
...	...	...
999995	Not really.	Không hoàn toàn.
999996	What would he be selling at three o'clock in t...	Thì ông là một người bán hàng mà. Ông bán gì v...
999997	- Here she is.	- Nó đây rồi.
999998	Yeah, just take your shirts off.	Phải, cởi áo ra.
999999	Your new boyfriend is clearly insane.	Rõ là tên bạn trai mới của người bị điên.

Hình 3.4 Hình ảnh dữ liệu tập train tiếng Anh và tiếng Việt

	en	vi
0	We are in a dive.	Chúng ta đang lao xuống.
1	She brought us the job in the first place.	Anh ngủ với cô ta chưa Teddy?
2	- No.	- Không.
3	Beautiful country.	Một đất nước đẹp tuyệt.
4	Go save Riley.	Đi cứu Riley đi.
...	...	...
1995	There wasn't a good chance to greet you at you...	Hôm đám cưới tôi không có cơ hội chúc mừng cô.
1996	- Don't. - Maybe I should try humpback.	- Có lẽ tôi nên thử tiếng cá voi lưng gù.
1997	Oh, I think you'll recognize it.	Anh nghĩ em sẽ nhận ra đấy.
1998	Computer code?	Mã máy tính?
1999	Get out of my village	Ra khỏi làng của tôi.

2000 rows × 2 columns

Hình 3.5 Hình ảnh dữ liệu tập test tiếng Anh và tiếng Việt

	en	vi
0	You can act as him, too?	Anh cũng làm việc cho hắn ta?
1	I'm sorry. I am nervous today. I had bad dreams.	Xin lỗi, hôm nay tôi thấy khó chịu Tối qua tôi...
2	I wouldn't give her that pleasure. It's up to ...	Em không cho mẹ vinh hạnh đó đâu.
3	- Leave that in this bag.	- Bỏ nó vào túi.
4	Well, this is a domestic investigation.	Đây là việc điều tra nội địa.
...	...	...
1995	There's no way a career waitress comes to work...	Một phục vụ bàn chuyên nghiệp không bao giờ đi...
1996	You know, I always thought it'd get fixed at t...	Con biết không, bố luôn nghĩ sẽ sửa sai được ở...
1997	Fifteen thousand units!	15 ngàn!
1998	Why?	Tại sao?
1999	So what's new with you? Not much.	- Thế có gì mới với cậu không?

2000 rows × 2 columns

Hình 3.6 Hình ảnh dữ liệu tập validation tiếng Anh và tiếng Việt

### 3.1.3 Tập dữ liệu tiếng Anh – tiếng Tây Ban Nha

Dữ liệu dưới được lấy từ Dataset “opus100” trên hugging face, dữ liệu bao gồm 1,004,000 cặp câu Anh - Tây Ban Nha được chia thành tập dữ liệu Bảng 3.3 sau:

Bảng 3.3 Bảng thống kê dữ liệu tiếng Anh – tiếng Tây Ban Nha

Tập dữ liệu	Số lượng cặp câu
<b>Train</b>	1,000,000 cặp câu
<b>Test</b>	2000 cặp câu
<b>Validation</b>	2000 cặp câu

Hình 3.7, Hình 3.8 và Hình 3.9 bên dưới thể hiện dữ liệu tập của dữ liệu song ngữ giữa tiếng Anh và tiếng Tây Ban Nha.

	en	es
0	It was the asbestos in here, that's what did it!	Fueron los asbestos aquí. ¡Eso es lo que ocurrió!
1	I'm out of here.	Me voy de aquí.
2	One time, I swear I pooped out a stick of chalk.	Una vez, juro que cagué una barra de tiza.
3	And I will move, do you understand me?	Y prefiero mudarme, ¿Entiendes?
4	- Thank you, my lord.	- Gracias.
...	...	...
999995	He's not supposed to be here, so... there must...	El no debería estar aquí, así que... Debe habe...
999996	But Loreen is.	Pero Loreen.
999997	Charles O'Brien?	- ¿Charles O'Brien?
999998	It's... it's great.	Es genial.
999999	In contact with Priscilla Midori and Víctor Ma...	Priscilla Midori y Marcello Víctor, los cerebr...

1000000 rows × 2 columns

Hình 3.7 Hình ảnh dữ liệu tập train tiếng Anh và tiếng Tây Ban Nha

	en	es
0	If your country produced ODS for this purpose,...	Si su país produjo SAO para estos usos, sírvase...
1	♪ Juvie the great man, who else could it be bu...	# Juvie el gran hombre, ¿quién podría ser sino...
2	The home planet is running out.	El planeta madre se está agotando.
3	Oon't girls ever kill their mothers?	Las chicas no matan a sus madres? .
4	Humanitarian, recovery and development activities	Actividades humanitarias, de recuperación y de...
...	...	...
1995	Megan! I need to talk to you.	Megan, necesito hablar contigo.
1996	- Now that I have your attention, imagine we a...	- Le veo interesado, añádale otro cero al precio.
1997	Do not be concerned, these tips may help you p...	No se preocupe, estos consejos pueden ayudarle...
1998	She started slipping last year.	Empezó a irse el año pasado.
1999	For if many died through one man's falling awa...	15Mas no como el delito, tal fué el don: porqu...

2000 rows × 2 columns

Hình 3.8 Hình ảnh dữ liệu tập test tiếng Anh và tiếng Tây Ban Nha

	en	es
0	I don't even remember what the fight was about.	No recuerdo por qué fue la pelea.
1	Here are the sites of each of those that have ...	Estos son los sitios en que cada Congreso ha t...
2	I'm the man who killed Blackbeard.	Sí. Soy el hombre que mató a Barbanegra.
3	Don't get smart.	No te hagas el inteligente.
4	Is there an exact moment in the life of a sold...	¿Existe un límite de cuándo se padece y cuándo...
...	...	...
1995	[Scoffs] I believe the script says,	Me parece que el guión dice:
1996	You didn't even have a case against him.	Ni siquiera tenían un caso en su contra.
1997	Ok. She's dead.	Como lo desees, cariño.
1998	Opinion of Advocate General Léger delivered on...	Conclusiones del Abogado General Sr. P. Léger,...
1999	Prepare, yourselves.	Preparense.

2000 rows × 2 columns

Hình 3.9 Hình ảnh dữ liệu tập validation tiếng Anh và tiếng Tây Ban Nha



## 3.2 Xử lý dữ liệu và cài đặt chạy thực nghiệm

### 3.2.1 Xử lý dữ liệu

Dữ liệu sẽ được loại bỏ ký tự đặc biệt. Cụ thể đối với :

+ Tiếng Anh: tìm kiếm và loại bỏ các ký tự không có trong bảng chữ cái tiếng Anh, khoảng trắng thừa.

+ Tiếng Việt: tìm kiếm và loại bỏ các ký tự không có trong bảng chữ cái tiếng Việt, loại bỏ khoảng trắng thừa.

+ Tiếng Tây Ban Nha: tìm kiếm và loại bỏ các ký tự không có trong bảng chữ cái tiếng Tây Ban Nha, khoảng trắng thừa.

Sau đó chuyển tất cả dữ liệu sẽ được chuyển sang dạng chữ in thường. Thêm tag ngôn ngữ đích vào cột input của dữ liệu.

Sau đó sử dụng tokenizer từ mT5 chuyển dữ liệu sang dạng tensor với cùng một độ dài.

### 3.2.2 Cài đặt và chạy thực nghiệm

#### Đối với mô hình song ngữ mT5

Chúng tôi khởi tạo cấu hình mô hình mT5

Tăng kích thước của embedding tokenizer để phù hợp với số lượng token sau khi thêm các token ngôn ngữ đặc biệt.

Tạo DataLoader để dữ liệu được chia thành các batch nhỏ để xử lý dễ dàng và hiệu quả trong quá trình huấn luyện với batch size là 15

Sử dụng AdamW làm optimizer với learning rate =  $5e-5$ .

Mỗi epoch trong quá trình huấn luyện bao gồm việc tính toán loss, thực hiện backpropagation, và cập nhật trọng số mô hình.

Mô hình và trạng thái optimizer được lưu lại tại các điểm kiểm tra nhất định trong quá trình huấn luyện, cho phép việc tiếp tục huấn luyện từ điểm dừng hoặc thực hiện đánh giá sau này tham số và giá trị trong Bảng 3.4:

Bảng 3.4 Các tham số huấn luyện của mô hình song ngữ mT5

Tham số	Giá trị
Num layer	8
Hidden size	512
Batch size	15
Num head	6
Optimizer	adam
Train steps	634.275

### Đối với mô hình đa ngữ mT5

Chúng tôi khởi tạo cấu hình mô hình mT5

Tăng kích thước của embedding tokenizer để phù hợp với số lượng token sau khi thêm các token ngôn ngữ đặc biệt.

Tạo DataLoader để dữ liệu được chia thành các batch nhỏ để xử lý dễ dàng và hiệu quả trong quá trình huấn luyện với batch size là 15.

Sử dụng Adam  $W$  làm optimizer với learning rate =  $5e-5$ .

Mô hình và trạng thái optimizer được lưu lại tại các điểm kiểm tra nhất định trong quá trình huấn luyện, cho phép việc tiếp tục huấn luyện từ điểm dừng hoặc thực hiện đánh giá sau này. Tham số và giá trị trong Bảng 3.5:

Với mỗi epoch, tiến hành huấn luyện mô hình trên tập huấn luyện bằng cách sử dụng dữ liệu đầu vào và target, tính toán loss và cập nhật trọng số mô hình. Sau đó lưu trạng thái của mô hình của mô hình sau mỗi epoch. Sau đó tính BLEU để đánh giá chất lượng bản dịch của mô hình trên tập dữ liệu kiểm tra.

Bảng 3.5 Các tham số huấn luyện của mô hình đa ngữ mT5

Tham số	Giá trị
<b>Num layer</b>	8
<b>Hidden size</b>	512
<b>Batch size</b>	15
<b>Num head</b>	6
<b>Optimizer</b>	adam
<b>Train steps</b>	4.089.360

### 3.3 Phân tích kết quả

Đây là kết quả trong 15 epoch đầu tiên :

- Mô hình song ngữ Bảng 3.6:

Bảng 3.6 Kết quả mô hình song ngữ

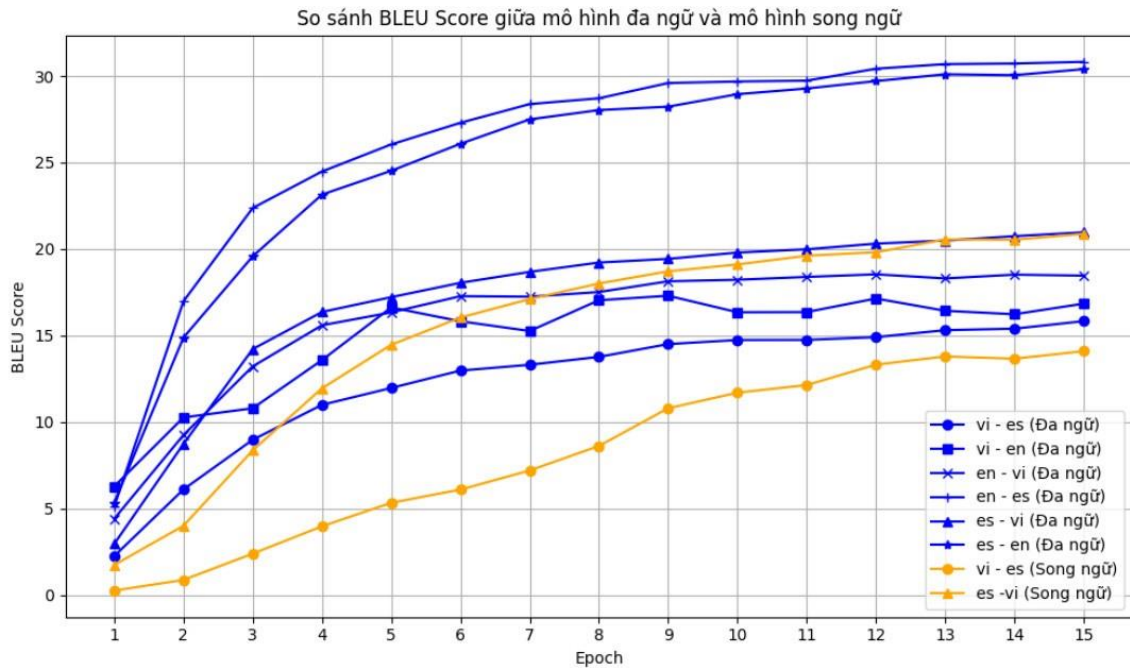
	<b>vi → es</b>	<b>es → vi</b>
<b>Mô hình song ngữ</b>	14.09	20.87

- Mô hình đa ngữ Bảng 3.7:

Bảng 3.7 Kết quả mô hình đa ngữ

	<b>vi → es</b>	<b>es → vi</b>	<b>vi → en</b>	<b>en → vi</b>	<b>es → en</b>	<b>en → es</b>
<b>Mô hình đa ngữ</b>	15.83	20.96	16.84	18.46	30.39	30.82





Hình 3.10 So sánh BLEU Score giữa mô hình đa ngữ và mô hình song ngữ

Dựa vào biểu đồ BLEU Score Hình 3.10 giữa mô hình đa ngữ và mô hình song ngữ, có thể thấy rằng mô hình đa ngữ thường đạt được điểm số BLEU cao hơn so với mô hình song ngữ qua hầu hết các epoch. Điều này cho thấy mô hình đa ngữ có khả năng cải thiện chất lượng dịch nhiều hơn so với mô hình song ngữ. Sự cải thiện này có thể do mô hình đa ngữ học được một không gian biểu diễn ngôn ngữ phong phú hơn, giúp nó có khả năng hiểu và chuyển đổi ngữ cảnh giữa các ngôn ngữ một cách linh hoạt hơn.

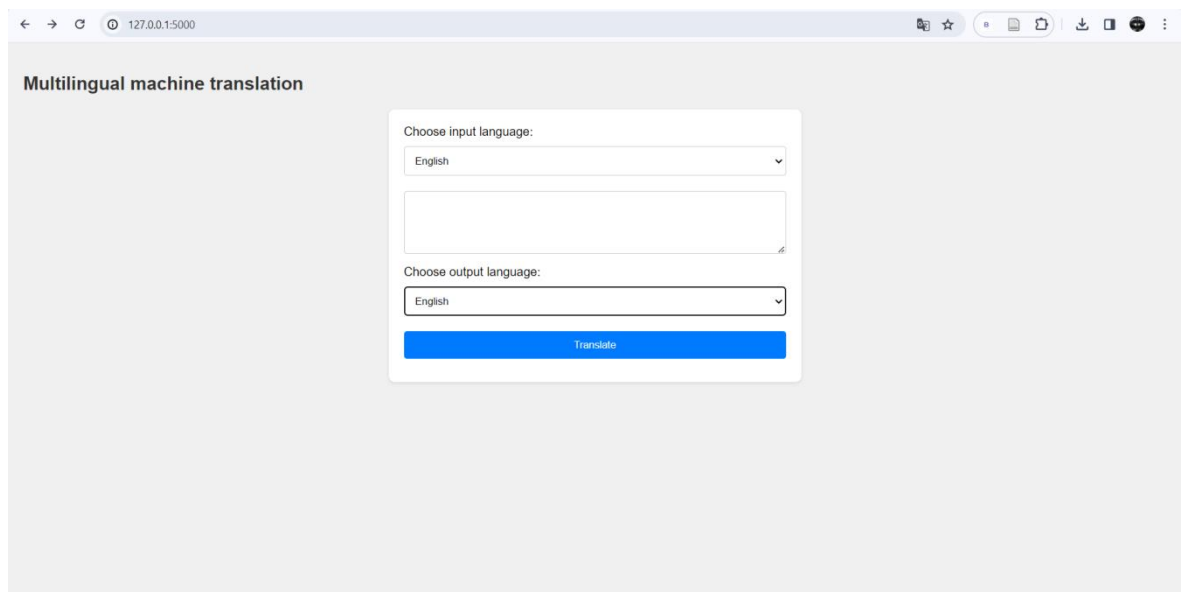
Tuy nhiên, về sau sự chênh lệch BLEU Score giữa mô hình đa ngữ và song ngữ không quá lớn, điều này có thể do nhiều yếu tố như đặc điểm của cặp ngôn ngữ được dịch, chất lượng và số lượng dữ liệu tiền huấn luyện. Mặc dù vậy, mô hình đa ngữ vẫn thể hiện ưu thế về khả năng tổng quát hóa và hiệu quả trong việc xử lý đa dạng tác vụ dịch ngôn ngữ.

Nhìn chung, kết quả cho thấy mô hình đa ngữ là một lựa chọn tốt hơn cho các ứng dụng dịch máy, đặc biệt trong trường hợp cần xử lý nhiều ngôn ngữ và có yêu cầu cao về chất lượng dịch. Sự linh hoạt và khả năng hiểu biết sâu rộng về ngôn ngữ

của mô hình đa ngữ làm cho nó trở thành một công cụ mạnh mẽ trong lĩnh vực xử lý ngôn ngữ tự nhiên.

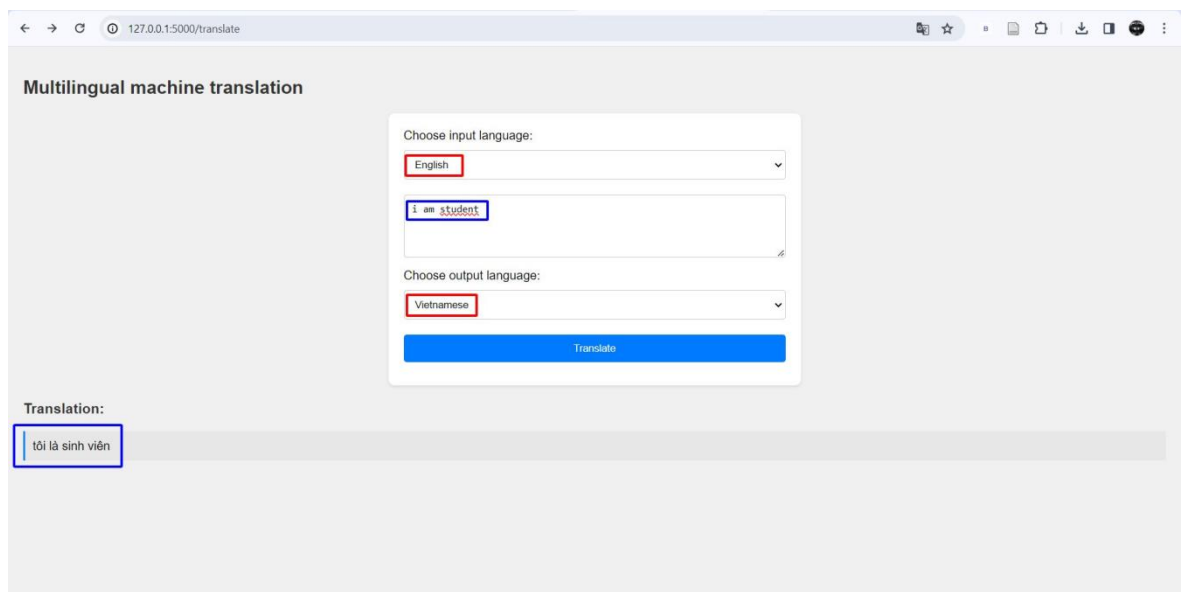
### 3.4 Demo website dịch máy đa ngôn ngữ

Hình 3.11 là giao diện trang chủ của website do thư viện Flask xây dựng.



Hình 3.11 Giao diện trang chủ

Khi muốn dịch từ tiếng Anh sang tiếng Việt ta chọn ngôn ngữ đầu vào là English và ngôn ngữ đầu ra là Vietnamese như Hình 3.12:



Hình 3.12 Hình ảnh demo hướng dẫn dịch từ tiếng Anh sang tiếng Việt

Khi muốn dịch từ tiếng Anh sang tiếng Tây Ban Nha ta chọn ngôn ngữ đầu vào là English và ngôn ngữ đầu ra là Spanish như Hình 3.13:

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/translate". The page title is "Multilingual machine translation". The interface includes two dropdown menus for language selection. The "Choose input language:" dropdown is set to "English". Below it, a text input field contains the text "I am student". The "Choose output language:" dropdown is set to "Spanish". A blue "Translate" button is positioned below the output language dropdown. In the "Translation:" section, the output text "estudiante de am" is displayed in a text box.

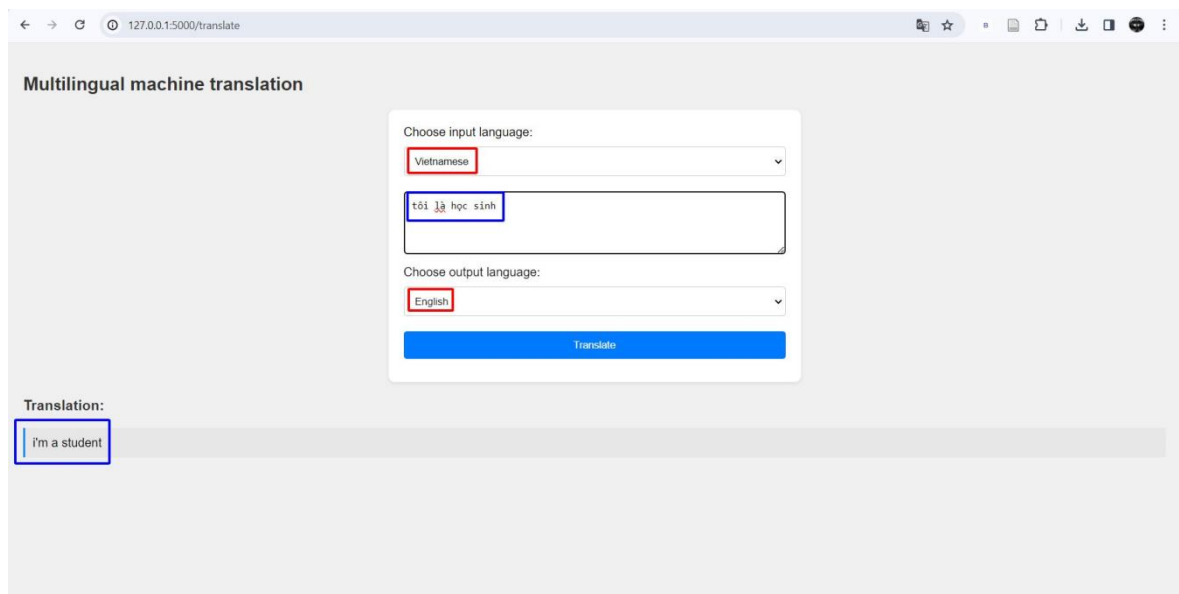
Hình 3.13 Hình ảnh demo hướng dẫn dịch từ tiếng Anh sang tiếng Tây Ban Nha

Khi muốn dịch từ tiếng Việt sang tiếng Tây Ban Nha ta chọn ngôn ngữ đầu vào là Vietnamese và ngôn ngữ đầu ra là Spanish như Hình 3.14:

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/translate". The page title is "Multilingual machine translation". The interface includes two dropdown menus for language selection. The "Choose input language:" dropdown is set to "Vietnamese". Below it, a text input field contains the text "Tôi là học sinh". The "Choose output language:" dropdown is set to "Spanish". A blue "Translate" button is positioned below the output language dropdown. In the "Translation:" section, the output text "soy estudiante" is displayed in a text box.

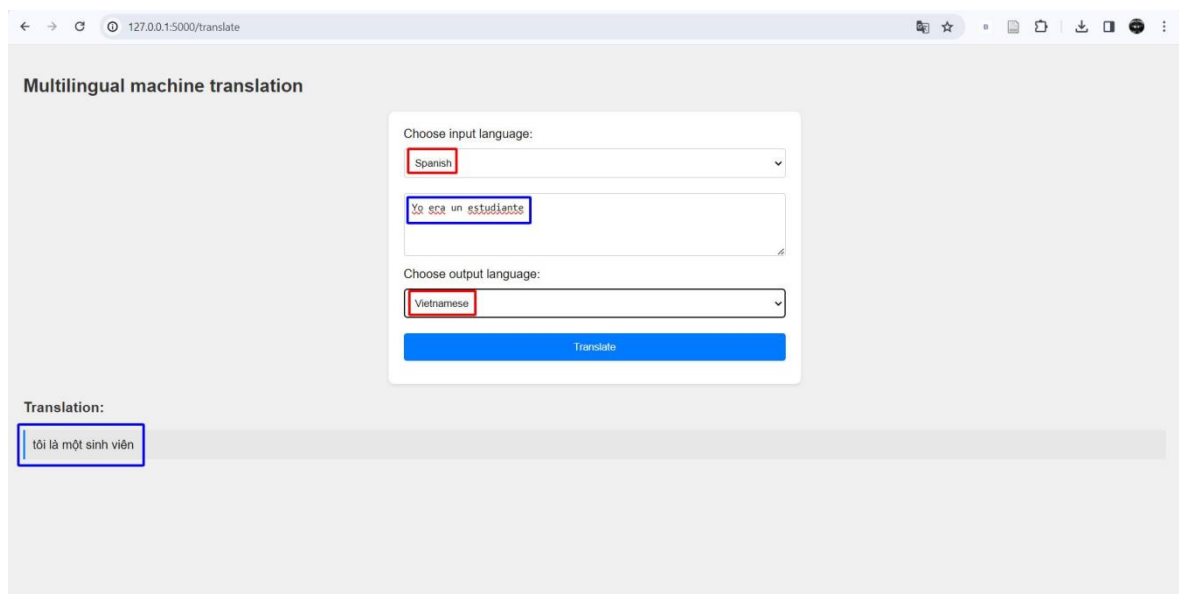
Hình 3.14 Hình ảnh demo hướng dẫn dịch từ tiếng Việt sang tiếng Tây Ban Nha

Khi muốn dịch từ tiếng Việt sang tiếng Anh ta chọn ngôn ngữ đầu vào là Vietnamese và ngôn ngữ đầu ra là English như Hình 3.15:



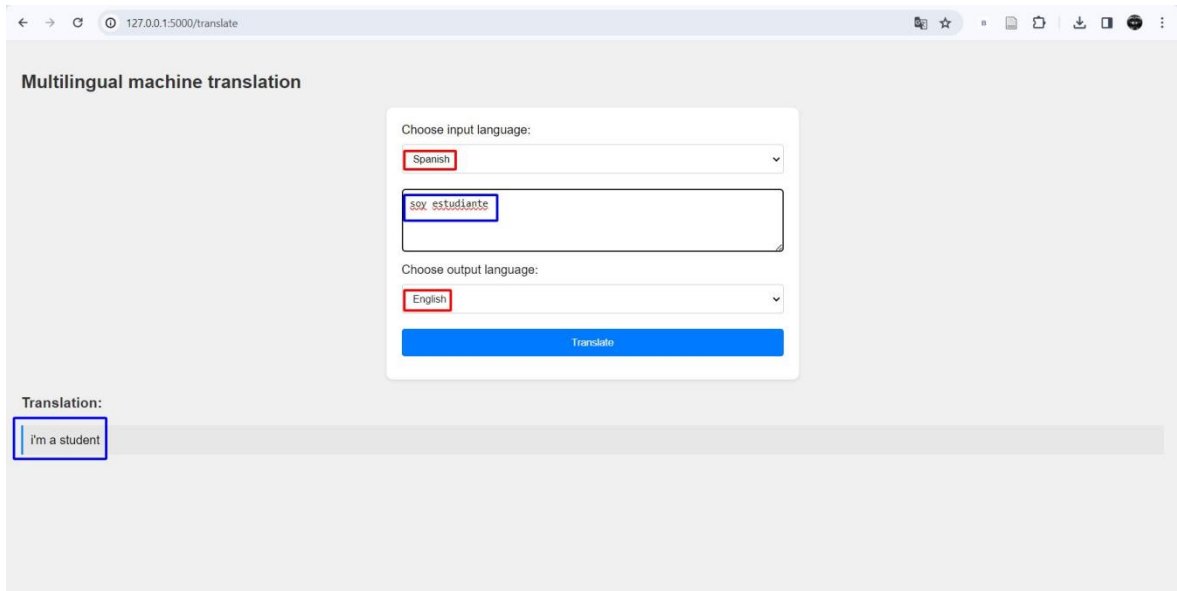
Hình 3.15 Hình ảnh demo hướng dẫn dịch từ tiếng Việt sang tiếng Anh

Khi muốn dịch từ tiếng Tây Ban Nha sang tiếng Việt ta chọn ngôn ngữ đầu vào là Spanish và ngôn ngữ đầu ra là Vietnamese như Hình 3.16:



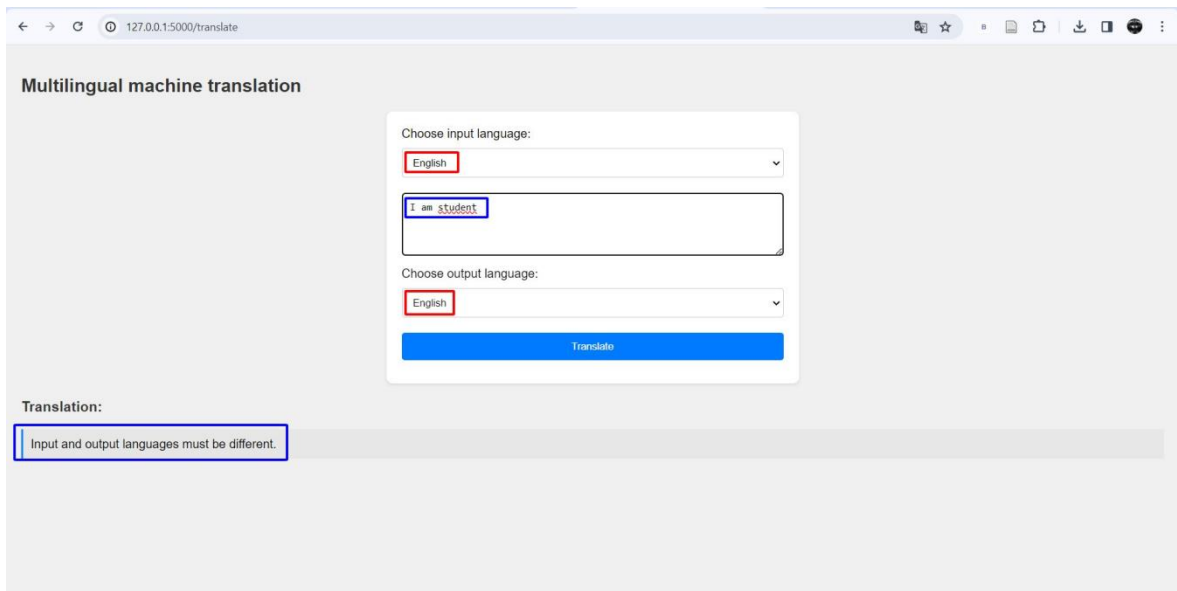
Hình 3.16 Hình ảnh demo hướng dẫn dịch từ tiếng Tây Ban Nha sang tiếng Việt

Khi muốn dịch từ tiếng Tây Ban Nha sang tiếng Anh ta chọn ngôn ngữ đầu vào là Spanish và ngôn ngữ đầu ra là English như Hình 3.17:



Hình 3.17 Hình ảnh demo hướng dẫn dịch từ tiếng Tây Ban Nha sang tiếng Anh

Trường hợp đặc biệt, khi ta chọn ngôn ngữ input và ngôn ngữ output trùng nhau thì website trả về kết quả là “Input and output languages must be different.” như Hình 3.18:



Hình 3.18 Hình ảnh demo trường hợp input trùng với output

## CHƯƠNG 4. KẾT LUẬN

### 4.1 Kết quả đạt được

#### **Đối với mô hình song ngữ:**

Chúng tôi thấy điểm BLEU từ việc dịch từ tiếng Việt sang tiếng Tây Ban Nha và ngược lại tăng dần qua mỗi epoch, cho thấy mô hình liên tục học hỏi và cải thiện khả năng dịch qua thời gian.

Mô hình tiếp thu kiến thức từ dữ liệu training hiệu quả. Khả năng dịch chính xác, nhanh hơn và cải thiện tốt.

#### **Đối với mô hình đa ngữ:**

Việc sử dụng mô hình đa ngữ có thể không đạt được sự cải thiện đột biến về chất lượng dịch so với mô hình song ngữ. Tuy nhiên mô hình đa ngữ vẫn cho thấy khả năng cải thiện dần dần qua các epoch tốt hơn của mô hình song ngữ, chứng tỏ khả năng áp dụng học hỏi của mô hình đa ngữ.

### 4.2 Hạn chế của phương pháp giải quyết bài toán

Mô hình có độ phức tạp tính toán cao, mô hình đa ngữ đòi hỏi nhiều tài nguyên tính toán hơn so với mô hình song ngữ. Gây ra tăng chi phí và thời gian đào tạo và tinh chỉnh mô hình.

Việc thiếu dữ liệu đào tạo chất lượng cao làm hạn chế khả năng học và hiệu suất của mô hình. Đặc biệt là với các ngôn ngữ có tài nguyên hạn chế.

### 4.3 Hướng phát triển trong tương lai

Tối ưu hóa mô hình đa ngữ: giúp giảm sự chênh lệch về hiệu suất của các cặp ngôn ngữ. Tập trung vào cải thiện hiệu suất cho các cặp này bằng cách chỉnh sửa lại các siêu tham số như là learning rate, kích thước batch, các siêu tham số khác.

Chúng tôi sẽ khám phá nhiều kiến trúc mô hình mới. Thử nghiệm với các biến thể của Transformer như BERT, GPT, Marian...

## TÀI LIỆU THAM KHẢO

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. “TensorFlow: A System for Large-Scale Machine Learning.” arXiv. <http://arxiv.org/abs/1605.08695>.

Agić, Željko, and Ivan Vulić. 2019. “JW300: A Wide-Coverage Parallel Corpus for Low-Resource Languages.” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, edited by Anna Korhonen, David Traum, and Lluís Màrquez, 3204–10. Florence, Italy: Association for Computational Linguistics. <https://doi.org/10.18653/v1/P19-1310>.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. “Layer Normalization.” arXiv. <http://arxiv.org/abs/1607.06450>.

Bahdanau, Dzmitry. 2016. “Neural Machine Translation by Jointly Learning to Align and Translate.” arXiv. <http://arxiv.org/abs/1409.0473>.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2016. “Neural Machine Translation by Jointly Learning to Align and Translate.” arXiv. <https://doi.org/10.48550/arXiv.1409.0473>.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. “Neural versus Phrase-Based Machine Translation Quality: A Case Study.” arXiv. <http://arxiv.org/abs/1608.04631>.

Berglund, Martin, and Brink van der Merwe. 2023. “Formalizing BPE Tokenization.” *Electronic Proceedings in Theoretical Computer Science* 388 (September): 16–27. <https://doi.org/10.4204/EPTCS.388.4>.

Bertoldi, Nicola, and Marcello Federico. 2009. “Domain Adaptation for Statistical Machine Translation with Monolingual Resources.” In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, edited by Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder, 182–89. Athens, Greece: Association for Computational Linguistics. <https://aclanthology.org/W09-0432>.

Britz, Denny, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. “Massive Exploration of Neural Machine Translation Architectures.” arXiv. <http://arxiv.org/abs/1703.03906>.

Cettolo, Mauro, Christian Girardi, and Marcello Federico. 2012. “WIT3: Web Inventory of Transcribed and Translated Talks.” In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, edited by Mauro Cettolo, Marcello Federico, Lucia Specia, and Andy Way, 261–68. Trento, Italy: European Association for Machine Translation. <https://aclanthology.org/2012.eamt-1.60>.

Chakrabarti, Kushal, and Nikhil Chopra. 2021. “Generalized AdaGrad (G-AdaGrad) and Adam: A State-Space Perspective.” arXiv. <http://arxiv.org/abs/2106.00092>.

Cho, Kyunghyun, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. “Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation.” arXiv. <http://arxiv.org/abs/1406.1078>.

Choi, Dami, Christopher J. Shallue, Zachary Nado, Jaehoon Lee, Chris J. Maddison, and George E. Dahl. 2020. “On Empirical Comparisons of Optimizers for Deep Learning.” arXiv. <http://arxiv.org/abs/1910.05446>.

Duda, Jarek. 2019. “SGD Momentum Optimizer with Step Estimation by Online Parabola Model.” arXiv. <http://arxiv.org/abs/1907.07063>.

Freitag, Markus, and Orhan Firat. 2020. “Complete Multilingual Neural Machine Translation.” arXiv. <http://arxiv.org/abs/2010.10239>.

Ghojogh, Benyamin, and Ali Ghodsi. 2023. “Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and Survey.” arXiv. <http://arxiv.org/abs/2304.11461>.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. “Deep Residual Learning for Image Recognition.” arXiv. <http://arxiv.org/abs/1512.03385>.



Khyani, Divya, and Siddhartha B S. 2021. “An Interpretation of Lemmatization and Stemming in Natural Language Processing.” *Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology* 22 (January): 350–57.

Kingma, Diederik P., and Jimmy Ba. 2017. “Adam: A Method for Stochastic Optimization.” arXiv. <http://arxiv.org/abs/1412.6980>.

Kudo, Taku, and John Richardson. 2018. “SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing.” arXiv. <http://arxiv.org/abs/1808.06226>.

Kudo, Taku, John Richardson, and Kudo and Richardson. 2018. “SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing.” arXiv. <http://arxiv.org/abs/1808.06226>.

Lakew, Surafel M., Matteo Negri, and Marco Turchi. 2020. “Low Resource Neural Machine Translation: A Benchmark for Five African Languages.” arXiv. <http://arxiv.org/abs/2003.14402>.

“N-Gram Language Models.Pdf.” n.d. Accessed March 13, 2024. <https://web.stanford.edu/~jurafsky/slp3/3.pdf>.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. “Bleu: A Method for Automatic Evaluation of Machine Translation.” In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, edited by Pierre Isabelle, Eugene Charniak, Dekang Lin, and Papineni, 311–18. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics. <https://doi.org/10.3115/1073083.1073135>.

Papineni, Kishore, Salim Roukos, Todd Ward, Wei-Jing Zhu, and Kishore Papineni. 2002. “Bleu: A Method for Automatic Evaluation of Machine Translation.” In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, edited by Pierre Isabelle, Eugene Charniak, and Dekang Lin, 311–18. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics. <https://doi.org/10.3115/1073083.1073135>.

Paschalidou, Despoina, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. 2021. “ATISS: Autoregressive Transformers for Indoor Scene Synthesis.” arXiv. <http://arxiv.org/abs/2110.03675>.

“PyTorch DataLoader: A Complete Guide • Datagy.” n.d. Accessed March 13, 2024. <https://datagy.io/pytorch-dataloader/>.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.” arXiv. <http://arxiv.org/abs/1910.10683>.

Ruder, Sebastian. 2017. “An Overview of Gradient Descent Optimization Algorithms.” arXiv. <http://arxiv.org/abs/1609.04747>.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. “Neural Machine Translation of Rare Words with Subword Units.” arXiv. <http://arxiv.org/abs/1508.07909>.

Song, Xinying, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. “Fast WordPiece Tokenization.” arXiv. <http://arxiv.org/abs/2012.15524>.

Tiedemann, Jorg. n.d. “Parallel Data, Tools and Interfaces in OPUS.”

Turinici, Gabrel. 2023. “The Convergence of the Stochastic Gradient Descent (SGD) : A Self-Contained Proof.” <https://doi.org/10.5281/zenodo.4638694>.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. “Attention Is All You Need.” arXiv. <http://arxiv.org/abs/1706.03762>.

Wu, Liwei, Shanbo Cheng, Mingxuan Wang, and Lei Li. 2021. “Language Tags Matter for Zero-Shot Neural Machine Translation.” arXiv. <http://arxiv.org/abs/2106.07930>.

Yang, Jinbiao. 2024. “Rethinking Tokenization: Crafting Better Tokenizers for Large Language Models.”

Zhang, Biao, Deyi Xiong, and Jinsong Su. 2019. “A GRU-Gated Attention Model for Neural Machine Translation.” arXiv. <http://arxiv.org/abs/1704.08430>.

Zhuang, Fuzhen, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. “A Comprehensive Survey on Transfer Learning.” arXiv. <http://arxiv.org/abs/1911.02685>.