**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\* \* \* ======**

****

**ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH KHOA HỌC MÁY TÍNH**

**ĐỀ TÀI: NGHIÊN CỨU ÁP DỤNG DEEP LEARNING TRONG BÀI TOÁN DỊCH MÁY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Giảng viên hướng dẫn** | | : | TS. Trần Hùng Cường |  |
| **Sinh viên** | | : | Nguyễn Lương Nam Anh |  |
| **Mã sinh viên** | | : | 2020605072 |  |
| **Lớp** | | : | KHMT02 |  |
| **Khóa** | | : | 15 |  |
|  | |  |  |  |
|  |

**Hà Nội - 2024**

LỜI CẢM ƠN

Để thực hiện và hoàn thành tốt báo cáo đồ án tốt nghiệp, em đã nhận được sự giúp đỡ và hướng dẫn rất tận tính của thầy giáo Trần Hùng Cường thuộc Khoa công nghệ thông tin, trường Đại học Công Nghiệp Hà Nội. Với tình cảm sâu sắc và chân thành, xin phép em được bày tỏ lòng biết ơn đến thầy.

Trong quá trình làm đồ án, do giới hạn về mặt kiến thức cũng như kinh nghiệm thực tiễn nên đề tài không tránh khỏi những thiếu sót. Em rất mong nhận được ý kiến đóng góp từ thầy để bài báo cáo được hoàn thiện hơn.

Em xin chúc thầy luôn dồi dào sức khỏe, vui vẻ và thành công trong cuộc sống.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Nguyễn Lương Nam Anh

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc166968063)

[DANH MỤC NHỮNG TỪ VIẾT TẮT 4](#_Toc166968064)

[DANH MỤC BẢNG BIỂU 5](#_Toc166968065)

[DANH MỤC HÌNH ẢNH 6](#_Toc166968066)

[LỜI NÓI ĐẦU 7](#_Toc166968067)

[CHƯƠNG 1: TỔNG QUAN VỀ BÀI TOÁN 9](#_Toc166968068)

[1.1 Tổng quan về xử lý ngôn ngữ tự nhiên 9](#_Toc166968069)

[1.1.1 Khái quát về xử lý ngôn ngữ tự nhiên 9](#_Toc166968070)

[1.1.2 Cách hoạt động của xử lý ngôn ngữ tự nhiên 10](#_Toc166968071)

[1.1.3 Các thành phần của xử lý ngôn ngữ tự nhiên 10](#_Toc166968072)

[1.1.4 Ưu và nhược điểm của xử lý ngôn ngữ tự nhiên 12](#_Toc166968073)

[1.2 Bài toán dịch máy qua văn bản 13](#_Toc166968074)

[1.2.1 Giới thiệu bài toán 13](#_Toc166968075)

[1.2.2 Lý do chọn đề tài 15](#_Toc166968076)

[1.2.3 Mô tả chi tiết đầu vào và đầu ra bài toán 16](#_Toc166968077)

[1.2.4 Các khó khăn và thách thức của bài toán 16](#_Toc166968078)

[1.2.5 Ứng dụng của bài toán 17](#_Toc166968079)

[CHƯƠNG 2: CÁC KỸ THUẬT GIẢI QUYẾT BÀI TOÁN 19](#_Toc166968080)

[2.1 Mô hình BART 19](#_Toc166968081)

[2.1.1 Lý do chọn mô hình 19](#_Toc166968082)

[2.1.2 Giới thiệu chung 19](#_Toc166968083)

[2.1.3 Chi tiết mô hình BART 20](#_Toc166968084)

[2.1.4 Xử lý đầu vào mô hình BART 26](#_Toc166968085)

[2.2 Tokenizer trong mô hình BART 28](#_Toc166968086)

[2.2.1 Giới thiệu chung 28](#_Toc166968087)

[2.2.2 Byte-Pair Encoding tokenization 29](#_Toc166968088)

[2.2.3 So sánh các kỹ thuật tokenizer 31](#_Toc166968089)

[2.3 Đơn vị đo của bài toán 32](#_Toc166968090)

[2.3.1 Các đơn vị đo phổ biến trong dịch máy 32](#_Toc166968091)

[2.3.2 Độ đo SacreBLEU 33](#_Toc166968092)

[2.4 Quy trình giải quyết bài toán 35](#_Toc166968093)

[2.4.1 Kỹ thuật Back-Translation 35](#_Toc166968094)

[2.4.2 Quy trình giải quyết 36](#_Toc166968095)

[CHƯƠNG 3: THỰC NGHIỆM 39](#_Toc166968096)

[3.1 Bộ dữ liệu thực nghiệm 39](#_Toc166968097)

[3.2 Huấn luyện các mô hình 41](#_Toc166968098)

[3.2.1 Mô hình mBART50 từ Anh sang Việt 43](#_Toc166968099)

[3.2.2 Mô hình mBART50 từ Việt sang Anh 47](#_Toc166968100)

[3.2.3 Back Translation từ Anh sang Việt 49](#_Toc166968101)

[3.3 Các kết quả thực nghiệm 52](#_Toc166968102)

[CHƯƠNG 4: Xây dựng chương trình demo 55](#_Toc166968103)

[4.1 Giới thiệu về các framework sử dụng 55](#_Toc166968104)

[4.1.1 Flask 55](#_Toc166968105)

[4.1.2 Transformers 56](#_Toc166968106)

[4.2 Phân tích hệ thống 57](#_Toc166968107)

[4.2.1 Biểu đồ use case tổng quát 57](#_Toc166968108)

[4.2.2 Mô tả chi tiết các use case 58](#_Toc166968109)

[4.3 Giao diện hệ thống 60](#_Toc166968110)

[4.4 Các chức năng của hệ thống 62](#_Toc166968111)

[4.4.1 Dịch văn bản từ tiếng Anh sang tiếng Việt 62](#_Toc166968112)

[4.4.2 Dịch văn bản từ tiếng Việt sang tiếng Anh 63](#_Toc166968113)

[KẾT LUẬN 65](#_Toc166968114)

[TÀI LIỆU THAM KHẢO 66](#_Toc166968115)

DANH MỤC NHỮNG TỪ VIẾT TẮT

|  |  |
| --- | --- |
| NLP | Natural Language Processing |
| AI | Artificial Intelligence |
| NMT | Neural Machine Translation |
| BART | Bidirectional Autoregressive Transformers |
| GPT | Generative Pre-trained Transformer |
| BERT | Bidirectional Encoder Representations from Transformers |
| MT | Machine Translation |
| BLEU | BiLingual Evaluation Understudy |
| BPE | Byte-Pair Encoding |
| BT | Back Translation |
| GPU | Graphics Processing Unit |
| HTTP | Hypertext Transfer Protocol |
| API | Application Programming Interface |

DANH MỤC BẢNG BIỂU

[Bảng 2.1 So sánh các kỹ thuật tokenizer 31](#_Toc166966835)

[Bảng 3.1 Dữ liệu thực nghiệm 41](#_Toc166966836)

[Bảng 3.2 Kết quả tốt nhất của các mô hình 53](#_Toc166966837)

[Bảng 3.3 Thời gian huấn luyện của các mô hình 54](#_Toc166966838)

DANH MỤC HÌNH ẢNH

[Hình 1.1 Minh họa về xử lý ngôn ngữ tự nhiên 9](#_Toc166878807)

[Hình 1.2 Minh họa về cách phân tích từ vựng 11](#_Toc166878808)

[Hình 1.3 Minh họa về dịch máy 14](#_Toc166878809)

[Hình 2.1 Minh họa về BART 20](#_Toc166878810)

[Hình 2.2 Minh họa kiến trúc mô hình Transformers 21](#_Toc166878811)

[Hình 2.3 Kiến trúc mô hình BART 21](#_Toc166878812)

[Hình 2.4 Khối Bidirectional Encoder 22](#_Toc166878813)

[Hình 2.5 Bộ mã hóa hai chiều của mô hình BART 23](#_Toc166878814)

[Hình 2.6 Khối Encoder 23](#_Toc166878815)

[Hình 2.7 Multi-Head Attention 24](#_Toc166878816)

[Hình 2.8 Bộ giải mã hồi quy của mô hình BART 26](#_Toc166878817)

[Hình 2.9 Các phép biến đổi xử lý đầu vào trong BART 27](#_Toc166878818)

[Hình 2.10 Pipeline phương pháp giải quyết bài toán 37](#_Toc166878819)

[Hình 3.1 Ví dụ về dataset IWSLT2015-en-vi 39](#_Toc166878820)

[Hình 3.2 Ví dụ về dataset PhoMT tiếng Anh 40](#_Toc166878821)

[Hình 3.3 Ví dụ về dataset PhoMT tiếng Việt 41](#_Toc166878822)

[Hình 3.4 Biểu đồ giá trị Bleu trong quá trình huấn luyện 52](#_Toc166878823)

[Hình 3.5 Biểu đồ giá trị Loss trong quá trình huấn luyện 53](#_Toc166878824)

[Hình 4.1 Mình họa thư viện Flask 55](#_Toc166878825)

[Hình 4.2 Minh họa thư viện Transformers 56](#_Toc166878826)

[Hình 4.3 Biểu đồ use case tổng quát 58](#_Toc166878827)

[Hình 4.4 Giao diện mặc định khi khởi động 61](#_Toc166878828)

[Hình 4.5 Giao diện khi thực hiện dịch sang tiếng Việt 61](#_Toc166878829)

[Hình 4.6 Giao diện khi thực hiện dịch sang tiếng Anh 62](#_Toc166878830)

[Hình 4.7 Giao diện khi thực hiện dịch sang tiếng Việt 63](#_Toc166878831)

[Hình 4.8 Giao diện khi thực hiện dịch sang tiếng Anh 64](#_Toc166878832)

LỜI NÓI ĐẦU

Trong một thời đại công nghệ ngày nay, việc nghiên cứu và phát triển các hệ thống ngôn ngữ tự nhiên đang trở thành một lĩnh vực quan trọng, đặc biệt là trong ngữ cảnh thực tế như dịch văn bản, phân loại văn bản, tóm tắt văn bản, tìm kiếm và trích xuất thông tin, và nhiều ứng dụng khác.

Tôi chọn đề tài dịch máy này vì mong muốn có thể dịch được qua lại các loại ngôn khác nhau trên thế giới hoặc các dân tộc trong nước.

Mục tiêu nghiên cứu của bài báo cáo này là dịch văn bản từ tiếng Anh sang tiếng Việt và ngược lại.

Trong bài báo cáo tôi sẽ sử dụng mô hình BART cùng với phương pháp Back Translation để tăng cường dữ liệu.

Phần chính của bài báo cáo sẽ đi sâu vào các khía cạnh quan trọng của lĩnh vực xử lý ngôn ngữ tự nhiên trong bài toán dịch máy. Chúng ta sẽ xem xét các phương pháp giải quyết bài toán, công cụ, và các ứng dụng cụ thể trong thực tế.

Trong báo cáo này, tôi quyết định trình bày thành 4 chương như sau:

**Chương 1: Tổng quan về bài toán**

Trong chương 1, tôi tiến hành nêu các kiến thức tổng quan về xử lý ngôn ngữ tự nhiên, nêu khái quát cơ bản về bài toán dịch máy.

**Chương 2: Các kỹ thuật giải quyết bài toán**

Tại chương 2, tôi trình bày các kỹ thuật giải quyết bài toán hiện nay và các ưu và nhược điểm của chúng, cũng như các nghiên cứu nổi bật đã đạt được thành công nhất định từ những kỹ thuật đó.

**Chương 3: Thực nghiệm**

Tại chương 3, tôi tập chung trình bày về quá trình thực nghiệm khi áp dụng phương pháp đã chọn vào bài toán.

**Chương 4: Xây dựng sản phẩm demo**

Để tận dụng kết quả thực nghiệm đã thu được, tôi tiến hành xây dựng sản phẩm demo là một trang web cho chức năng dịch văn bản.

**Phần kết luận:**

Phần cuối này, tôi sẽ tổng hợp các kết quả đạt được, các hướng phát triển đề tài nghiên cứu trong tương lai.

CHƯƠNG 1: TỔNG QUAN VỀ BÀI TOÁN

1.1 Tổng quan về xử lý ngôn ngữ tự nhiên

1.1.1 Khái quát về xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (NLP) là một lĩnh vực của trí tuệ nhân tạo (AI) tập trung vào việc nghiên cứu và ứng dụng các kỹ thuật và công cụ cho phép máy tính xử lý, phân tích, giải thích và suy luận về ngôn ngữ của con người. NLP là một lĩnh vực liên ngành và nó kết hợp các kỹ thuật được thiết lập trong các lĩnh vực như ngôn ngữ học và khoa học máy tính.



Hình 1.1 Minh họa về xử lý ngôn ngữ tự nhiên

***Lịch sử của NLP:***

Đây là những sự kiện quan trọng trong lịch sử Xử lý ngôn ngữ tự nhiên:

* + ***1950****:* NLP bắt đầu khi Alan Turing xuất bản một bài báo có tên “Máy móc và trí thông minh”. Trong bài báo này, ông đã đặt ra “Thử thách Turing”, một thử thách để xác định khả năng của máy móc trong việc mô phỏng trí thông minh con người.
  + ***1950****:* Nỗ lực tự động hóa bản dịch giữa tiếng Nga và tiếng Anh. Đây là một trong những ứng dụng đầu tiên của NLP, được gọi là Máy dịch tự động.
  + ***1960****:* Công trình của Chomsky và những người khác về lý thuyết ngôn ngữ hình thức và cú pháp đã tạo ra một nền tảng quan trọng cho NLP. Chomsky đã đề xuất mô hình “Chomsky Hierarchy”, một hệ thống phân loại ngôn ngữ dựa trên tính phức tạp của cấu trúc ngữ pháp.
  + ***1990****:* Các mô hình xác suất và dựa trên dữ liệu đã trở nên khá chuẩn. Điều này đã mở ra một kỷ nguyên mới trong NLP, khi mà các phương pháp dựa trên quy tắc được thay thế bởi các mô hình học máy. Các mô hình như Hidden Markov Models (HMMs) và n-grams đã trở nên phổ biến trong các tác vụ như phân loại văn bản và nhận dạng tiếng nói.
  + ***2000****:* Một lượng lớn dữ liệu nói và văn bản có sẵn. Sự phát triển của internet đã tạo ra một lượng lớn dữ liệu văn bản có sẵn, điều này đã thúc đẩy sự phát triển của các mô hình NLP dựa trên dữ liệu lớn. Điều này cũng đã tạo điều kiện cho sự phát triển của các mô hình ngôn ngữ dựa trên n-gram và các mô hình học sâu.

1.1.2 Cách hoạt động của xử lý ngôn ngữ tự nhiên

NLP sử dụng một loạt kỹ thuật và phương pháp để xử lý và phân tích ngôn ngữ tự nhiên. Quá trình này bao gồm một số bước sau đây:

* + Tokenization: Quá trình chia câu và văn bản thành các phần tử nhỏ hơn như từ, cụm từ hoặc câu.
  + Morphological analysis: Phân tích từ loại, hình thái và cấu trúc của từ để hiểu ý nghĩa của chúng.
  + Parsing: Phân tích câu để hiểu cấu trúc ngữ pháp và mối quan hệ giữa các thành phần trong câu.

1.1.3 Các thành phần của xử lý ngôn ngữ tự nhiên



Hình 1.2 Minh họa về cách phân tích từ vựng

Năm thành phần chính của xử lý ngôn ngữ tự nhiên trong AI là:

* + *Phân tích hình thái và từ vựng:*

Phân tích từ vựng là một từ vựng bao gồm các từ và cách diễn đạt của nó. Nó mô tả việc phân tích, xác định và mô tả cấu trúc của từ. Nó bao gồm việc chia văn bản thành các đoạn văn, từ và câu.

Các từ riêng lẻ được phân tích thành các thành phần của chúng và các mã thông báo không phải từ như dấu chấm câu được tách ra khỏi các từ.

* + *Phân tích ngữ nghĩa:*

Phân tích ngữ nghĩa là một cấu trúc được tạo bởi bộ phân tích cú pháp để gán ý nghĩa. Thành phần này chuyển các chuỗi từ tuyến tính thành cấu trúc. Nó cho thấy các từ được liên kết với nhau như thế nào.

Ngữ nghĩa chỉ tập trung vào nghĩa đen của từ, cụm từ và câu. Điều này chỉ trừu tượng hóa ý nghĩa từ điển hoặc ý nghĩa thực sự từ ngữ cảnh nhất định. Các cấu trúc được gán bởi bộ phân tích cú pháp luôn có ý nghĩa được gán

* + *Phân tích thực dụng:*

Phân tích thực dụng đề cập đến nội dung giao tiếp và xã hội tổng thể cũng như ảnh hưởng của nó đến việc diễn giải. Nó có nghĩa là trừu tượng hóa hoặc rút ra cách sử dụng ngôn ngữ có ý nghĩa trong các tình huống. Trong phân tích này, trọng tâm chính luôn là những gì được nói trong việc diễn giải lại ý nghĩa của nó.

Phân tích thực dụng giúp người dùng khám phá tác động dự kiến này bằng cách áp dụng một bộ quy tắc đặc trưng cho ra những cuộc đối thoại tích cực.

* + *Phân tích cú pháp:*

Các từ thường được chấp nhận là đơn vị cú pháp nhỏ nhất. Cú pháp đề cập đến các nguyên tắc và quy tắc chi phối cấu trúc câu của bất kỳ ngôn ngữ riêng lẻ nào.

Cú pháp tập trung vào thứ tự thích hợp của các từ có thể ảnh hưởng đến ý nghĩa của nó. Điều này liên quan đến việc phân tích các từ trong câu bằng cách theo dõi cấu trúc ngữ pháp của câu. Các từ được chuyển thành cấu trúc để cho thấy các từ có liên quan với nhau như thế nào.

* + *Tích hợp diễn ngôn:*

Nó có nghĩa là một cảm giác về bối cảnh. Ý nghĩa của bất kỳ câu nào phụ thuộc vào câu đó. Nó cũng xem xét ý nghĩa của following kết án.

Ví dụ, từ “that” trong câu “He want that” phụ thuộc vào ngữ cảnh diễn ngôn trước đó.

1.1.4 Ưu và nhược điểm của xử lý ngôn ngữ tự nhiên

Ưu điểm của NLP:

* + Người dùng có thể đặt câu hỏi về bất kỳ chủ để nào và nhận được phản hồi trực tiếp trong vòng vài giây.
  + Hệ thống NLP cung cấp câu trả lời cho các câu hỏi bằng ngôn ngữ tự nhiên.
  + Hệ thống NLP đưa ra câu trả lời chính xác cho các câu hỏi, không có thông tin không cần thiết hoặc không mong muốn.
  + Độ chính xác của câu trả lời tang theo lượng thông tin liên quan được cung cấp trong câu hỏi.
  + Quá trình NLP giúp máy tính giao tiếp với con người bằng ngôn ngữ của họ và thực hiện các nhiệm vụ khác liên quan đến ngôn ngữ.
  + Cho phép bạn thực hiện nhiều thao tác dữ liệu dựa trên ngôn ngữ hơn cho con người mà không mệt mỏi và một cách không thiên vị và nhất quán.
  + Cấu trúc nguồn dữ liệu phi cấu trúc cao

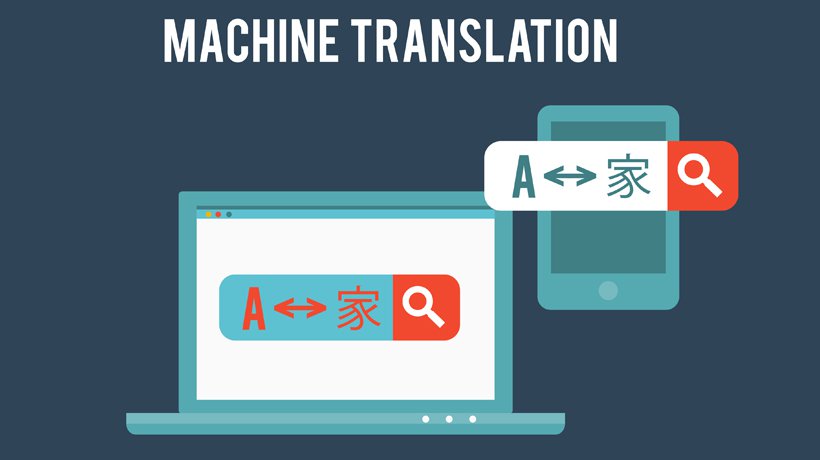
Nhược điểm của NLP:

* + Với ngôn ngữ truy vấn, hệ thống có thể không cung cấp được câu trả lời chính xác cho câu hỏi có từ ngữ kém hoặc mơ hồ.
  + Hệ thống được xây dựng chỉ cho một nhiệm vụ duy nhất và cụ thể, nó không thể thích ứng với các lĩnh vực và vấn đề mới vì chức năng hạn chế.
  + Hệ thống NLP không có giao diện người dung, thiếu các tính năng cho phép người dùng tương tác sâu hơn với hệ thống.

1.2 Bài toán dịch máy qua văn bản

1.2.1 Giới thiệu bài toán

Dịch tự động hay còn gọi là dịch máy (machine translation) là một nhánh của xử lý ngôn ngữ tự nhiên thuộc phân ngành trí tuệ nhân tạo, nó là sự kết hợp giữa ngôn ngữ, dịch thuật và khoa học máy tính. Như tên gọi, dịch tự động thực hiện dịch một ngôn ngữ này (gọi là ngôn ngữ nguồn) sang một ngôn ngữ khác (gọi là ngôn ngữ đích) một cách tự động, không có sự can thiệp của con người trong quá trình dịch.



Hình 1.3 Minh họa về dịch máy

Lịch sử một số phương pháp dịch máy:

* + ***Rule-based Machine Translation (RBMT):*** Tập trung vào các quy tắc giúp chuyển đổi văn bản trong ngôn ngữ nguồn (source) sang ngôn ngữ đích (target) trên các cấp độ: từ vựng, cú pháp hoặc ngữ nghĩa.

Các quy tắc thường do nhà ngôn ngữ học phát triển. Do vậy hạn chế chính của phương pháp này là nó đòi hỏi rất nhiều nguồn lực về chuyên môn/ chuyên gia (có thể rất tốn kém) để xây dựng rất nhiều quy tắc và ngoại lệ, đồng thời nó không khái quát được cho những ngôn ngữ khác.

* + ***Statistical machine translation (SMT):*** sử dụng các mô hình thống kê (statistical model) học cách dịch văn bản từ ngôn ngữ nguồn sang ngôn ngữ đích dựa trên một bộ ngữ liệu (corpus) lớn.

Ý tưởng đằng sau dịch máy thống kê đến từ lý thuyết thông tin. Tài liệu được dịch theo phân bố xác suất p(*e*|*f*) trong đó *e* là ngôn ngữ đích (ví dụ, Tiếng Việt) dịch từ *f* là ngôn ngữ nguồn (ví dụ, Tiếng Nhật).

* + ***Neural machine translation (NMT):*** sử dụng các mô hình neural network để học một mô hình thống kê cho quá trình dịch máy. Với phương pháp này, người ta chỉ cần huấn luyện một hệ thống duy nhất trên tập văn bản nguồn và văn bản đích (end-to-end system), không cần phải xây dựng một pipeline gồm các hệ thống chuyên biệt giống như SMT, không cần phải có nhiều kiến thức chuyên môn về ngôn ngữ, nhờ vậy mà có thể áp dụng cho các cặp ngôn ngữ khác nhau khá dễ dàng.

1.2.2 Lý do chọn đề tài

Lý do chọn đề tài về bài toán dịch máy là kết quả của sự nhận thức về tầm quan trọng và tiềm năng lớn mà việc tự động dịch ngôn ngữ mang lại. Dưới đây là một số lý do cụ thể:

* + Tiện ích và ứng dụng rộng rãi: Dịch máy giúp giao tiếp và hiểu biết giữa các quốc gia và văn hóa trở nên thuận lợi hơn. Từ việc du lịch đến kinh doanh quốc tế và giao tiếp trong các tổ chức đa quốc gia, nhu cầu về dịch máy ngày càng tăng cao.
  + Thách thức kỹ thuật và khoa học: Bài toán dịch máy là một trong những thách thức phức tạp nhất trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Đối mặt với sự đa dạng của ngôn ngữ và ngữ cảnh, việc tạo ra các hệ thống dịch máy chính xác và hiệu quả là một thách thức lớn đối với các nhà nghiên cứu và nhà phát triển.
  + Tiềm năng thương mại và xã hội: Việc phát triển các công nghệ dịch máy không chỉ mang lại lợi ích thương mại mà còn góp phần vào việc xây dựng cộng đồng toàn cầu hơn, giúp cho mọi người có thể truy cập và chia sẻ thông tin một cách dễ dàng hơn.
  + Phản ánh sự tiến bộ trong NLP: Sự phát triển của các mô hình học sâu và các kỹ thuật tiên tiến trong NLP đã giúp cải thiện đáng kể hiệu suất của các hệ thống dịch máy. Việc nghiên cứu và áp dụng những tiến bộ này vào bài toán dịch máy là một cơ hội để khám phá và tận dụng những tiềm năng mới trong lĩnh vực này.

Tổng quát, lựa chọn đề tài dịch máy không chỉ phản ánh sự quan tâm và cam kết đối với lĩnh vực NLP mà còn thể hiện mong muốn góp phần vào việc tạo ra các công nghệ thông tin có ý nghĩa và tiện ích cho xã hội.

1.2.3 Mô tả chi tiết đầu vào và đầu ra bài toán

Machine translation là một bài toán sequence-to-sequence (seq2seq) điển hình do nó có đầu vào là một chuỗi (sequence) và đầu ra cũng là một chuỗi. Một thách thức của bài toán này là độ dài của chuỗi đầu vào và chuỗi đầu ra biến đổi liên tục và không giống nhau. Một trong những cách tiếp cận hiệu quả đó là Encoder-Decoder.

Hệ thống này bao gồm 2 model:

* + Model thứ nhất được gọi là Encoder, chịu trách nhiệm nhận chuỗi đầu vào (input sequence) và mã hóa (encode) nó thành một vector có độ dài cố định.
  + Model thứ hai là Decoder có nhiệm vụ giải mã (decode) vector trên và dự đoán chuỗi đầu ra.

1.2.4 Các khó khăn và thách thức của bài toán

Bài toán dịch máy (Machine Translation) đang đối mặt với nhiều thách thức và khó khăn sau:

* + Không khớp lĩnh vực: Mô hình dịch máy thường được huấn luyện trên dữ liệu từ nhiều lĩnh vực khác nhau, do đó có thể không tốt trong việc dịch các văn bản chuyên ngành hoặc cụ thể.
  + Số lượng dữ liệu huấn luyện: Hiệu suất của mô hình dịch máy thường phụ thuộc vào số lượng và chất lượng dữ liệu huấn luyện.
  + Từ hiếm: Việc dịch các từ hiếm hoặc các từ không xuất hiện trong dữ liệu huấn luyện là một thách thức lớn.
  + Câu dài: Mô hình dịch máy thường gặp khó khăn khi dịch các câu dài, đặc biệt khi cần duy trì ngữ cảnh qua nhiều mệnh đề.
  + Các cụm từ địa phương và ngữ cảnh: Máy dịch có thể không thể xử lý chính xác các cụm từ địa phương hoặc các cụm từ cần ngữ cảnh để hiểu.
  + Nghĩa và ngữ nghĩa: Máy dịch có thể gặp khó khăn trong việc dịch chính xác các sắc thái, tiếng lóng và các cụm từ có liên quan đến văn hóa.
  + Thiếu nhận thức văn hóa và sự thống nhất: Có thể gây hại cho thương hiệu do thiếu nhận thức văn hóa và sự thống nhất.

1.2.5 Ứng dụng của bài toán

Bài toán dịch máy (Machine Translation) có nhiều ứng dụng thực tế, bao gồm:

* + Dịch trang web: Các phần mềm dịch có khả năng dịch gần như tức thì và tương đối chính xác toàn bộ trang web được viết bằng các ngôn ngữ khác.
  + Hỗ trợ giao tiếp đa ngôn ngữ: Dịch máy giúp người dùng có thể giao tiếp với người khác mà không cần biết ngôn ngữ của họ.
  + Học ngôn ngữ: Dịch máy cũng có thể được sử dụng như một công cụ hỗ trợ học ngôn ngữ.
  + Dịch văn bản chuyên ngành: Dịch máy có thể giúp dịch các văn bản chuyên ngành từ ngôn ngữ này sang ngôn ngữ khác.
  + Sửa lỗi ngôn ngữ: Một số mô hình dịch máy còn được áp dụng vào bài toán tự động sửa lỗi ngôn ngữ.

Bài toán dịch máy đang ngày càng trở thành một lĩnh vực nghiên cứu quan trọng, đồng thời mang lại nhiều ứng dụng hứa hẹn trong thế giới thực.

CHƯƠNG 2: CÁC KỸ THUẬT GIẢI QUYẾT BÀI TOÁN

2.1 Mô hình BART

2.1.1 Lý do chọn mô hình

BART (Bidirectional Autoregressive Transformers) đã chứng minh hiệu suất dịch thuật vượt trội so với các mô hình Transformer khác như Transformer cơ bản, Transformer decoder-only, và Transformer encoder-decoder trên nhiều tập dữ liệu và ngôn ngữ.

* + Điểm BLEU cao hơn: BART thường đạt điểm BLEU cao hơn so với các mô hình khác, cho thấy khả năng dịch chính xác và trôi chảy hơn.
  + Xử lý tốt các câu phức tạp: BART có khả năng xử lý tốt các câu phức tạp, dài hoặc chứa nhiều ẩn dụ, nhờ vào cơ chế attention mạnh mẽ giúp tập trung vào các phần quan trọng của văn bản.

BART sử dụng cơ chế attention để tập trung vào các phần quan trọng của văn bản, giúp dịch chính xác và trôi chảy hơn, đặc biệt là trong việc dịch các câu phức tạp, dài hoặc chứa nhiều ẩn dụ.

* + Hiểu rõ ngữ cảnh: BART có thể hiểu rõ ngữ cảnh của câu, giúp dịch chính xác ý nghĩa của tác giả.
  + Dịch tự nhiên: BART tạo ra bản dịch tự nhiên và dễ đọc, giống như bản dịch do con người thực hiện.

2.1.2 Giới thiệu chung

BART là mô hình được giới thiệu bởi Facebook AI, một mô hình pretrained mới kết hợp ưu điểm của BERT và GPT. Sức mạnh của BERT nằm ở việc nắm bắt ngữ cảnh hai chiều, trong khi đó GPT có khả năng tự hồi quy. Với sự ra đời của BART, các nhiệm vụ sinh và đọc hiểu văn bản có thể được thực hiện với cùng một mô hình.



Hình 2.1 Minh họa về BART

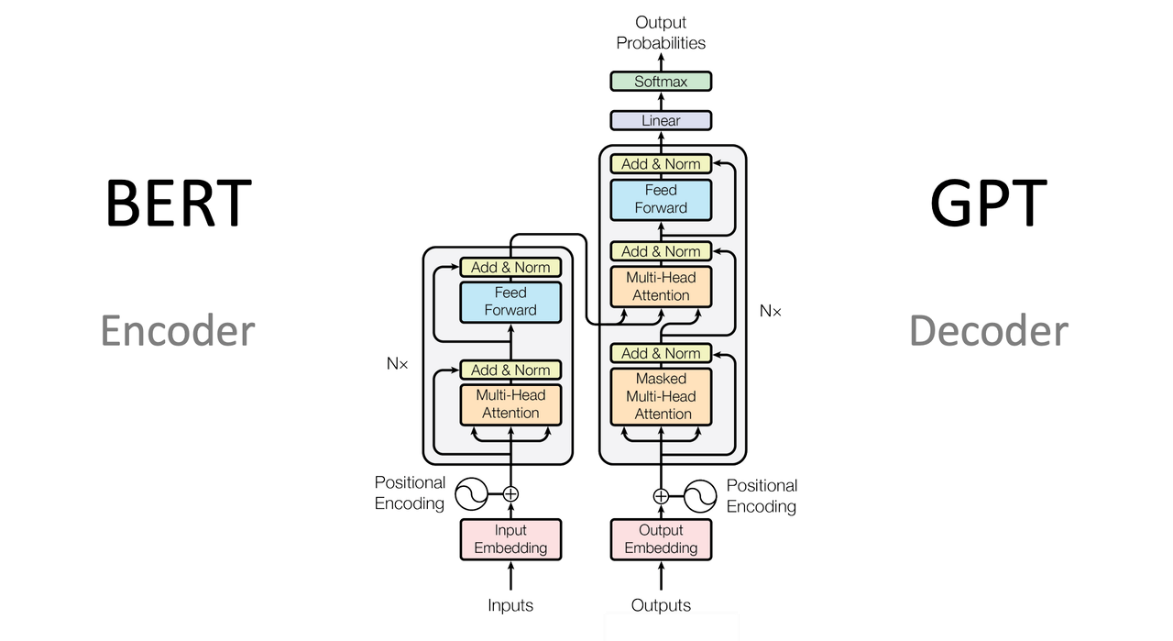
BART là một autoencoder khử nhiễu trên kiến trúc sequence-to-sequence, có thể được áp dụng trong đa dạng các nhiệm vụ khác nhau. Nó sử dụng kiến trúc transformers chuẩn cho bài toán dịch máy. Việc huấn luyện BART bao gồm việc tạo nhiễu trong văn bản với một hàm tùy ý và sử dụng mô hình để tái cấu trúc lại văn bản ban đầu. Ưu điểm chính của cách thức này là mô hình trở lên linh hoạt với văn bản đầu vào và tái tạo lại văn bản một cách hiệu quả.

BART cho thấy hiệu quả vượt trội trong cả nhiệm vụ sinh lẫn đọc hiểu văn bản. Cụ thể, BART có hiệu quả sánh ngang RoBERTa trên GLUE và SQuAD và đạt SOTA trong các nhiệm vụ về đối thoại trừu tượng, trả lời câu hỏi và tóm tắt.

Giống như các mô hình Transformer, BART gồm hai thành phần là Encoder và Decoder. Encoder được lấy từ BERT, nó có thể mã hóa xâu đầu vào theo cả hai chiều và lấy được nhiều thông tin ngữ cảnh hơn. Một số lượng ngẫu nhiên các token được che bằng mặt nạ và mô hình phải tự khôi phục chúng.

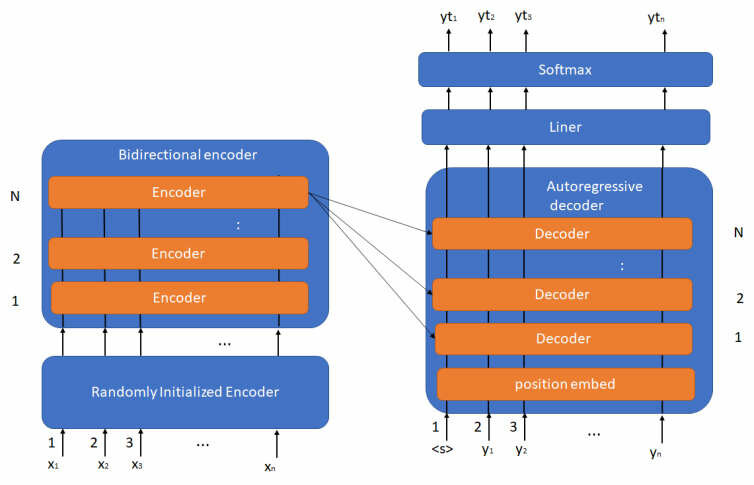
Decoder từ GPT được sử dụng để tái tạo lại đầu vào bị nhiễu. Mặc dù vậy, các từ chỉ có thể sinh từ bên trái, mô hình không thể học được tương tác hai chiều. Hàm kích hoạt GeLU được sử dụng thay thế cho ReLU. Kiến trúc cơ bản của BART sử dụng 6 tầng encoder và decoder trong khi kiến trúc mở rộng sử dụng 12 tầng.

2.1.3 Chi tiết mô hình BART



Hình 2.2 Minh họa kiến trúc mô hình Transformers

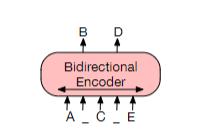
BART được xây dựng với mô hình autoencoder dùng cho việc loại bỏ nhiễu trong dữ liệu văn bản. BART được kết hợp từ BERT và GPT, trong đó BERT sử dụng mô hình Transformer với kiến trúc mã hóa kép (Bidirectional encoder), trong khi GPT sử dụng mô hình Transformer với kiến trúc bộ giải mã hồi quy tự động (Auto-regressive decoder).



Hình 2.3 Kiến trúc mô hình BART

Ngoài ra, còn có các thành phần khác như một bộ mã hóa được khởi tạo ngẫu nhiên để nhúng các câu của bộ mã hóa, thành phần tuyến tính để thay đổi hình dạng đầu ra từ tự hồi quy sang các vector tuyến tính, và thành phần softmax để dự đoán các vector tuyến tính thành các mã thông qua hàm softmax.

2.1.2.1 Bidirectional Encoder



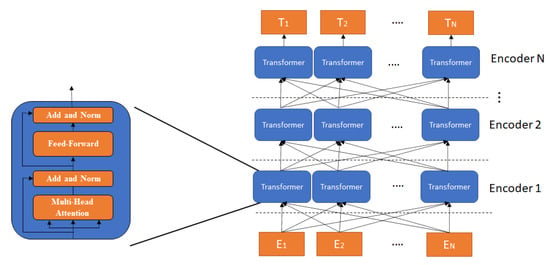
Hình 2.4 Khối Bidirectional Encoder

Trong mô hình BERT, các token ngẫu nhiên được thay thế bằng các token mask ([MASK]), và sau đó tài liệu được mã hóa theo cả hai chiều. Các token bị thiếu sau đó được dự đoán một cách độc lập. Điều này có nghĩa là BERT không thể dễ dàng được sử dụng cho việc sinh ra văn bản.

Lý do chính là do quá trình dự đoán các token bị thiếu trong BERT được thực hiện độc lập với nhau, không có sự phụ thuộc giữa các dự đoán. Điều này làm cho BERT không thể hiểu được ngữ cảnh toàn bộ của câu hoặc văn bản khi dự đoán các token bị thiếu, và do đó không thể tạo ra văn bản một cách liền mạch và logic như các mô hình khác có khả năng sinh văn bản.

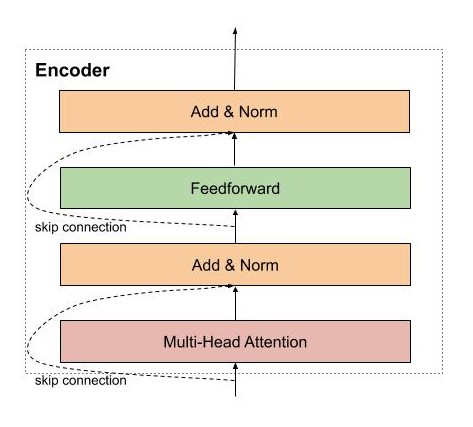
Vì vậy, mặc dù BERT rất mạnh mẽ trong việc hiểu ngữ cảnh và xử lý ngôn ngữ tự nhiên, nhưng do cách thức dự đoán token bị thiếu độc lập, nó không phù hợp cho việc sinh ra văn bản một cách tự nhiên và logic.

Hình 2.5 trình bày bộ mã hóa hai chiều của mô hình BART. Quá trình mã hóa bắt đầu với các token được nhúng như đầu vào cho các khối mã hóa transformer (tức là, lớp đầu tiên). Ngoài ra, đầu ra của lớp trước được truyền tới các lớp transformer tiếp theo. Ở lớp cuối cùng, mỗi bộ mã hóa transformer đầu ra một vector mô tả các đặc điểm của một token đầu vào.



Hình 2.5 Bộ mã hóa hai chiều của mô hình BART

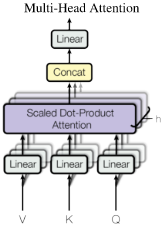
Trong Transformer có ba thành phần: multi-head attention, add and norm, và feed-forward. Attention đa đầu có một số lớp attention chạy song song. Thành phần cộng và chuẩn hóa thêm các vector đầu vào với đầu ra của attention đa đầu và áp dụng một dư thừa được chuẩn hóa. Feed-forward là một mạng nơ-ron tiếp theo.



Hình 2.6 Khối Encoder

* + Multi-head attention:

Mở rộng cơ chế attention ra thành nhiều heads, tức là mỗi câu sẽ được mã hóa với nhiều bộ query-key-value, cho ra nhiều ma trận attention.



Hình 2.7 Multi-Head Attention

* + - Attention đa đầu là một cơ chế quan trọng trong Transformer cho phép mô hình tập trung vào các phần quan trọng của dữ liệu đầu vào.
    - Nó bao gồm nhiều lớp attention chạy song song.
    - Mỗi lớp attention đa đầu chia dữ liệu đầu vào thành phần (head) và thực hiện attention trên từng phần đó.
    - Sau đó, các kết quả attention từ mỗi head được kết hợp thông qua việc kết hợp tuyến tính để tạo ra đầu ra cuối cùng.
  + Add and Norm (Cộng và chuẩn hóa):
* Sau khi tính toán attention đa đầu, đầu ra của nó được cộng với đầu vào ban đầu của layer đó.
* Sau đó, một quá trình chuẩn hóa được áp dụng để đảm bảo rằng đầu ra có cùng đơn vị và phân phối, giúp quá trình học trở nên ổn định hơn.
  + Feed-forward:
* Sau khi thông tin đã được kết hợp thông qua attention và add và norm, nó được đưa qua một lớp mạng nơ-ron tiếp theo.
* Lớp feed-forward này thường gồm hai lớp dày đặc (full connected) với một hàm kích hoạt ở giữa, chẳng hạn như hàm ReLU.
* Nó giúp mô hình học các biểu diễn phức tạp và phi tuyến tính của dữ liệu đầu vào.

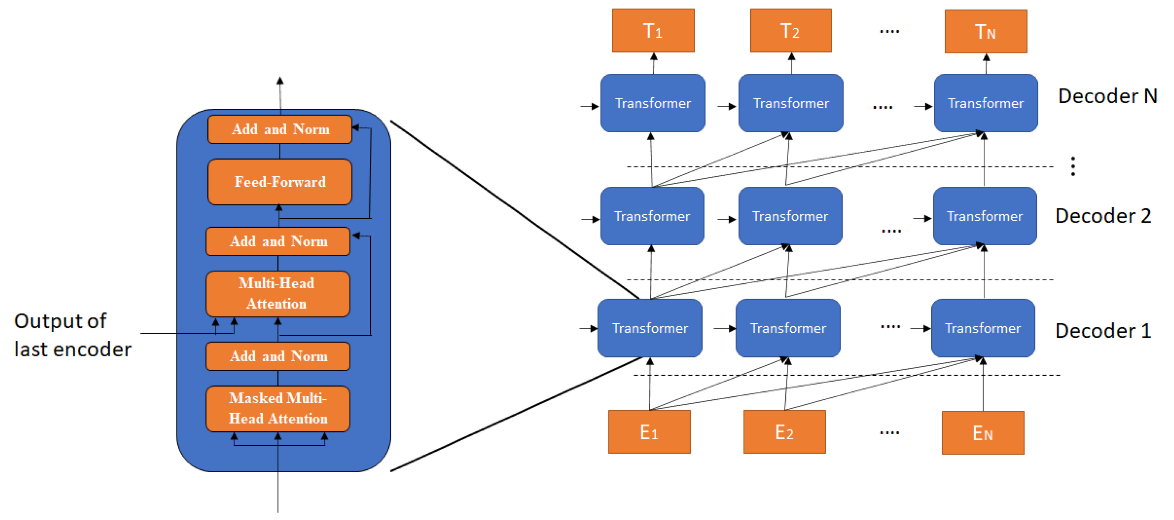
2.1.2.2 Auto-Regressive Decoder

Trong mô hình GPT, các token được dự đoán theo cách tự hồi quy (auto-regressively), có nghĩa là GPT có thể được sử dụng để sinh ra văn bản. Tuy nhiên, từ chỉ có thể phụ thuộc vào ngữ cảnh bên trái, do đó GPT không thể học được tương tác hai chiều.

Khi mô hình GPT dự đoán một từ, nó chỉ có thể xem xét các từ ở bên trái của từ đó trong câu. Điều này có nghĩa là GPT không thể sử dụng thông tin từ phía bên phải của từ đó để dự đoán, và do đó không thể học được các tương tác hai chiều trong văn bản.

Mặc dù GPT có khả năng sinh ra văn bản một cách tự nhiên và có thể dùng cho các nhiệm vụ sinh văn bản, nhưng hạn chế của nó là không thể học được tương tác hai chiều trong ngữ cảnh, điều này có thể ảnh hưởng đến khả năng hiểu và tạo ra văn bản phức tạp.

Bộ giải mã tự hồi quy của mô hình BART được trình bày trong Hình 2.8 Bộ giải mã tự hồi quy nhận các token được nhúng làm đầu vào. Đầu ra của mỗi lớp được xử lý cho lớp tiếp theo trong tất cả các lớp transformer. Ở lớp cuối cùng, mỗi bộ giải mã transformer tạo ra một vector đại diện cho các đặc điểm của các token đầu ra.



Hình 2.8 Bộ giải mã hồi quy của mô hình BART

Thành phần Transformer trong bộ giải mã tự hồi quy tương tự như trong bộ mã hóa hai chiều nhưng có một thành phần attention đa đầu thứ hai để liên kết đầu ra của bộ mã hóa hai chiều với bộ giải mã tự hồi quy.

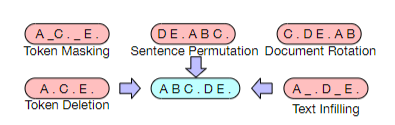
BART sử dụng bộ giải mã từ trái sang phải, đây là một đặc điểm của kiến trúc GPT. Bộ giải mã này hoạt động theo thứ tự từ trái sang phải trong quá trình huấn luyện trước, phù hợp với cách thiết lập được sử dụng trong quá trình sinh ra văn bản.

2.1.4 Xử lý đầu vào mô hình BART

Trong ngữ cảnh của mô hình BART, việc tạo ra nhiễu trong dữ liệu đầu vào là một phần quan trọng của quá trình huấn luyện. Các biến đổi này giúp mô hình học cách xử lý và phục hồi dữ liệu bị hỏng, từ đó cải thiện khả năng sinh văn bản và hiểu ngôn ngữ tự nhiên.

Các biến đổi này có thể bao gồm việc thay thế token bằng ký hiệu mask, xóa token ngẫu nhiên, điền văn bản vào các khoảng trống, hoán vị câu trong văn bản, và thay đổi thứ tự của các câu. Bằng cách kết hợp các biến đổi này, mô hình có thể học được nhiều loại nhiễu khác nhau và trở nên linh hoạt trong việc xử lý dữ liệu bị hỏng.

Việc thử nghiệm và kết hợp các biến đổi này giúp cải thiện hiệu suất của mô hình BART trong việc huấn luyện và tinh chỉnh cho các nhiệm vụ liên quan đến sinh văn bản và xử lý ngôn ngữ tự nhiên.



Hình 2.9 Các phép biến đổi xử lý đầu vào trong BART

Các biến đổi được thực hiện để tạo nhiễu trong dữ liệu đầu vào có thể bao gồm:

**1. Token Masking:** Theo phương pháp của BERT, các token ngẫu nhiên được chọn và thay thế bằng ký hiệu [MASK]. Điều này giúp mô hình học cách dự đoán các token bị ẩn.

**2. Token Deletion:** Các token ngẫu nhiên được xóa khỏi đầu vào. Khác với việc mask token, mô hình phải quyết định vị trí nào là các token bị thiếu.

**3. Text Infilling:** Một số đoạn văn bản được chọn, với độ dài của mỗi đoạn được rút ra từ phân phối Poisson (λ = 3). Mỗi đoạn được thay thế bằng một ký hiệu [MASK]. Các đoạn có độ dài bằng 0 tương ứng với việc chèn các token [MASK]. Phương pháp này giúp mô hình học cách dự đoán số lượng token bị thiếu trong một đoạn văn bản.

**4. Sentence Permutation:** Một tài liệu được chia thành các câu dựa trên dấu chấm, và các câu này được xáo trộn theo một thứ tự ngẫu nhiên. Điều này giúp mô hình học cách xử lý các câu trong một văn bản theo một thứ tự không cố định.

**5. Document Rotation:** Một token được chọn ngẫu nhiên, và tài liệu được xoay sao cho nó bắt đầu bằng token đó. Nhiệm vụ này giúp mô hình nhận biết điểm bắt đầu của tài liệu.

Các biến đổi này giúp mô hình BART học cách xử lý và tái tạo dữ liệu bị hỏng, từ đó cải thiện khả năng sinh văn bản và hiểu ngôn ngữ tự nhiên.

2.2 Tokenizer trong mô hình BART

Quá trình tokenization trong mô hình BART đóng vai trò quan trọng trong việc chuẩn bị dữ liệu đầu vào để có thể được sử dụng trong quá trình huấn luyện và dự đoán. Trong mô hình BART, tokenizer phải thực hiện các bước quan trọng để chia các đầu vào thành các token nhỏ, giúp mô hình hiểu và xử lý thông tin một cách hiệu quả.

2.2.1 Giới thiệu chung

Tokenization là quá trình chia một đoạn văn bản thành các đơn vị nhỏ hơn gọi là "token". Trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), tokenization là một bước quan trọng trong tiền xử lý dữ liệu, giúp máy tính hiểu được và xử lý thông tin ngôn ngữ một cách hiệu quả.

**Ý nghĩa của Tokenization:**

* Đơn vị ngữ nghĩa: Các token thường tương ứng với các đơn vị ngữ nghĩa như từ, subword, hoặc ký tự.
* Chuẩn hóa dữ liệu: Tokenization giúp chuẩn hóa dữ liệu bằng cách loại bỏ dấu câu, chấm câu và các ký tự không cần thiết khác.
* Tạo từ vựng: Các token được sử dụng để tạo từ vựng, là cơ sở cho việc mã hóa và huấn luyện mô hình NLP.

**Các Phương pháp Tokenization:**

* Tokenization dựa trên khoảng cách: Phân chia văn bản thành các token bằng cách sử dụng khoảng trắng hoặc dấu cách.
* Tokenization dựa trên từ (word-based): Chia văn bản thành các từ riêng lẻ.
* Tokenization dựa trên subword (subword-based): Chia văn bản thành các phần nhỏ hơn của từ, thường được sử dụng trong các ngôn ngữ có từ vựng lớn hoặc trong các tác vụ dịch máy.
* Tokenization dựa trên ký tự (character-based): Chia văn bản thành các ký tự riêng lẻ.

**Ứng dụng của Tokenization:**

* Xử lý ngôn ngữ tự nhiên (NLP): Tokenization là một bước quan trọng trong nhiều ứng dụng NLP như dịch máy, nhận diện thực thể, và phân loại văn bản.
* Xử lý văn bản trên mạng xã hội: Trong việc phân tích dữ liệu từ các nền tảng mạng xã hội, tokenization giúp xử lý và hiểu các thông điệp ngắn và tiện ích.
* Xử lý mã nguồn (source code): Trong phân tích mã nguồn, tokenization được sử dụng để chia mã nguồn thành các phần nhỏ hơn để phân tích cú pháp và nghĩa.

Tokenization là một khía cạnh quan trọng của xử lý ngôn ngữ tự nhiên và đóng vai trò quan trọng trong việc tiền xử lý dữ liệu và hiểu ngữ cảnh trong các ứng dụng liên quan đến ngôn ngữ.

Tokenizer trong mô hình BART thường sử dụng phương pháp mã hóa byte-pair (BPE) hoặc sentencepiece để tách các từ và subword từ dữ liệu đầu vào. Các từ và subword sau đó được mã hóa thành các token số hoặc mã token đặc biệt. Quá trình này đảm bảo rằng mô hình có thể hiểu được ngữ cảnh và cấu trúc của câu một cách chính xác

2.2.2 Byte-Pair Encoding tokenization

Byte-Pair Encoding (BPE) tiền thân được phát triển như một thuật toán để nén văn bản, sau đó được OpenAI sử dụng để tokenize khi huấn luyện trước mô hình GPT. Nó được sử dụng bởi rất nhiều mô hình Transformer, bao gồm GPT, GPT-2, RoBERTa, BART và DeBERTa.

Byte-Pair Encoding là một kỹ thuật phân tách token (tokenization) phổ biến trong xử lý ngôn ngữ tự nhiên (NLP), đặc biệt hữu ích cho các mô hình ngôn ngữ lớn (LLM). Nó hoạt động bằng cách mã hóa các từ thành chuỗi các đơn vị con (subword) dựa trên tần suất xuất hiện của chúng trong tập dữ liệu.

**Cách thức hoạt động:**

1. Khởi tạo từ điển: Bắt đầu với từ điển bao gồm tất cả các ký tự riêng biệt trong tập dữ liệu.
2. Lặp:
   1. Tìm cặp thường xuyên nhất: Xác định cặp ký tự liền kề xuất hiện nhiều nhất trong tập dữ liệu.
   2. Nối và cập nhật: Nối cặp ký tự đó thành một đơn vị con mới, sau đó thêm đơn vị con mới vào từ điển.
   3. Thay thế: Thay thế mọi lần xuất hiện của cặp ký tự ban đầu trong tập dữ liệu bằng đơn vị con mới.
3. Lặp lại: Lặp lại bước 2 cho đến khi đạt được kích thước từ điển mong muốn hoặc không còn cặp ký tự nào có thể hợp nhất.

**Ưu điểm của BPE:**

* + Xử lý từ vựng mở: BPE có thể xử lý các từ mới một cách hiệu quả bằng cách chia nhỏ chúng thành các đơn vị con đã có trong từ điển.
  + Giảm nhiễu dữ liệu: BPE có thể giúp giảm nhiễu dữ liệu trong tập dữ liệu bằng cách nhóm các ký tự thường xuất hiện cùng nhau.
  + Cải thiện hiệu suất mô hình: BPE có thể giúp cải thiện hiệu suất của các mô hình ngôn ngữ bằng cách cung cấp cho chúng thông tin chi tiết hơn về cấu trúc từ ngữ.

Kích thước từ điển BPE ảnh hưởng đến hiệu suất mô hình. Kích thước lớn hơn có thể giúp mô hình học được các chi tiết ngữ nghĩa tốt hơn, nhưng cũng có thể dẫn đến quá tải tính toán.

BPE có thể được kết hợp với các kỹ thuật khác như WordPiece để cải thiện hiệu quả hơn nữa.

2.2.3 So sánh các kỹ thuật tokenizer

Ngoài BPE tokenizer, còn có một số kỹ thuật tokenizer khác được sử dụng trong mô hình dịch máy, bao gồm:

* + Word tokenizer: Chia nhỏ văn bản thành các từ dựa trên khoảng trắng.
  + Subword tokenizer: Chia nhỏ các từ thành các đơn vị con nhỏ hơn, chẳng hạn như morpheme hoặc character n-gram.

Mỗi kỹ thuật tokenizer có ưu và nhược điểm riêng, dưới đây là bảng so sách các kỹ thuật tokenizer:

Bảng 2.1 So sánh các kỹ thuật tokenizer

|  |  |  |
| --- | --- | --- |
| Kỹ thuật tokenizer | Ưu điểm | Nhược điểm |
| Word tokenizer | Đơn giản, dễ hiểu | Không xử lý được các từ mới |
| Subword tokenizer | Xử lý được các từ mới | Có thể làm tăng kích thước tập dữ liệu |
| BPE tokenizer | Cân bằng giữa hiệu quả và khả năng xử lý | Phức tạp hơn word tokenizer và subword tokenizer |

2.3 Đơn vị đo của bài toán

Dịch máy là một lĩnh vực quan trọng trong xử lý ngôn ngữ tự nhiên (NLP) với mục tiêu tự động chuyển đổi văn bản từ ngôn ngữ này sang ngôn ngữ khác. Việc đánh giá hiệu suất của các mô hình dịch máy là một phần quan trọng trong quá trình phát triển và ứng dụng của chúng. Để đánh giá hiệu suất một cách khách quan và chính xác, cần sử dụng các đơn vị đo phù hợp.

2.3.1 Các đơn vị đo phổ biến trong dịch máy

Có nhiều đơn vị đo khác nhau được sử dụng trong dịch máy, mỗi đơn vị đo tập trung vào một khía cạnh khác nhau của hiệu suất dịch thuật. Dưới đây là một số đơn vị đo phổ biến nhất:

* **BLEU (BiLingual Evaluation Understudy):** BLEU là một trong những đơn vị đo phổ biến nhất trong dịch máy. Nó đo lường mức độ tương đồng giữa bản dịch tự động và các bản dịch do con người thực hiện. BLEU dựa trên so sánh tỷ lệ trùng khớp n-gram (n-gram là các chuỗi liên tiếp n từ) giữa bản dịch tự động và các bản dịch tham chiếu. Điểm BLEU cao hơn cho thấy bản dịch tự động gần giống với bản dịch do con người thực hiện hơn.
* **ROUGE (Recall-Oriented Understudy for Gisting Evaluation):** ROUGE là một đơn vị đo khác thường được sử dụng để đánh giá hiệu suất tóm tắt văn bản. Nó có thể được áp dụng cho dịch máy bằng cách đánh giá mức độ trùng khớp giữa bản dịch tự động và các bản tóm tắt do con người thực hiện. ROUGE sử dụng các chỉ số khác nhau như ROUGE-L (tập trung vào các chuỗi từ dài) và ROUGE-W (tập trung vào các từ đơn lẻ) để đánh giá hiệu suất.
* **METEOR (Metric for Evaluation of Translation with Explicit OR):** METEOR là một đơn vị đo kết hợp các khía cạnh của cả BLEU và ROUGE. Nó sử dụng sự kết hợp của tỷ lệ trùng khớp n-gram, ngữ nghĩa và trật tự từ để đánh giá hiệu suất dịch thuật. METEOR được đánh giá cao về khả năng phân biệt tốt hơn giữa các bản dịch chất lượng cao và chất lượng thấp.
* **CISBLEU (Case-Insensitive BLEU):** CISBLEU là biến thể của BLEU không phân biệt chữ hoa chữ thường. Nó được sử dụng để đánh giá hiệu suất dịch thuật cho các ngôn ngữ có chữ hoa chữ thường không đóng vai trò quan trọng trong ý nghĩa.
* **SACREBLEU (Scoring with Automatic Cleaning and Ranking of BLEU Outputs):** SacreBLEU là một biến thể khác của BLEU sử dụng kỹ thuật tự động làm sạch và xếp hạng đầu ra BLEU để cải thiện độ chính xác của đánh giá.

Trong bài toán dịch máy, tôi lựa chọn độ đo SacreBLEU để đánh giá hiệu suất của mô hình cho bài toán. Độ đo SacreBLEU phù hợp với mục tiêu, đặc điểm và kích thước của tập dữ liệu để tôi huấn luyện.

2.3.2 Độ đo SacreBLEU

SacreBLEU (Scoring with Automatic Cleaning and Ranking of BLEU Outputs) là một biến thể của BLEU (BiLingual Evaluation Understudy) được sử dụng để đánh giá hiệu suất dịch máy. So với BLEU truyền thống, SACREBLEU mang lại nhiều ưu điểm, giúp cung cấp đánh giá chính xác và đáng tin cậy hơn về chất lượng bản dịch.

**Tính toán SacreBLEU có thể được chia thành 4 bước sau:**

1. Làm sạch bản dịch tự động:

Bước đầu tiên là làm sạch bản dịch tự động để loại bỏ các lỗi chính tả, dấu chấm câu và các lỗi nhỏ khác. SacreBLEU sử dụng các kỹ thuật tự động để sửa lỗi dựa trên ngữ pháp và ngữ nghĩa của ngôn ngữ.

1. Xếp hạng bản dịch tham chiếu:

Bước tiếp theo là xếp hạng các bản dịch tham chiếu để chọn ra bản dịch phù hợp nhất cho từng câu. SacreBLEU sử dụng các kỹ thuật xếp hạng tinh vi dựa trên nhiều yếu tố, bao gồm độ tương đồng ngữ nghĩa, độ trôi chảy và độ phù hợp với ngữ cảnh.

1. Tính toán điểm BLEU:

Sau khi làm sạch bản dịch tự động và xếp hạng bản dịch tham chiếu, SACREBLEU sử dụng công thức BLEU truyền thống để tính toán điểm BLEU cho từng câu. Tuy nhiên, thay vì sử dụng tất cả các bản dịch tham chiếu, SACREBLEU chỉ sử dụng bản dịch tham chiếu được xếp hạng cao nhất.

1. Kết hợp điểm BLEU:

Cuối cùng, SACREBLEU kết hợp điểm BLEU của tất cả các câu để tính toán điểm SACREBLEU tổng thể cho bản dịch. Quá trình kết hợp này có thể sử dụng các phương pháp khác nhau, ví dụ như trung bình cộng hoặc trung bình trọng số.

**Công thức BLEU truyền thống:**

BLEU = 100 \* exp (1/4 \* Σₙ (wₙ \* log(pₙ)))

Trong đó:

* + - BLEU là điểm số BLEU của bản dịch máy tính.
    - Σₙ (wₙ \* log(pₙ)) là tổng của trọng số của các n-gram nhân với logarithm tự nhiên của precision của n-gram tương ứng.
    - exp là hàm mũ.
    - 1/4 là hằng số mà một số bài báo đã đề xuất là giá trị tốt nhất cho việc tính toán BLEU.
    - wₙ là trọng số của n-gram, thường là 1/n nếu cân nhắc tất cả các kích thước n-gram.
    - pₙ là precision của n-gram.

**SACREBLEU cải thiện BLEU truyền thống bằng cách:**

* Sử dụng kỹ thuật làm sạch bản dịch tự động để giảm nhiễu.
* Sử dụng kỹ thuật xếp hạng bản dịch tham chiếu để chọn ra bản dịch phù hợp nhất.
* Kết hợp điểm BLEU của tất cả các câu một cách hiệu quả.

Kết quả là SACREBLEU mang lại đánh giá chính xác và đáng tin cậy hơn về chất lượng bản dịch so với BLEU truyền thống.

2.4 Quy trình giải quyết bài toán

2.4.1 Kỹ thuật Back-Translation

Back translation (BT) là một kỹ thuật dịch máy thống kê (SMT) được sử dụng để tăng cường dữ liệu huấn luyện cho các mô hình dịch máy. Kỹ thuật này hoạt động bằng cách dịch văn bản nguồn sang ngôn ngữ trung gian, sau đó dịch lại sang ngôn ngữ mục tiêu.

Back translation dựa trên giả thuyết rằng bản dịch ngược (từ ngôn ngữ mục tiêu sang ngôn ngữ nguồn) sẽ gần giống với văn bản nguồn ban đầu. Do đó, bản dịch ngược có thể được sử dụng để bổ sung dữ liệu huấn luyện cho mô hình dịch máy, giúp cải thiện hiệu suất dịch thuật

**Back translation có một số ưu điểm như sau:**

* Tăng cường dữ liệu huấn luyện: BT có thể được sử dụng để tạo ra nhiều dữ liệu huấn luyện hơn so với dịch trực tiếp từ ngôn ngữ nguồn sang ngôn ngữ mục tiêu.
* Cải thiện hiệu suất dịch thuật: BT có thể giúp cải thiện hiệu suất dịch thuật, đặc biệt là cho các cặp ngôn ngữ có ít dữ liệu song song.
* Dễ dàng triển khai: BT có thể được triển khai dễ dàng bằng cách sử dụng các công cụ dịch máy hiện có.

**Back translation cũng có một số nhược điểm như sau:**

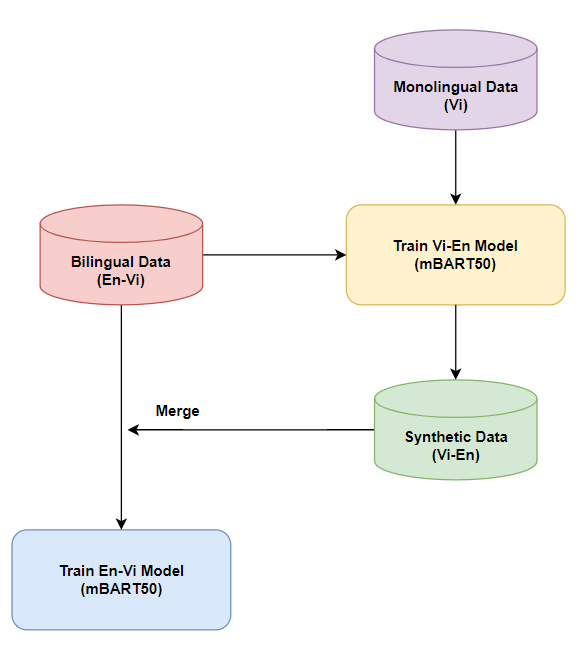
* Chất lượng bản dịch ngược: Chất lượng bản dịch ngược có thể ảnh hưởng đến hiệu suất dịch thuật tổng thể. Nếu bản dịch ngược không chính xác, nó có thể làm giảm hiệu suất dịch thuật.
* Tốn thời gian và chi phí: BT có thể tốn thời gian và chi phí vì nó yêu cầu hai lần dịch cho mỗi câu.
* Có thể tạo ra lỗi: BT có thể tạo ra lỗi dịch thuật mới nếu bản dịch ngược không chính xác.

Kỹ thuật dịch ngược chỉ là một trong nhiều kỹ thuật để huấn luyện mô hình dịch máy.

Hiệu suất của mô hình dịch máy phụ thuộc vào nhiều yếu tố, bao gồm chất lượng của bộ dữ liệu, cấu hình mô hình và thuật toán huấn luyện.

Để đạt được hiệu quả dịch tốt nhất, cần sử dụng kết hợp nhiều kỹ thuật và thuật toán khác nhau.

2.4.2 Quy trình giải quyết



Hình 2.10 Pipeline phương pháp giải quyết bài toán

Kỹ thuật dịch ngược bao gồm các bước sau:

* *Bước 1*: Xây dựng bộ dữ liệu đơn ngữ (Monolingual data) dành cho tiếng Việt.
  + Thu thập dữ liệu văn bản tiếng Việt.
  + Loại bỏ nhiễu và chuẩn hóa dữ liệu.
* *Bước 2*: Huấn luyện mô hình mBART50 với chiều dịch là Việt - Anh.
  + Sử dụng bộ dữ liệu song ngữ Việt - Anh để huấn luyện mô hình mBART50 với chiều dịch Việt - Anh.
  + Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.
* *Bước 3*: Tạo bộ dữ liệu tổng hợp sử dụng mô hình mBART50 Vi-En trên bộ dữ liệu đơn ngữ.
  + Sử dụng mô hình mBART50 Vi-En đã huấn luyện để dịch các văn bản tiếng Việt sang tiếng Anh.
  + Kết hợp các bản dịch tiếng Anh tự động với các văn bản tiếng Việt gốc để tạo ra bộ dữ liệu tổng hợp.
* *Bước 4*: Kết hợp bộ dữ liệu tổng hợp với bộ dữ liệu gốc để huấn luyện mô hình En-Vi
  + Tạo mô hình mBART50 mới với chiều dịch Anh - Việt.
  + Tải bộ dữ liệu tổng hợp và bộ dữ liệu gốc vào mô hình.
  + Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.

CHƯƠNG 3: THỰC NGHIỆM

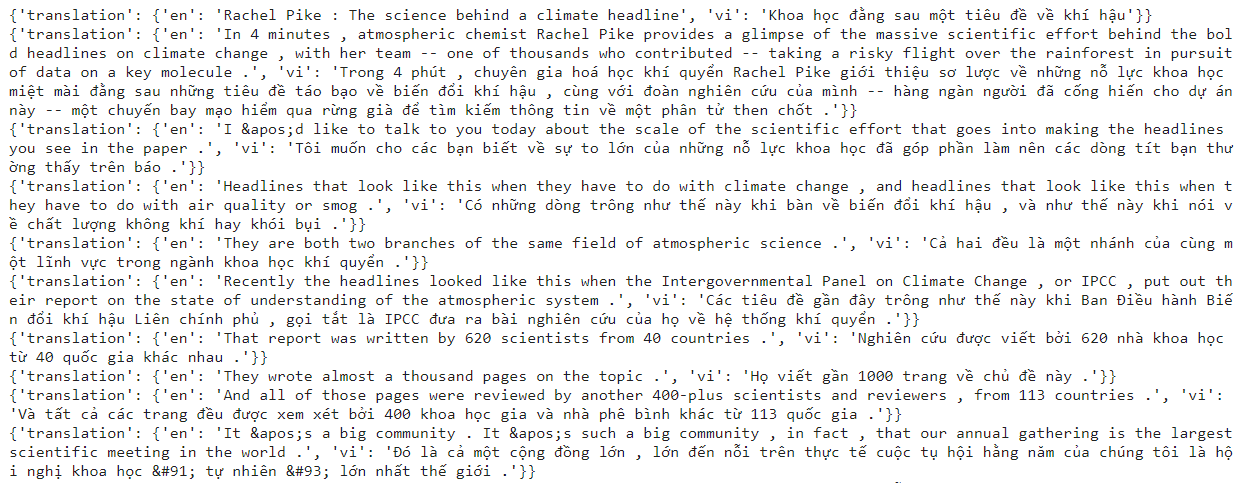
3.1 Bộ dữ liệu thực nghiệm

Quá trình thực nghiệm được tôi tiến hành trên 2 bộ dữ liệu song ngữ Anh-Việt là IWSLT2015-en-vi và PhoMT.

Trong đó, bộ dữ liệu IWSLT2015-en-vi là một bộ sưu tập câu song ngữ Anh-Việt được sử dụng rộng rãi trong nghiên cứu dịch máy. Nó được phát triển cho hội thảo IWSLT (International Workshop on Speech and Language Translation) năm 2015 và là một trong những tập dữ liệu dịch máy tiếng Anh sang tiếng Việt được sử dụng phổ biến nhất.

Đặc điểm:

* Nguồn gốc: Các câu được trích xuất từ các nguồn đa dạng như Wikipedia, LibriSpeech, TED Talks và các trang web tin tức.
* Định dạng: Các câu được lưu trữ trong tệp JSON với cấu trúc đơn giản, dễ dàng xử lý bằng các công cụ lập trình.



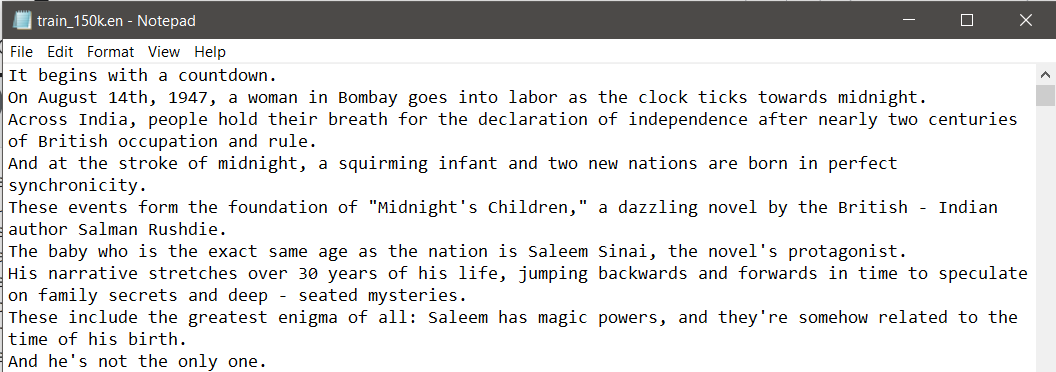
Hình 3.1 Ví dụ về dataset IWSLT2015-en-vi

Tập dữ liệu IWSLT2015-en-vi là một nguồn tài nguyên có giá trị cho nghiên cứu dịch máy tiếng Anh sang tiếng Việt. Với kích thước vừa phải, chất lượng cao và tính đa dạng, nó được sử dụng rộng rãi để đánh giá hiệu suất, huấn luyện và phát triển các mô hình dịch máy. Tuy nhiên, cần lưu ý rằng kích thước tương đối nhỏ và thiếu tính đa dạng về chủ đề có thể hạn chế khả năng tổng quát hóa của mô hình dịch máy.

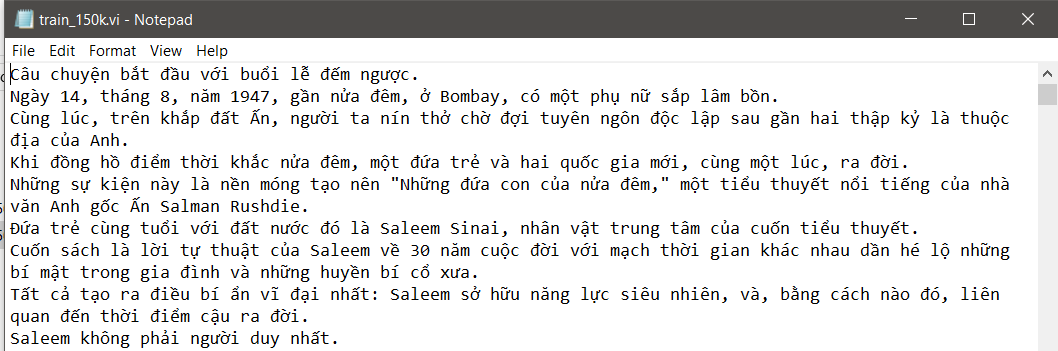
Bộ dữ liệu còn lại là PhoMT, một bộ dữ liệu dịch máy Việt-Anh quy mô lớn được phát triển bởi VinAI Research. Bộ dữ liệu này được giới thiệu tại EMNLP 2021 và được đánh giá cao về chất lượng và độ tin cậy.

Đặc điểm:

* Chất lượng cao: Câu được dịch bởi các chuyên gia ngôn ngữ giàu kinh nghiệm, đảm bảo độ chính xác và trôi chảy.
* Độ tin cậy cao: Câu được kiểm tra kỹ lưỡng để loại bỏ lỗi và đảm bảo tính nhất quán.
* Đa dạng: Bao gồm nhiều chủ đề khác nhau, từ tin tức, giải trí đến khoa học kỹ thuật, giúp nâng cao hiệu suất dịch máy cho nhiều lĩnh vực.



Hình 3.2 Ví dụ về dataset PhoMT tiếng Anh



Hình 3.3 Ví dụ về dataset PhoMT tiếng Việt

PhoMT là một bộ dữ liệu dịch máy Việt-Anh chất lượng cao và quy mô lớn. Bộ dữ liệu này là một nguồn tài nguyên quý giá cho nghiên cứu và ứng dụng dịch máy.

PhoMT được cung cấp miễn phí cho mục đích nghiên cứu và phi thương mại. Có thể tải xuống bộ dữ liệu từ trang web VinAI Research: <https://github.com/VinAIResearch>.

Thông số chi tiết của các bộ dữ liệu, xem trong bảng dưới đây

Bảng 3.1 Dữ liệu thực nghiệm

|  |  |  |  |
| --- | --- | --- | --- |
| Dữ liệu | Số lượng train | Số lượng validation | Số lượng test |
| IWSLT2015-en-vi | 133317 | 1553 | 1269 |
| PhoMT | 150000 | 18719 | 19151 |

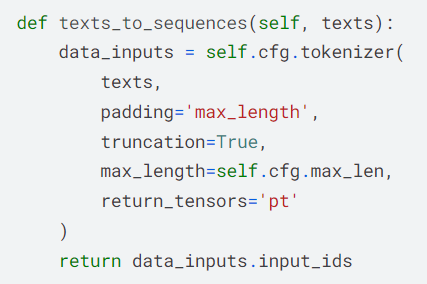
3.2 Huấn luyện các mô hình

Các mô hình được nghiên cứu tiến hành thực nghiệm trên môi trường Kaggle với ngôn ngữ lập trình Python. Kaggle hỗ trợ GPU P100 được sử dụng 30 giờ/tuần giúp quá trình huấn luyện các mô hình diễn ra nhanh hơn.

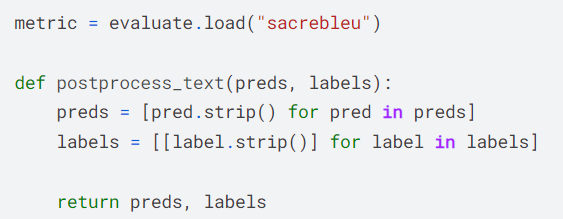
Đầu tiên, hàm read\_data được thiết kế để đọc và trích xuất dữ liệu từ bộ dữ liệu "iwslt2015-en-vi", cụ thể là bộ dữ liệu dịch tiếng Anh sang tiếng Việt. Cụ thể, văn bản nguồn được trích xuất từ trường có khóa là self.cfg.src\_lang và văn bản đích từ trường có khóa là self.cfg.tgt\_lang. Điều này giúp đảm bảo rằng dữ liệu đầu vào và đầu ra được trích xuất từ ngôn ngữ được xác định trước trong cấu hình. Dưới đây là mã nguồn của hàm này:

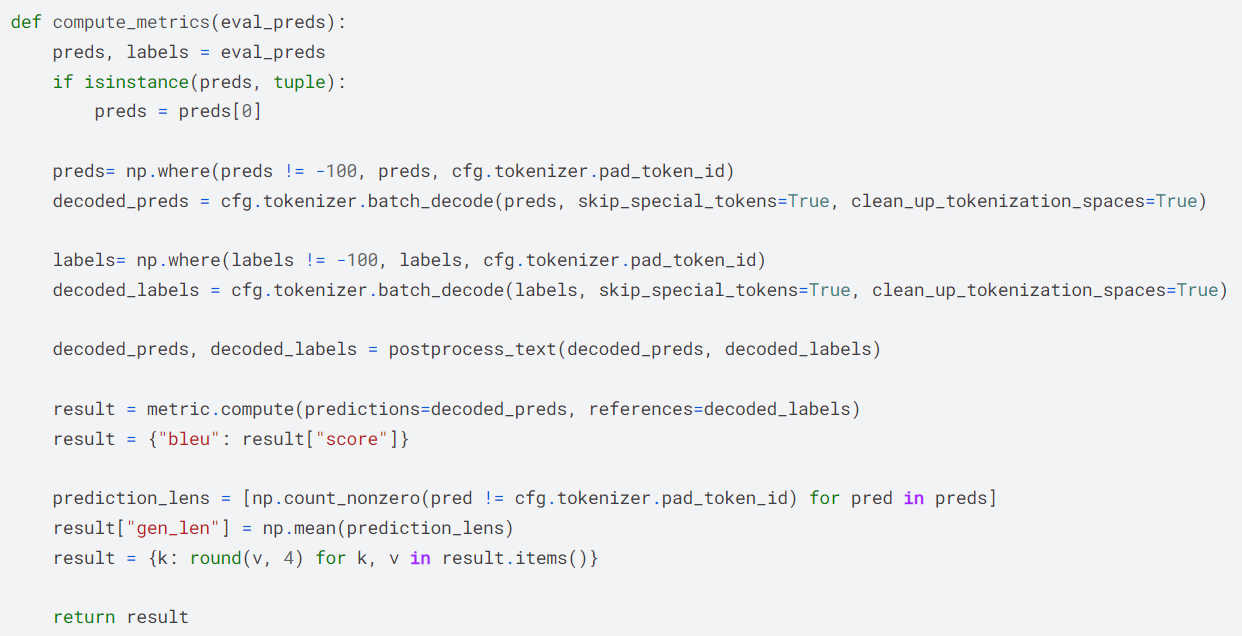


Hàm texts\_to\_sequences chịu trách nhiệm chuyển đổi các văn bản thành chuỗi số (sequences), sử dụng một tokenizer được cung cấp trong cấu hình. Hàm texts\_to\_sequences là một phần quan trọng của quá trình chuẩn bị dữ liệu cho mô hình dịch máy. Nó chuyển đổi các văn bản thành dạng số hóa mà mô hình có thể hiểu được, giúp đảm bảo rằng dữ liệu đầu vào được chuẩn hóa và tối ưu hóa cho quá trình huấn luyện và dự đoán.



Đoạn mã sau đây dùng để đánh giá chất lượng bản dịch của mô hình dịch máy:





Sử dụng metric sacrebleu để tính điểm BLEU: Điểm BLEU là thang đo phổ biến để đánh giá mức độ giống nhau giữa bản dịch tự động và bản dịch của con người.

Tính toán độ dài trung bình của bản dịch: Đây là số lượng từ trung bình trong mỗi câu dịch.

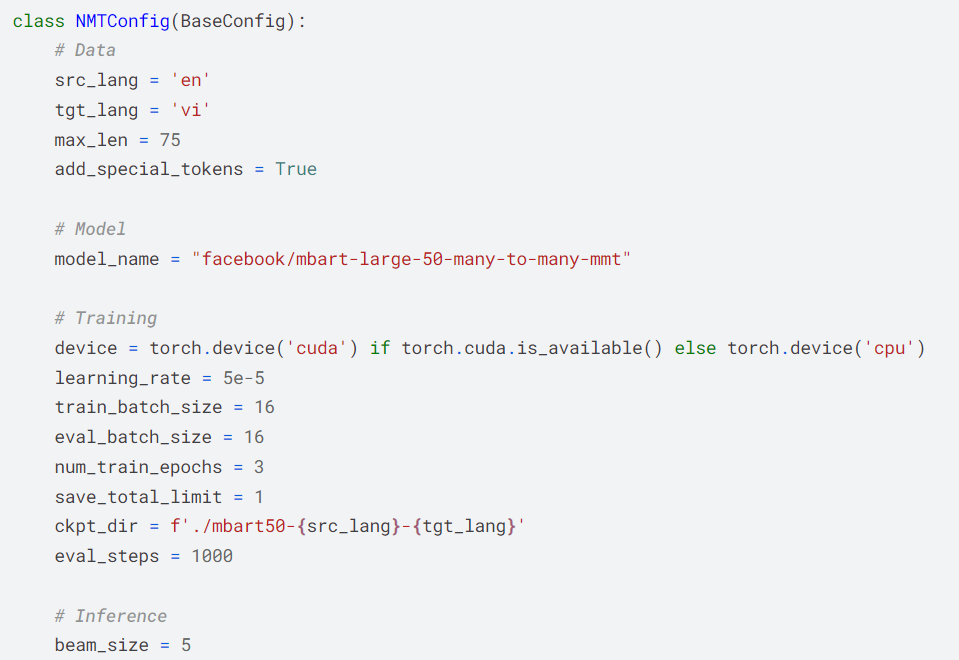
Trước khi tính toán, mã thực hiện một số bước xử lý trên văn bản:

* Loại bỏ các khoảng trắng thừa.
* Chuyển đổi văn bản thành dạng phù hợp cho metrix.
* Thay thế các từ đặc biệt bằng các ký hiệu quy định.

Kết quả cuối cùng là một từ điển với hai thông tin:

* bleu: Điểm BLEU của bản dịch.
* gen\_len: Độ dài trung bình của bản dịch.

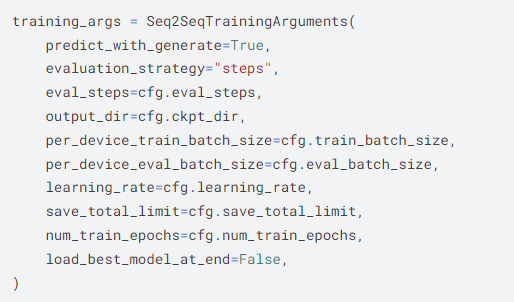
3.2.1 Mô hình mBART50 từ Anh sang Việt

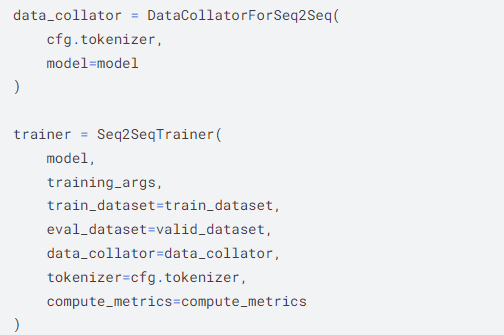


Lớp NMTConfig trong đoạn mã được thiết kế để cấu hình các thông số cần thiết cho một mô hình dịch máy (Neural Machine Translation - NMT). src\_lang là ngữ nguồn (mặc định là 'en' - tiếng Anh), tgt\_lang là ngôn ngữ đích (mặc định là 'vi' - tiếng Việt). Độ dài tối đa của chuỗi, thiết lập là 75 token. Đây là giới hạn trên cho độ dài của các câu sau khi được token hóa, add\_special\_tokens để bằng True để quy định việc có thêm các token đặc biệt (như [CLS], [SEP]) vào chuỗi token hay không.

Mô hình sử dụng bộ pretrained ở đây là "facebook/mbart-large-50-many-to-many-mmt". Đây là một mô hình dịch đa ngôn ngữ của Facebook AI, hỗ trợ dịch nhiều ngôn ngữ khác nhau. Cấu hình để huấn luyện có learning rate 5e-5 và huấn luyện 3 epoch.

Sau khi khai báo các phương pháp, thủ tục và module sẽ sử dụng trong quá trình huấn luyện như trên, tôi tiến hành huấn luyện với bộ dữ liệu đã chuẩn bị. Đoạn mã khởi tạo trình huấn luyện và huấn luyện như sau:



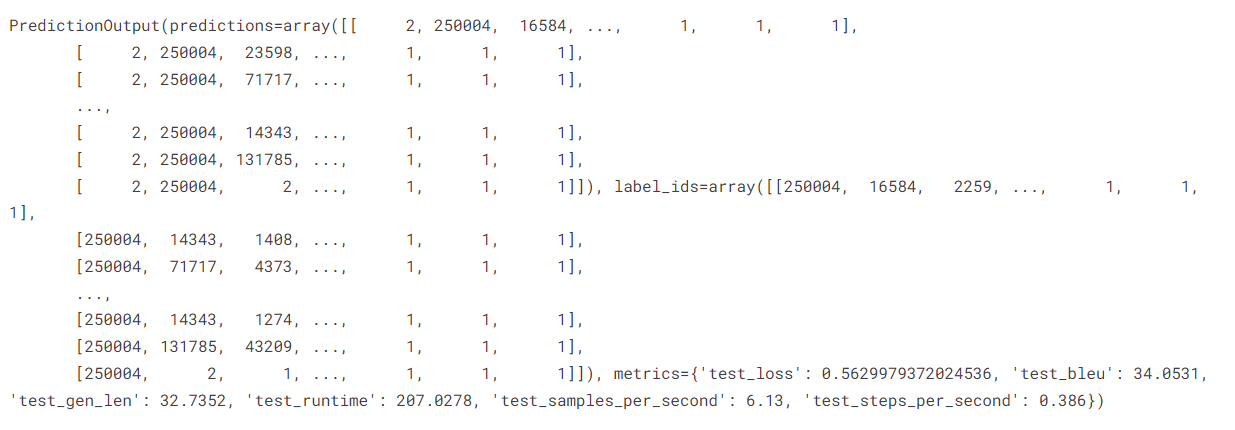




Dưới đây là kết quả của quá trình huấn luyện:

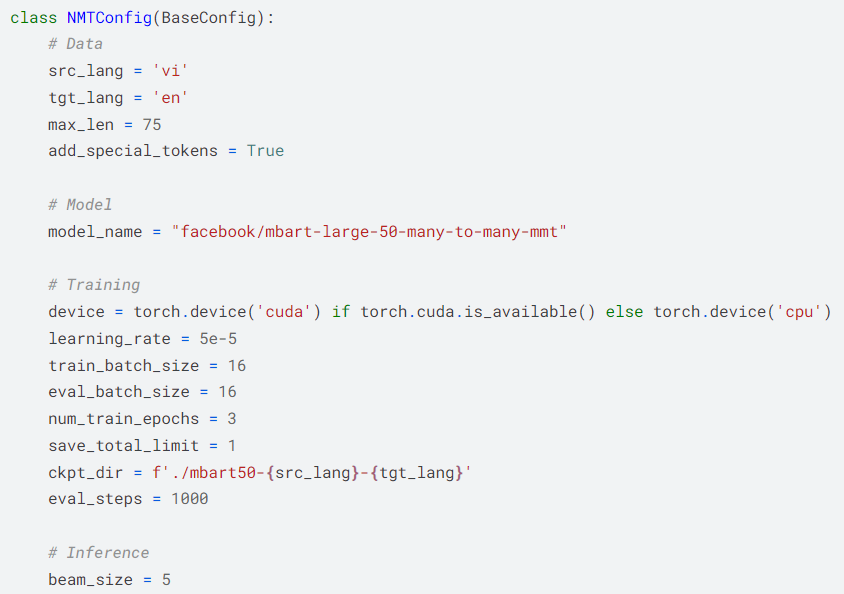


Kết quả huấn luyện cho thấy giá trị bleu ở step 20000 có giá trị cao nhất là 34.3937 và loss là 0.565974 lên tôi sẽ lấy model ở step 20000. Sau đó tôi sử dụng mô hình dự đoán bộ test của IWSLT2015 có 1269 mẫu được kết quả như sau:



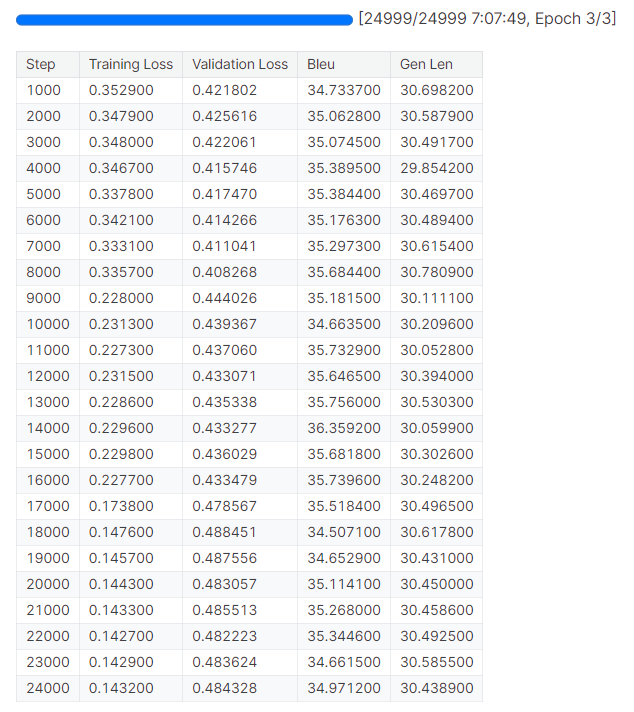
Giá trị bleu là 34.0531 và loss là 0.56299 so với kết quả huấn luyện thì thấy giá trị không bị lệch quá nhiều, vì vậy mô hình đang hoạt động ổn định.

3.2.2 Mô hình mBART50 từ Việt sang Anh

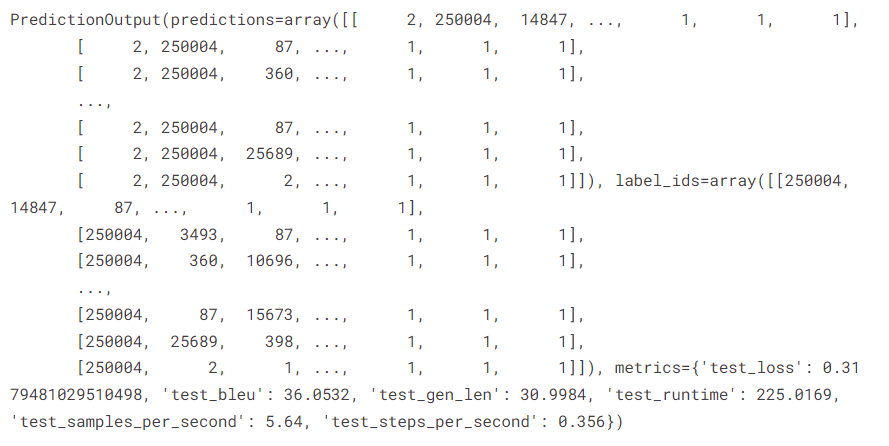


Lớp NMTConfig trong đoạn mã được thiết kế giống với cấu hình mô hình từ Anh sang Việt, chỉ khác mỗi src\_lang là ngữ nguồn (mặc định là 'vi' - tiếng Việt), tgt\_lang là ngôn ngữ đích (mặc định là 'en' - tiếng Anh). Độ dài tối đa của chuỗi, thiết lập là 75 token.

Dưới đây là kết quả của quá trình huấn luyện:



Kết quả huấn luyện cho thấy giá trị bleu ở step 14000 có giá trị cao nhất là 36.3592 và có loss là 0.433277 lên tôi sẽ lấy model ở step 14000. Sau đó tôi sử dụng mô hình dự đoán bộ test của IWSLT2015 có 1269 mẫu được kết quả như sau:



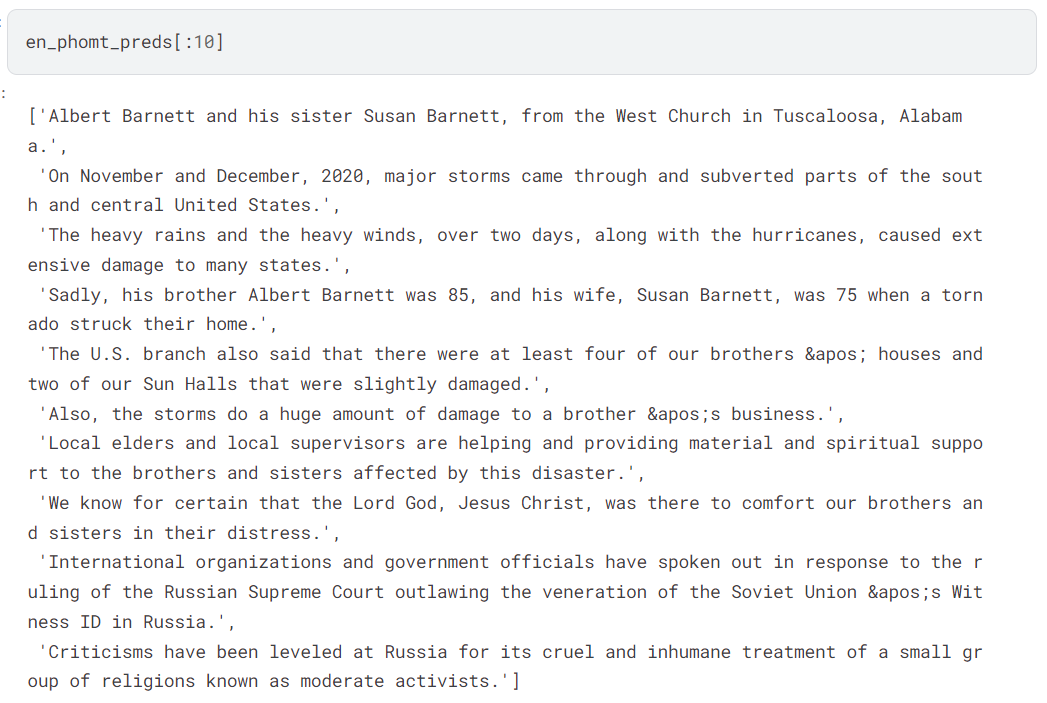
Giá trị bleu là 36.0532 và loss là 0.31 so với kết quả huấn luyện thì thấy giá trị không bị lệch quá nhiều, vì vậy mô hình đang hoạt động ổn định.

3.2.3 Back Translation từ Anh sang Việt

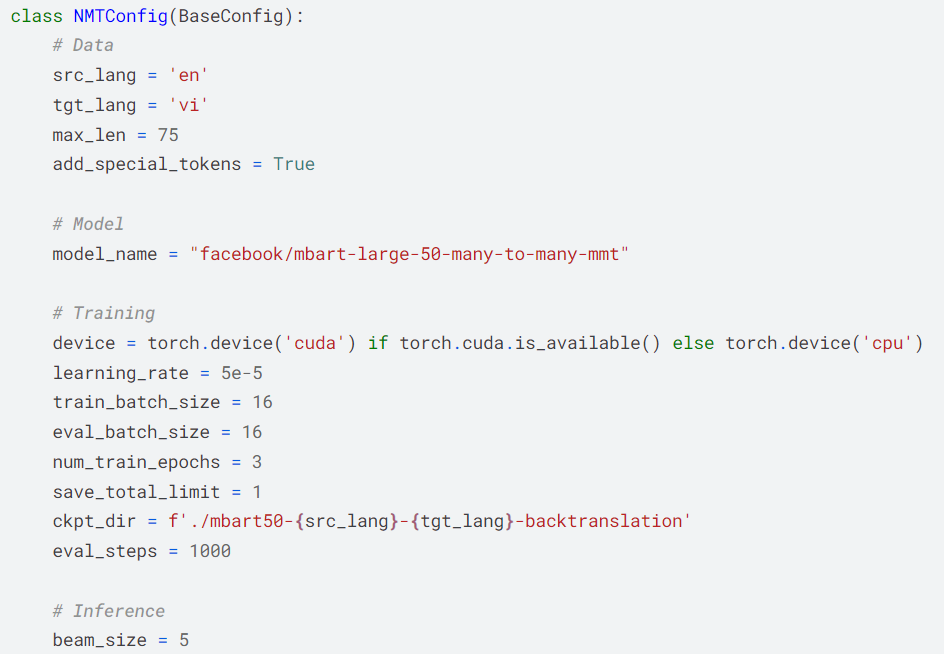
Trước khi huấn luyện, tôi lấy mô hình dịch đã huấn luyện trước dịch từ Việt sang Anh để chuyển bộ dữ liệu crawl trên các nguồn khác nhau. Ở đây thì tôi sẽ sử dụng bộ test của PhoMT với ngôn ngữ tiếng Việt đã được thu thập trên các nguồn đa dạng.



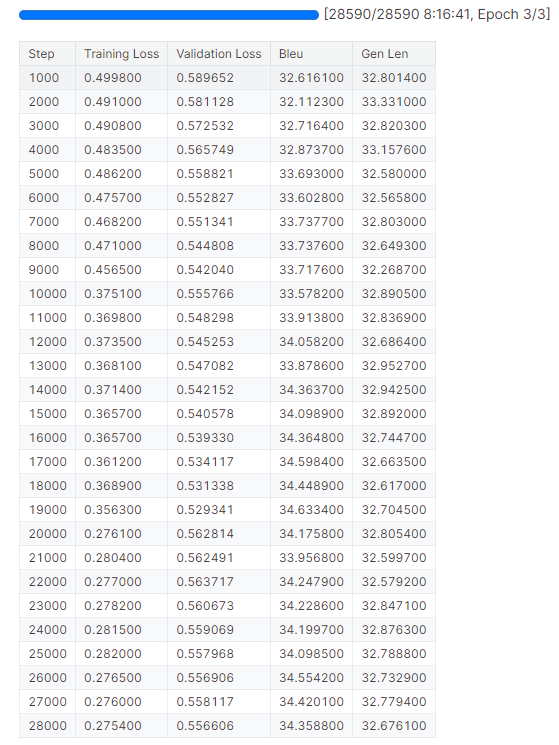
Bộ test sau khi được dịch sang tiếng Anh để cho vào bộ dữ liệu huấn luyện:



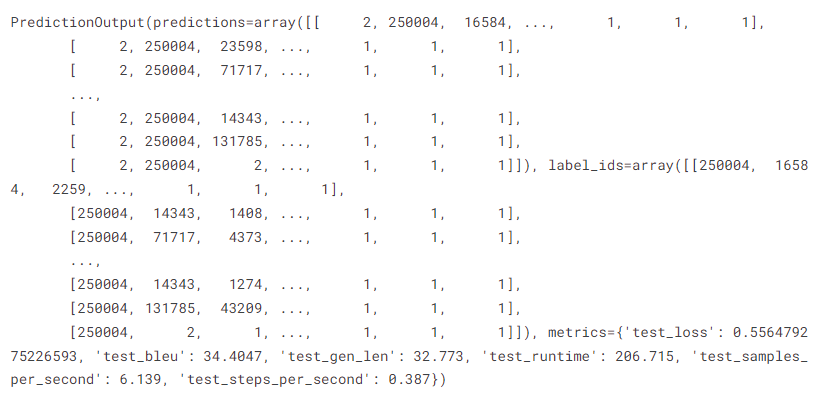
Cấu hình để huấn luyện sẽ giống với mBART50 dịch từ Anh sang Việt:



Dưới đây là kết quả của quá trình huấn luyện:



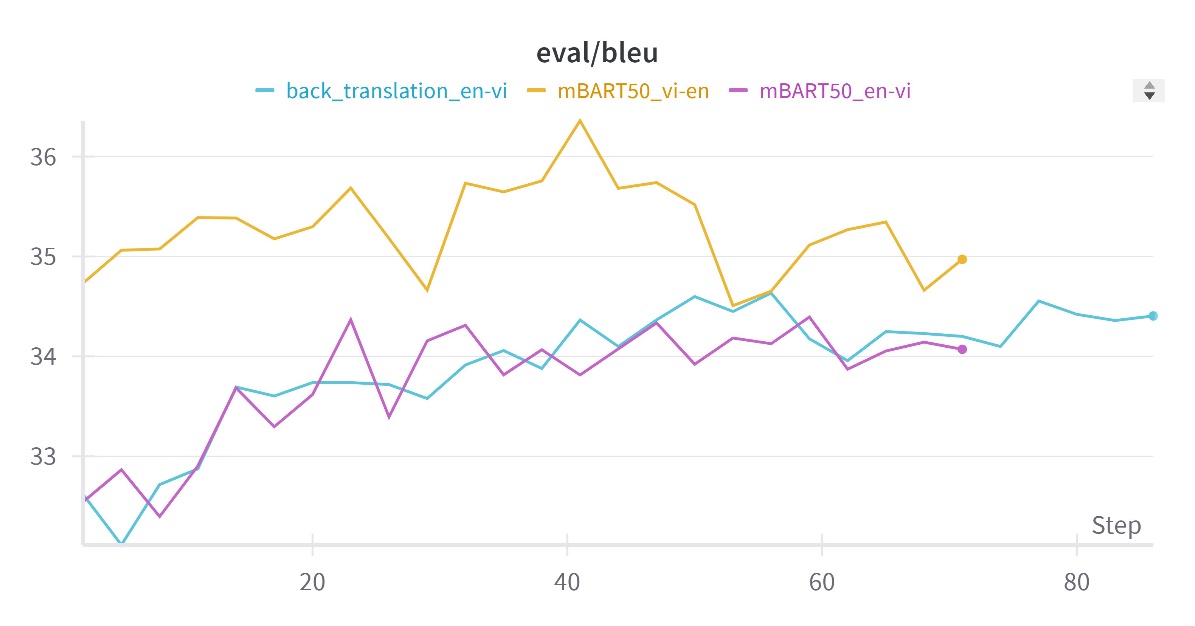
Kết quả huấn luyện cho thấy giá trị bleu ở step 19000 có giá trị cao nhất là 34.6334 và loss là 0.529341 lên tôi sẽ lấy model ở step 19000. Sau đó tôi sử dụng mô hình dự đoán bộ test của IWSLT2015 có 1269 mẫu được kết quả như sau:



Giá trị bleu là 34.4047 và loss là 0.556 so với kết quả huấn luyện thì thấy giá trị không bị lệch quá nhiều, vì vậy mô hình đang hoạt động ổn định.

3.3 Các kết quả thực nghiệm

Trước tiên, để kiểm tra tính ổn định cũng như tốc độ hội tụ của mô hình mBART50 được đề xuất. Tôi tiến hành thử nghiệm mô hình với bộ dữ liệu IWSLT2015 và bộ test của PhoMT để tăng cường dữ liệu. Hình 3.1 và 3.2 lần lượt biểu thị giá trị bleu và giá trị hàm tổn thất qua các step trong quá trình huấn luyện.



Hình 3.4 Biểu đồ giá trị Bleu trong quá trình huấn luyện



Hình 3.5 Biểu đồ giá trị Loss trong quá trình huấn luyện

Các kết quả cho thấy quá trình huấn luyện diễn ra ổn định, không còn hiện tượng giảm đột ngột (drop) của Bleu sau 3 epoch trên bộ dữ liệu IWSLT2015.

Dưới đây là các bảng đánh giá lấy model tốt nhất qua từng step khi huấn luyện:

Bảng 3.2 Kết quả tốt nhất của các mô hình

|  |  |  |  |
| --- | --- | --- | --- |
| Mô hình | Bleu Score | Loss | Gen Len |
| mBART50\_vi-en | 36.3592 | 0.433277 | 32.8526 |
| mBART50\_en-vi | 34.3937 | 0.565974 | 32.6627 |
| back\_translation\_en-vi | 34.6334 | 0.529341 | 32.7045 |

Bảng 3.3 Thời gian huấn luyện của các mô hình

|  |  |  |
| --- | --- | --- |
| Mô hình | Training time | Số lượng dữ liệu huấn luyện |
| mBART50\_vi-en | 25669 | 133317 |
| mBART50\_en-vi | 25907 | 133317 |
| back\_translation\_en-vi | 29801 | 152469 |

Ta thấy kết quả giá trị Bleu của model dịch từ tiếng Việt sang Anh có giá trị cao nhất nên cho thấy model đang hoạt động tốt hơn khi dịch từ tiếng Anh sang tiếng Việt.

Giá trị Bleu của model khi sử dụng back translation để tăng cường dữ liệu có giá trị Bleu tốt hơn khi không sử dụng nên ta có thể thấy rằng khi ta có bộ dữ liệu ít thì có thể sử dụng phương pháp này để tăng cường dữ liệu sẽ làm mô hình được cải thiện thêm khá đáng kể.

CHƯƠNG 4: Xây dựng chương trình demo

4.1 Giới thiệu về các framework sử dụng

4.1.1 Flask

Flask là một Web Framework rất nhẹ của Python, dễ dàng giúp người mới bắt đầu học Python có thể tạo ra website nhỏ. Flask cũng dễ mở rộng để xây dựng các ứng dụng web phức tạp. Điểm nổi bật khi sử dụng Flask là dễ dàng triển khai mô hình trí tuệ nhân tạo nên Web để sử dụng, do đó nên rất được nhiều người sử dụng.



Hình 4.1 Mình họa thư viện Flask

Các tính năng của Flask Framework:

* Phát triển máy chủ
* Phát triển trình gỡ lỗi
* Khả năng tương thích công cụ dự trên ứng dụng Google
* Nhiều tiện ích mở rộng cho các tính năng mong muốn
* Hỗ trợ bảo mật cookie
* Cung cấp xử lý HTTP request
* API độc đáo và mạch lạc, hỗ trợ RESTful API
* Dễ dàng triển khai

Với mục tiêu xây dựng một Backend server không quá phức tạp nhưng phải xử lý được các yêu cầu của người dùng với tốc dộ nhanh, tôi quyết định nghiên cứu và sử dụng framework Flask.

4.1.2 Transformers



Hình 4.2 Minh họa thư viện Transformers

Transformers là một thư viện mã nguồn mở được phát triển bởi Hugging Face, cung cấp các mô hình ngôn ngữ mạnh mẽ và hiệu quả cho nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên (NLP) như:

* Tóm tắt văn bản
* Dịch máy
* Trả lời câu hỏi
* Tạo văn bản sáng tạo
* Phân loại văn bản

Điểm nổi bật của Transformers:

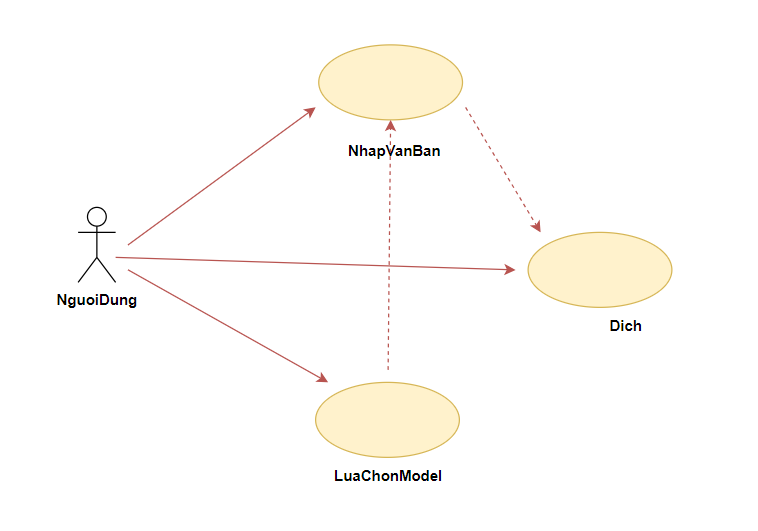
* Hiệu quả: Transformers sử dụng kiến trúc mạng nơ-ron tiên tiến như attention và encoder-decoder để xử lý ngôn ngữ hiệu quả hơn so với các mô hình truyền thống.
* Dễ sử dụng: Transformers cung cấp API đơn giản và dễ sử dụng, cho phép người dùng dễ dàng tích hợp các mô hình vào ứng dụng của họ.
* Đa dạng: Transformers cung cấp nhiều mô hình ngôn ngữ đã được đào tạo sẵn cho nhiều nhiệm vụ NLP khác nhau, giúp người dùng tiết kiệm thời gian và công sức.
* Cộng đồng: Transformers có cộng đồng người dùng và nhà phát triển lớn và tích cực, luôn sẵn sàng hỗ trợ và chia sẻ kiến thức.

Transformers là một công cụ mạnh mẽ và linh hoạt cho xử lý ngôn ngữ tự nhiên. Với hiệu quả, tính dễ sử dụng và đa dạng, Transformers đã trở thành lựa chọn phổ biến cho nhiều nhà phát triển và doanh nghiệp trong nhiều lĩnh vực khác nhau.

4.2 Phân tích hệ thống

Hệ thống demo cho bài toán trên là một chương trình có giao diện với chức năng chính là thực hiện dịch ngôn ngữ từ đoạn văn bản mà người dùng nhập vào. Ngoài ra, hệ thống còn có chức năng là lựa chọn ngôn ngữ nguồn và ngôn ngữ đích để dịch.

4.2.1 Biểu đồ use case tổng quát



Hình 4.3 Biểu đồ use case tổng quát

4.2.2 Mô tả chi tiết các use case

4.2.2.1 Use case Nhập văn bản

|  |
| --- |
| **Tên use case: Nhập văn bản** |
| **Mô tả tóm tắt:** Use case cho phép người dùng nhập văn bản muốn dịch vào hệ thống |
| **Luồng sự kiện**   * Luồng cơ bản:   1. Người dùng mở ứng dụng và điều hướng đến trang hoặc màn hình có ô nhập văn bản.  2. Hệ thống hiển thị ô hoặc khu vực nhập văn bản.  3. Người dùng nhấn vào ô hoặc khu vực nhập văn bản để bắt đầu nhập liệu.  4. Hệ thống kích hoạt bàn phím (nếu trên thiết bị di động) hoặc chuẩn bị sẵn sàng để nhận đầu vào từ bàn phím (trên máy tính).  5. Người dùng nhập văn bản vào ô hoặc khu vực nhập liệu.  6. Hệ thống hiển thị các ký tự mà người dùng nhập vào.  7. Người dùng hoàn tất việc nhập liệu. |
| **Các yêu cầu đặc biệt:** Không có |
| **Tiền điều kiện:** Không có |
| **Hậu điều kiện:** Không có |
| **Các điểm mở rộng:** Không có |

4.2.2.2 Use case Lựa chọn model

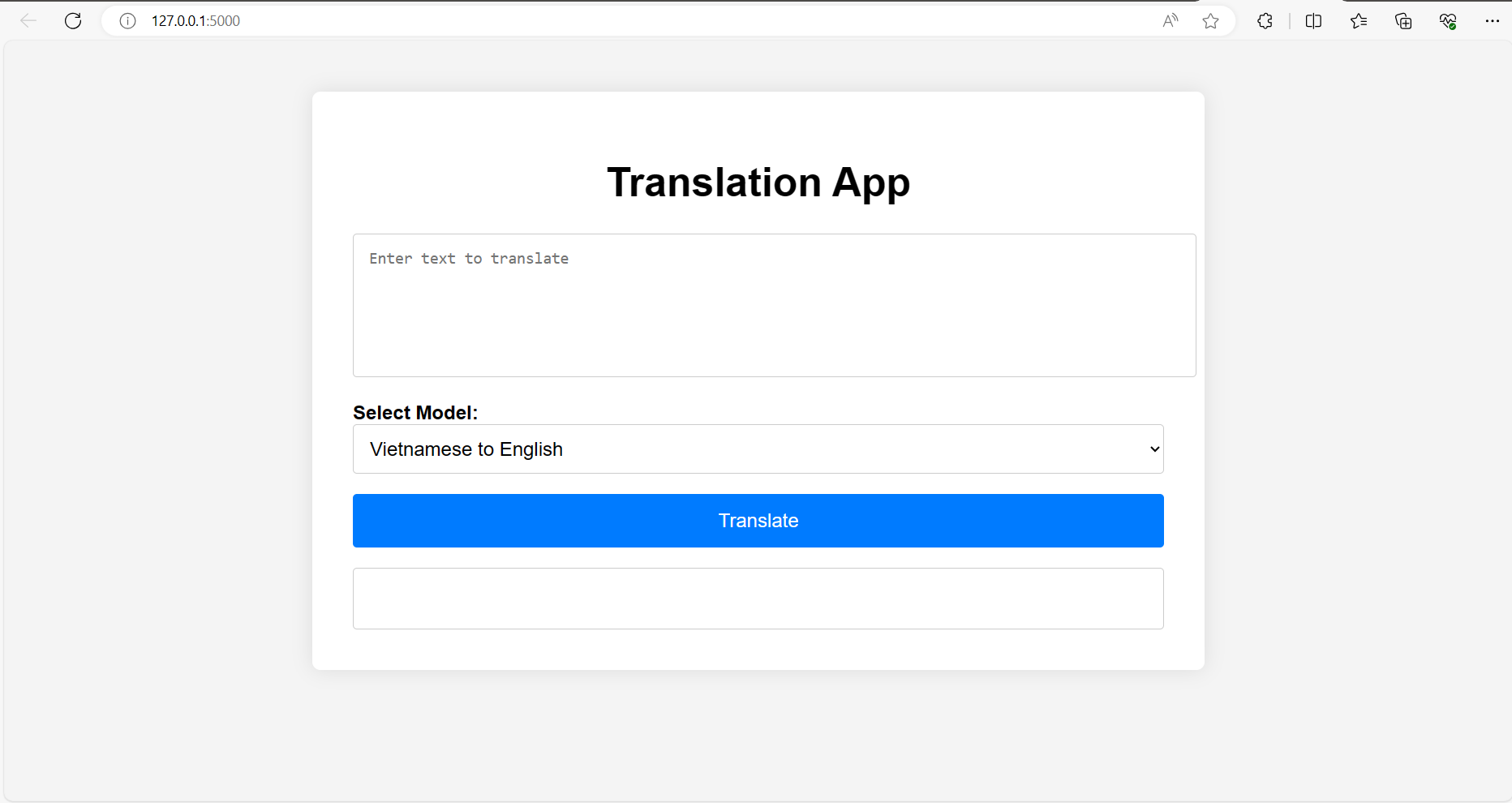
|  |
| --- |
| **Tên use case: Lựa chọn model** |
| **Mô tả tóm tắt:** Use case cho phép người dùng lựa chọn ngôn ngữ nguồn và ngôn ngữ đích |
| **Luồng sự kiện**   * Luồng cơ bản:   1. Người dùng điều hướng đến vùng có ô “Select Model”  2. Hệ thống hiển thị các lựa chọn để người dùng chọn ngôn ngữ nguồn và đích.  3. Người dùng nhấn vào ô hoặc khu vực hiển thị lựa chọn.  4. Người dùng hoàn tất việc lựa chọn. |
| **Các yêu cầu đặc biệt:** Không có |
| **Tiền điều kiện:** Không có |
| **Hậu điều kiện:** Không có |
| **Các điểm mở rộng:** Không có |

4.2.2.3 Use case Dich

|  |
| --- |
| **Tên use case: Dịch** |
| **Mô tả tóm tắt:** Use case cho phép người dùng dịch ngôn ngữ văn bản mà người dùng đã nhập |
| **Luồng sự kiện**   * Luồng cơ bản:   1. Người dùng nhấn nút "Translate".  2. Hệ thống xử lý và dịch văn bản từ ngôn ngữ nguồn sang ngôn ngữ đích.  3. Hệ thống hiển thị văn bản đã dịch bằng ngôn ngữ đích cho người dùng |
| **Các yêu cầu đặc biệt:** Không có |
| **Tiền điều kiện:** Không có |
| **Hậu điều kiện:** Không có |
| **Các điểm mở rộng:** Không có |

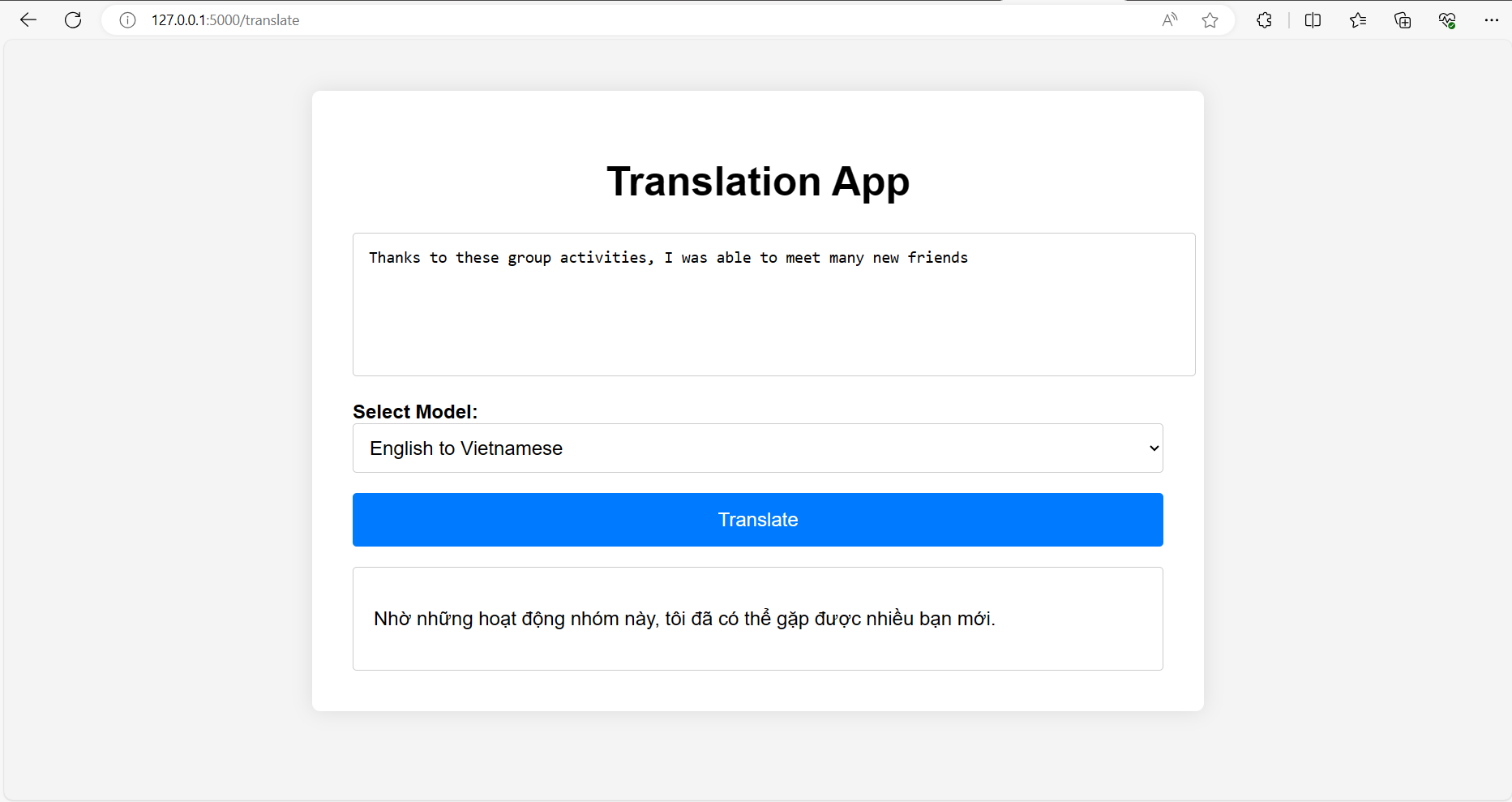
4.3 Giao diện hệ thống

Giao diện được thiết kế đơn giản với tông màu trắng, thao tác đơn giản và dễ sử dụng, tương lai hệ thống sẽ được nâng cấp giao diện và thêm các chức năng phù hợp với người dùng. Chi tiết giao diện được trình bày qua các hình bên dưới.



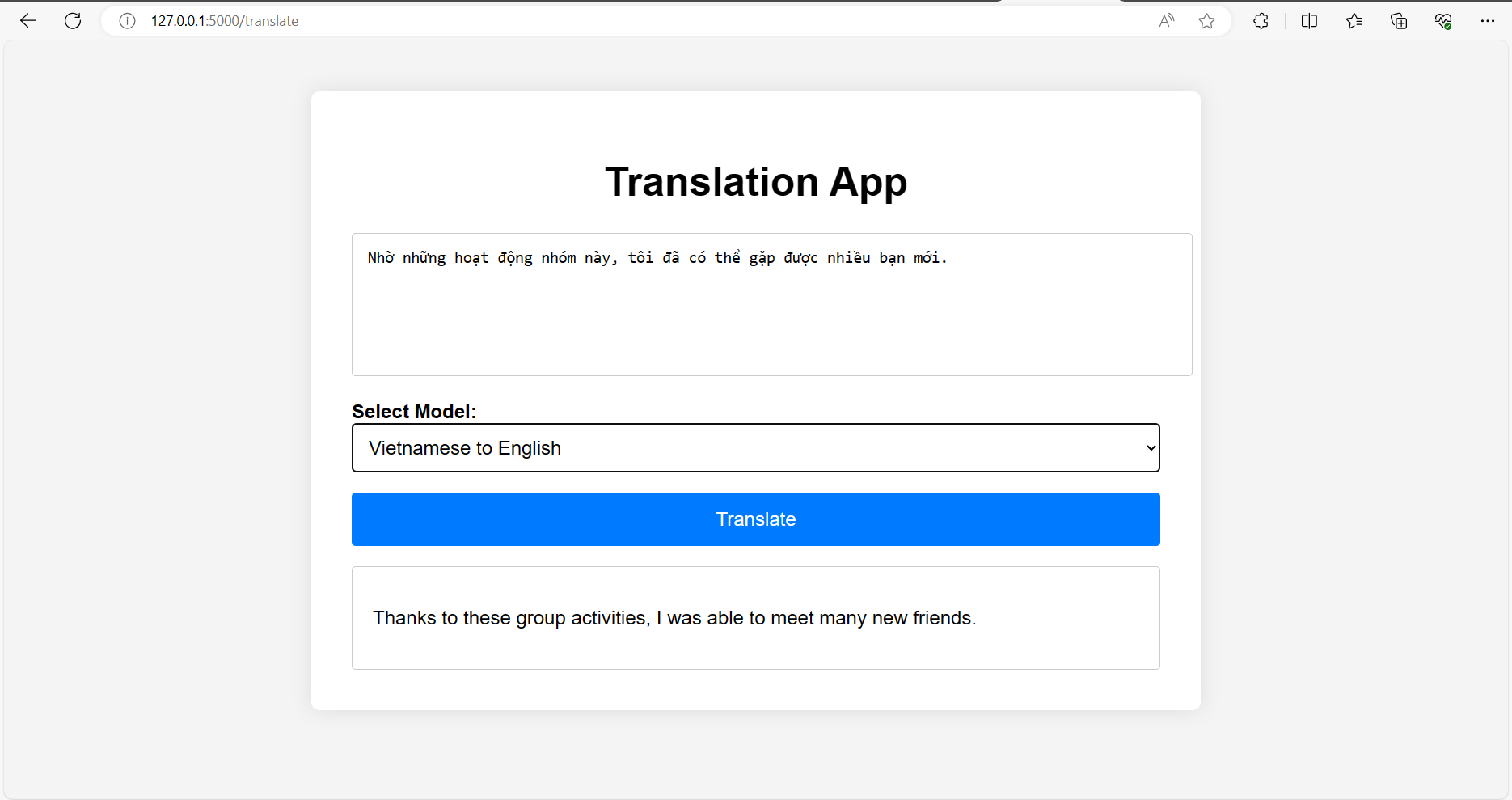
Hình 4.4 Giao diện mặc định khi khởi động

Đây là giao diện khi màn hình chương trình khi được khởi động, ta có thể lựa chọn ngôn ngữ để dịch.



Hình 4.5 Giao diện khi thực hiện dịch sang tiếng Việt

Đây là giao diện khi sử dụng dịch từ tiếng Anh sang tiếng Việt.



Hình 4.6 Giao diện khi thực hiện dịch sang tiếng Anh

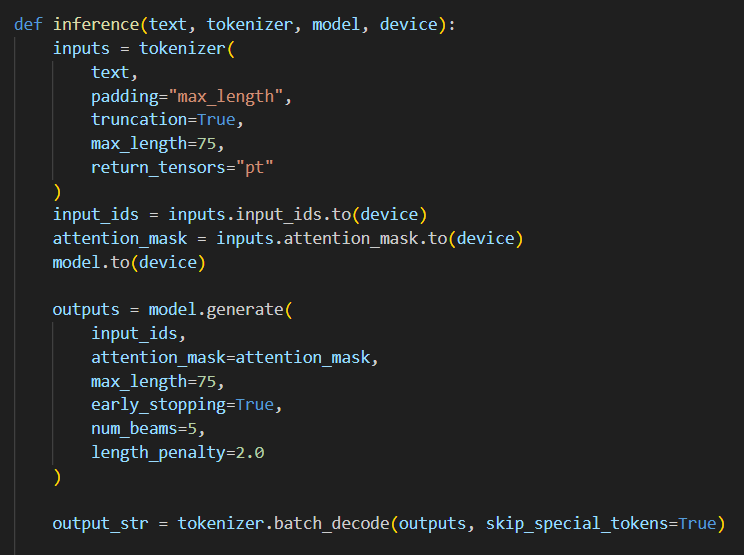
Đây là giao diện khi sử dụng dịch từ tiếng Việt sang tiếng Anh.

4.4 Các chức năng của hệ thống

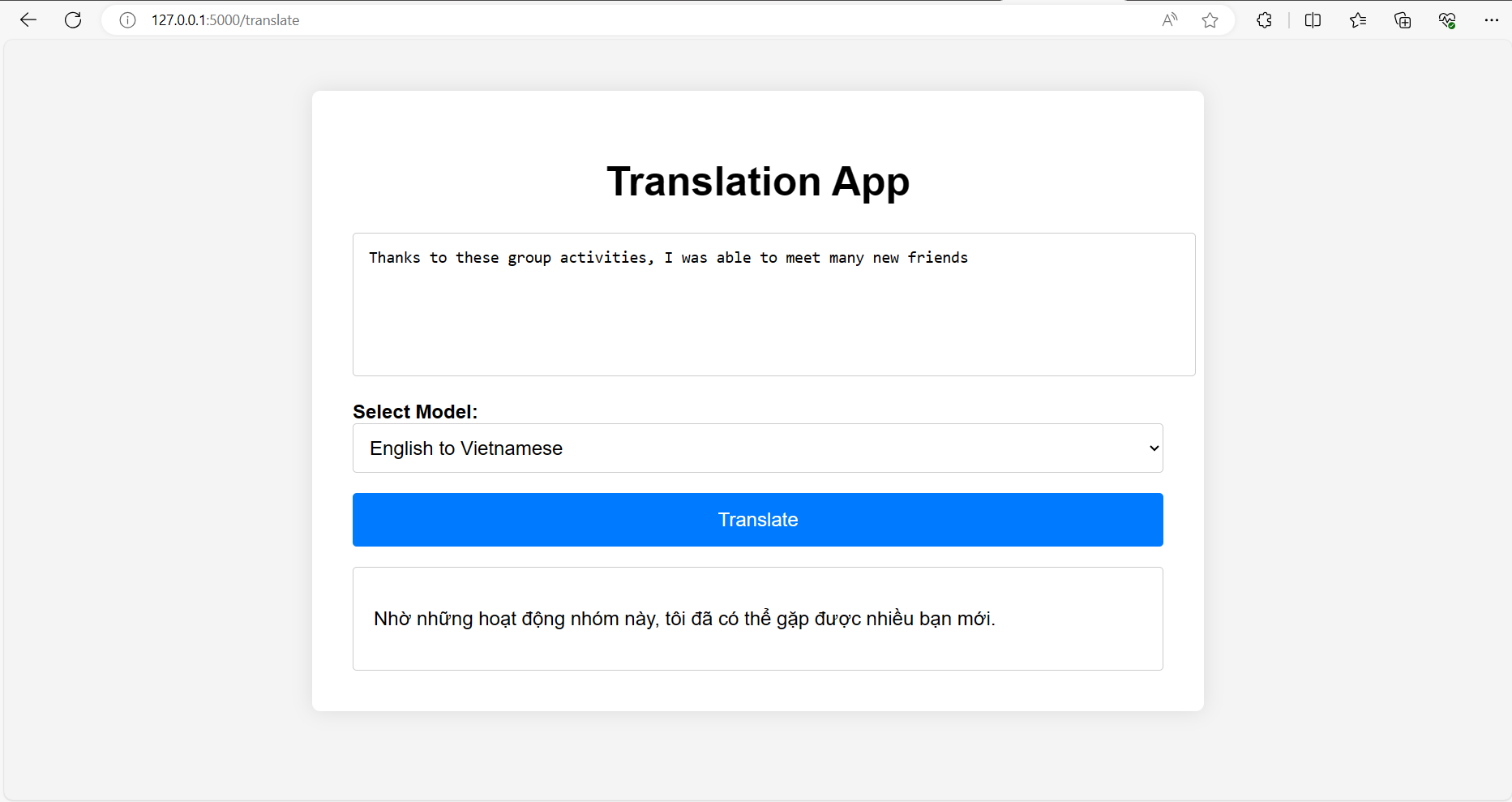
4.4.1 Dịch văn bản từ tiếng Anh sang tiếng Việt

Đây là chức năng dịch văn bản từ tiếng Anh sang tiếng Việt, chức năng này sử dụng model mBART50 Back Translation, hệ thống sẽ sử dụng phương thức POST để lấy văn bản người dùng nhập vào rồi đưa về hệ thống để xử lý.

Dưới đây là hàm xử lý có tác dụng dịch văn bản khi đã được nạp mô hình:



Dưới đây là kết quả khi sử dụng mô hình dịch từ Anh sang Việt:

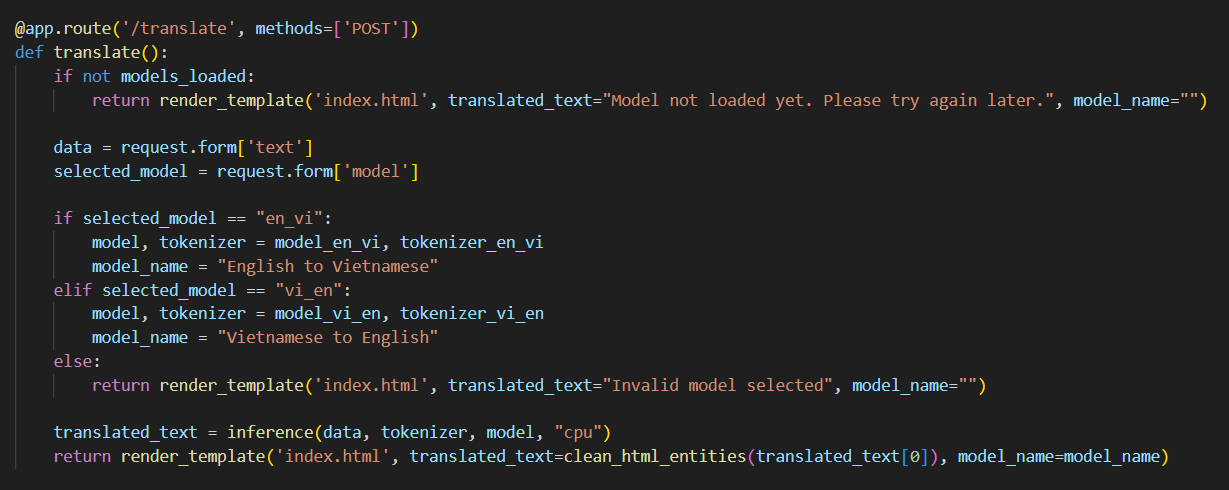


Hình 4.7 Giao diện khi thực hiện dịch sang tiếng Việt

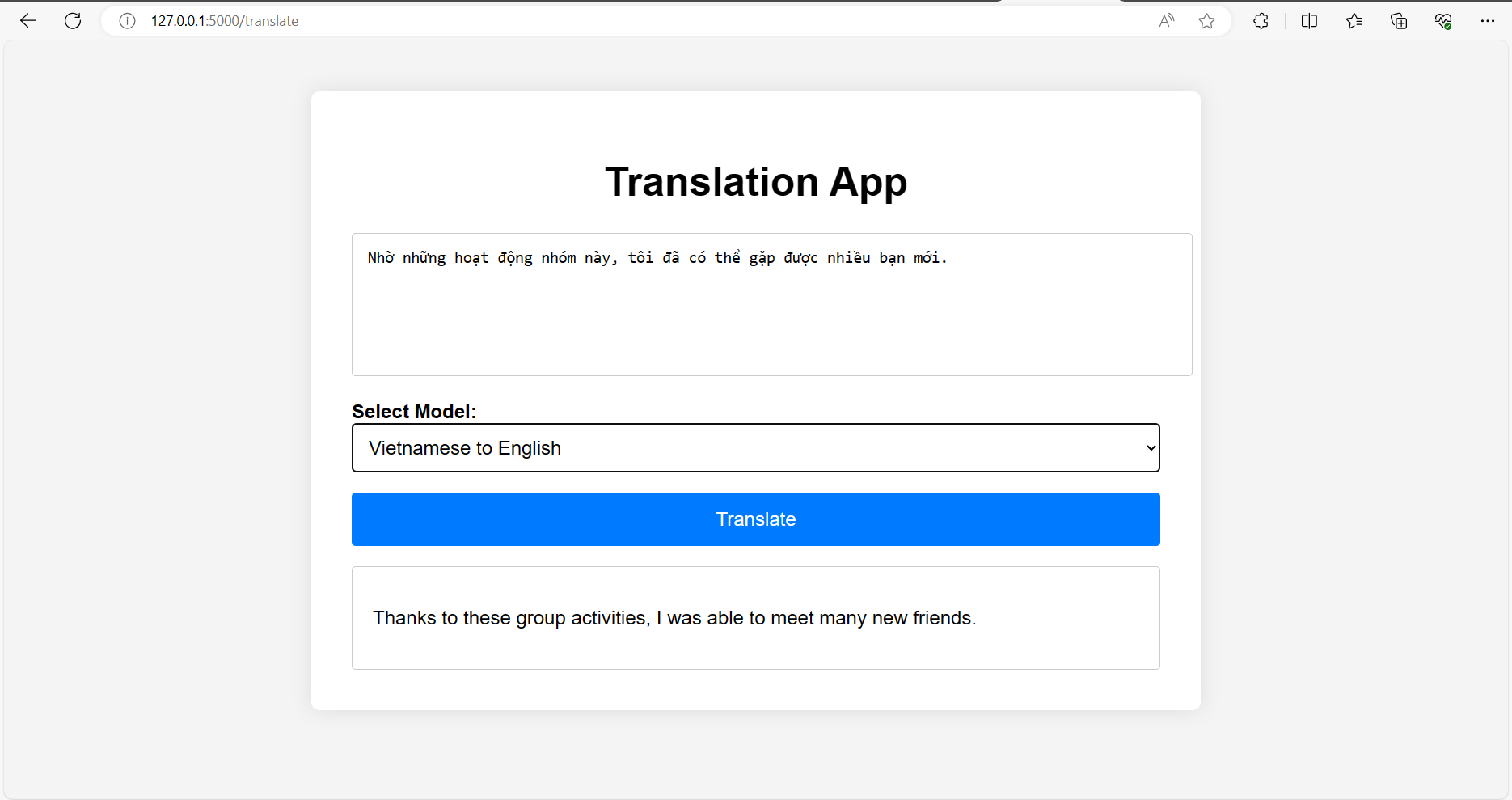
4.4.2 Dịch văn bản từ tiếng Việt sang tiếng Anh

Cách hoạt động của chức năng dịch từ Việt sang Anh cũng tương tự chức năng trên, chỉ khác là mô hình sử dụng là mô hình mBART50 dịch từ Việt sang Anh và khi sử dụng lựa chọn model thì hệ thống sẽ tải mô hình đúng với lựa chọn đấy.

Dưới đây là hàm translate sử dụng phương thức POST để sử lý văn bản người dùng nhập:



Dưới đây là kết quả khi sử dụng mô hình dịch từ Việt sang Anh:



Hình 4.8 Giao diện khi thực hiện dịch sang tiếng Anh

KẾT LUẬN

Thời gian thực hiện đồ án tốt nghiệp vừa qua đã mang lại cho em nhiều kỹ năng và kiến thức bổ ích. Trong suốt quá trình này, em đã nghiên cứu và áp dụng nhiều kỹ thuật khác nhau của trí tuệ nhân tạo để giải quyết bài toán dịch máy, qua đó không chỉ mở rộng kiến thức chuyên môn mà còn phát triển các kỹ năng làm việc độc lập và quản lý thời gian hiệu quả. Những kinh nghiệm này chắc chắn sẽ là nền tảng vững chắc cho sự nghiệp của em trong tương lai.

Em đã tìm hiểu và ứng dụng mô hình học sâu cũng như khai thác các nghiên cứu đã được công bố để hoàn thiện đề tài “Nghiên cứu áp dụng Deep Learning trong bài toán dịch máy”. Em cũng đã tự xây dựng và thử nghiệm bộ dữ liệu dành cho bài toán này. Dưới sự hướng dẫn của thầy Trần Hùng Cường, em đã tìm ra các kỹ thuật để những bản dịch càng chuẩn xác hơn và đã xây dựng được một chương trình dịch máy đơn giản.

Chương trình đã đạt được những kết quả tích cực với độ chính xác cao. Tuy nhiên, do hạn chế về tài nguyên và thời gian nên hệ thống mới chỉ xử lý tốt một số ngôn ngữ nhất định, điều này vẫn chưa đủ để hệ thống trở thành một công cụ dịch thuật hoàn chỉnh. Hơn nữa, hệ thống vẫn còn hạn chế về tốc độ và thời gian xử lý. Do đó, em rất mong nhận được những ý kiến đóng góp từ các thầy cô để cải thiện hệ thống, hướng tới phát triển một ứng dụng dịch máy thương mại trong tương lai.

Em xin gửi lời cảm ơn chân thành đến thầy giáo, Tiến sĩ Trần Hùng Cường, vì sự hướng dẫn tận tình và sự hỗ trợ quỹ báu trong quá trình thực hiện đề tài. Em kính chúc thầy luôn dồi dào sức khỏe và đạt được nhiều thành công trong những nghiên cứu sắp tới.

Em xin trân trọng cảm ơn!

Nguyễn Lương Nam Anh

TÀI LIỆU THAM KHẢO

[1] “Speech and Language Processing” (3rd Edition) by Daniel Jurafsky and James H. Martin

[2] “Foundations of Statistical Natural Language Processing” by Christopher D. Manning and Hinrich Schütze

[3] Steven Bird, Ewan Klein, and Edward Loper (2009). “Natural Language Processing with Python”.

[4] Lewis, M., Liu, Y., Ifeanyi, N., Wu, S., Ott, M., Lee, J., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461

[5] Radford, A., Narasimhan, K., Srinivasan, P., Chung, I., & Polosukhin, I. (2018). Improving language understanding by generative pre-training. <https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf>

[6] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 14276-14289. https://arxiv.org/abs/2005.14165

[7] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), Volume 1 (Long Papers), 4171-4186. <https://aclanthology.org/N19-1423>

[8] Hướng dẫn xử lý ngôn ngữ tự nhiên: NLP là gì? Ví dụ <https://www.guru99.com/vi/nlp-tutorial.html>

[9] BART: Sự kết hợp giữa BERT và GPT. <https://trituenhantao.io/kien-thuc/bart-su-ket-hop-giua-bert-va-gpt/>

[10] Tìm hiểu về BLEU và WER - Metric cho 1 số tác vụ trong NLP (<https://viblo.asia/p/tim-hieu-ve-bleu-va-wer-metric-cho-1-so-tac-vu-trong-nlp-Eb85oA16Z2G>)