

TRƯỜNG ĐẠI HỌC PHENIKAA
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN

**Đề tài: Xây dựng mô hình dịch máy sử dụng kiến trúc
Transformer**

Học Phần: Xử lý ngôn ngữ tự nhiên-1-1-24(N01)
Giảng Viên: TS. Phạm Tiến Lâm
Ths. Nguyễn Văn Sơn
Sinh viên: Nguyễn Thái Đức - 22014381
Đặng Minh Trí - 22010157

Hà Nội, Tháng 11 Năm 2024

Mục lục

1	Lời giới thiệu	4
1.1	Xử lý ngôn ngữ tự nhiên và tầm quan trọng	4
1.2	Nội dung của bài tập lớn	4
1.3	Mục tiêu bài tập lớn	5
2	Cơ sở lý thuyết	6
2.1	Tiền xử lý ngôn ngữ tự nhiên	6
2.1.1	Ngôn ngữ tự nhiên là gì?	6
2.1.2	Quy trình xử lý ngôn ngữ tự nhiên	6
2.1.3	Vectorization	8
2.2	Mô hình Transformer	9
2.2.1	Chuẩn bị dữ liệu đầu vào cho khối Encoder	10
2.2.2	Quá trình encoder	11
2.2.3	Chuẩn bị cho quá trình decoder	14
2.2.4	Quá trình decoder	14
3	Triển khai mô hình, khảo nghiệm độ chính xác của mô hình	17
3.1	Bài toán	17
3.2	Tổng quan về dữ liệu	17
3.2.1	Mô tả chung về dữ liệu	17
3.2.2	Tiền xử lý dữ liệu	19
3.2.3	Look up table	19
3.3	Triển khai mô hình	20
4	Triển khai mô hình chạy trên thời gian thực	22
4.1	Đánh giá mô hình	23
4.2	Phát triển trong tương lai	24
5	Tổng Kết	25

Danh sách hình vẽ

2.1	Quy trình của một mô hình xử lý ngôn ngữ tự nhiên thông thường	6
2.2	Sơ đồ hoạt động của mô hình Transformer	10
2.3	Cấu trúc của encoder	11
2.4	Mô tả quá trình thực hiện linear và softmax	16
3.1	File "train.en"	18
3.2	File "train.vi"	18
3.3	File "tst2013.en"	18
3.4	File "tst2013.vi"	19
3.5	Quá trình xử lý dữ liệu	19
3.6	Kết quả khi sau khi train với epochs=30	20
4.1	Giao diện cơ bản	23
4.2	Chất lượng bản dịch	23

Chương 1

Lời giới thiệu

1.1 Xử lý ngôn ngữ tự nhiên và tầm quan trọng

Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) là một lĩnh vực của trí tuệ nhân tạo (AI) chuyên nghiên cứu và phát triển các phương pháp để máy tính có thể hiểu, diễn giải, và tạo ra ngôn ngữ của con người. NLP bao gồm nhiều kỹ thuật khác nhau, từ phân tích ngữ pháp, dịch máy, tóm tắt văn bản, đến nhận dạng ngữ cảnh và ý nghĩa của từ. Tầm quan trọng của NLP ngày càng gia tăng trong thời đại số hóa, nơi mà thông tin và dữ liệu ngôn ngữ xuất hiện liên tục trên các nền tảng trực tuyến.

NLP giúp máy tính giao tiếp với con người một cách tự nhiên và dễ dàng hơn, hỗ trợ tự động hóa các tác vụ như dịch văn bản, tìm kiếm thông tin, và phân loại nội dung. Điều này mang lại lợi ích lớn cho các ngành như giáo dục, y tế, và tài chính, đồng thời tạo điều kiện cho sự phát triển của các ứng dụng trợ lý ảo, chatbot, và công cụ phân tích dữ liệu. Nhờ NLP, việc khai thác tri thức từ dữ liệu ngôn ngữ ngày càng hiệu quả, từ đó thúc đẩy sự phát triển của các giải pháp thông minh và nâng cao trải nghiệm người dùng.

1.2 Nội dung của bài tập lớn

Ý tưởng xây dựng một ứng dụng dịch máy là tạo ra một công cụ giúp chuyển đổi ngôn ngữ tự động từ tiếng Anh sang tiếng Việt, nhằm phá bỏ rào cản ngôn ngữ và hỗ trợ người dùng trong giao tiếp quốc tế. Ứng dụng này sẽ tận dụng sức mạnh của mô hình Transformer, vốn đã được chứng minh về độ chính xác và khả năng duy trì ngữ cảnh khi dịch. Với khả năng dịch nhanh và hiệu quả, ứng dụng dịch máy có thể được triển khai rộng rãi trong nhiều lĩnh vực thực tiễn như giáo dục, thương mại, du lịch và truyền thông.

Ứng dụng được sử dụng rộng rãi trong nhiều ngành nghề. Trong giáo dục, ứng dụng sẽ giúp sinh viên và người học tiếp cận tài liệu tiếng nước ngoài dễ dàng hơn. Trong thương mại, nó có thể hỗ trợ các doanh nghiệp giao tiếp và trao đổi

thông tin với đối tác toàn cầu. Ngoài ra, ứng dụng còn có thể là công cụ đắc lực cho những ai yêu thích du lịch, giúp họ vượt qua rào cản ngôn ngữ khi khám phá các nền văn hóa mới.

1.3 Mục tiêu bài tập lớn

- **Phân tích ứng dụng thực tiễn:** Áp dụng các thuật toán đã học vào các bài toán thực tế, từ đó đánh giá hiệu quả và tiềm năng của chúng.
- **Phát triển kỹ năng lập trình và phân tích dữ liệu:** Tăng cường kỹ năng lập trình, giải quyết bài toán liên quan đến xử lý ngôn ngữ tự nhiên
- **Tạo ra được sản phẩm có tính ứng dụng:** Tạo ra sản phẩm giúp chuyển đổi tiếng anh và tiếng việt giúp mọi người dễ dàng hơn trong việc tiếp xúc với các tài liệu tiếng anh.

Chương 2

Cơ sở lý thuyết

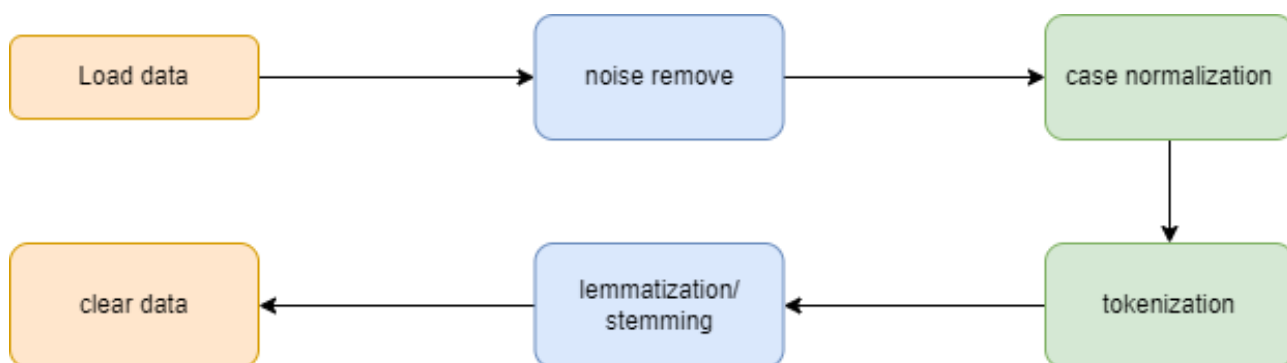
2.1 Tiền xử lý ngôn ngữ tự nhiên

2.1.1 Ngôn ngữ tự nhiên là gì?

Ngôn ngữ tự nhiên trong khoa học máy tính (Natural Language Processing - NLP) là một lĩnh vực nghiên cứu và ứng dụng của trí tuệ nhân tạo (AI) và khoa học máy tính. NLP tập trung vào sự tương tác giữa máy tính và ngôn ngữ của con người, nhằm mục tiêu giúp máy tính hiểu, giải thích và tạo ra văn bản hoặc lời nói bằng ngôn ngữ tự nhiên.

2.1.2 Quy trình xử lý ngôn ngữ tự nhiên

Quá trình tiền xử lý dữ liệu (data preprocessing) trong xử lý ngôn ngữ tự nhiên (NLP) là một bước quan trọng nhằm chuẩn bị dữ liệu thô cho các mô hình học máy và các thuật toán NLP. Dưới đây là các bước phổ biến trong quá trình tiền xử lý dữ liệu:



Hình 2.1: Quy trình của một mô hình xử lý ngôn ngữ tự nhiên thông thường

Noise Removal

Noise Removal là loại bỏ các ký tự đặc biệt, dấu câu, số, và các ký tự không cần thiết khác mà không đóng góp ý nghĩa cho bài toán.

Các ký tự không cần thiết như dấu câu, số, hoặc các ký tự đặc biệt không mang nhiều thông tin ngữ nghĩa và có thể làm "nhiều" dữ liệu. Việc loại bỏ chúng giúp mô hình tập trung vào các từ và cụm từ quan trọng hơn, từ đó cải thiện độ chính xác, đồng thời khi việc xóa bỏ các ký tự không cần thiết này giúp cho mô hình hoạt động nhanh chóng hơn khi không cần phải xử lý chúng.

Các thuật toán NLP như tokenization, stemming, lemmatization hoạt động hiệu quả hơn khi dữ liệu đã được loại bỏ các ký tự không cần thiết. Điều này giúp giảm tải tính toán và tăng tốc độ xử lý.

Case Normalization

Do chữ hoa và chữ thường được máy móc hiểu là hai ký tự hoàn toàn khác nhau nhưng với ngôn ngữ tự nhiên thì chúng được hiểu với nghĩa là như nhau. Vì vậy, việc chuẩn hóa chúng về cùng một dạng chữ in hoa hoặc chữ thường là điều vô cùng quan trọng để đảm bảo tính đồng nhất khi đưa vào mô hình học máy.

Tokenization

Tokenization là một bước quan trọng trong quá trình tiền xử lý dữ liệu trong xử lý ngôn ngữ tự nhiên, nó giúp phân chia văn bản thành các đơn vị nhỏ hơn, thường là từ hoặc câu.

VD: "I love machine learning "

Sau khi qua mô hình tokenization ta sẽ được kết quả như sau: ["I", "love", "machine", "learning"]

Tokenization chia văn bản thành các đơn vị nhỏ hơn gọi là token (thường là từ hoặc câu). Điều này giúp mô hình NLP xử lý văn bản một cách dễ dàng và có hệ thống hơn. Cùng với đó các token nhỏ hơn giúp mô hình hiểu rõ hơn về ngữ nghĩa và ngữ cảnh của từng phần trong văn bản.

Tokenization là bước tiền đề cho nhiều kỹ thuật NLP khác như stemming, lemmatization, part-of-speech tagging, và named entity recognition. Mỗi từ sau khi tokenization sẽ được xử lý riêng lẻ trong các bước tiếp theo này.

Tóm lại, tokenization là một bước không thể thiếu trong quá trình xử lý ngôn ngữ tự nhiên, giúp chuyển đổi văn bản thành các đơn vị dễ xử lý, duy trì ngữ nghĩa và ngữ cảnh, và tạo tiền đề cho các bước xử lý tiếp theo.

Stop words removal

Stop words removal là xem xét tầm quan trọng của từng từ trong một câu nhất định. Trong tiếng Anh, một số từ xuất hiện thường xuyên hơn những từ

khác như “is”, “a”, “the”, “and”. Vì chúng xuất hiện thường xuyên nên quy trình NLP gán cờ chúng là các từ dừng. Chúng được lọc ra để tập trung vào những từ quan trọng hơn. quá trình này giúp loại bỏ bớt những dữ liệu không cần thiết giúp giảm thời gian chạy của mô hình mà không gây ảnh hưởng đến kết quả sau cùng

Stemming và Lemmatization

Stemming: Chuyển các từ về dạng gốc của chúng bằng cách loại bỏ các hậu tố.

Lemmatization: Chuyển các từ về dạng cơ bản của chúng dựa trên từ điển

VD: Playes -> play

Stemming và Lemmatization là hai kỹ thuật quan trọng trong xử lý ngôn ngữ tự nhiên (NLP) nhằm chuẩn hóa từ vựng, giúp các mô hình và thuật toán NLP hoạt động hiệu quả hơn.

Tác dụng của stemming:

- Stemming giúp giảm các biến thể của một từ về cùng một gốc từ, bất kể dạng từ của chúng trong câu. Ví dụ, "running", "runner", và "ran" đều có thể được giảm về gốc từ "run".
- Bằng cách giảm các từ về gốc từ, stemming giúp giảm kích thước từ vựng, từ đó tiết kiệm bộ nhớ và tài nguyên tính toán.
- Khi các biến thể của từ được quy về một gốc từ, các mô hình học máy và NLP có thể học từ dữ liệu một cách nhất quán và hiệu quả hơn.

Tác dụng của Lemmatization:

- Lemmatization chuyển các từ về dạng cơ bản hoặc dạng từ điển của chúng, giúp giữ nguyên ý nghĩa ngữ pháp của từ trong câu. Ví dụ, "am", "is", "are" đều được chuyển về "be".
- Khác với stemming, lemmatization không chỉ cắt đuôi từ mà còn sử dụng từ điển ngôn ngữ để đảm bảo từ gốc có nghĩa và phù hợp ngữ cảnh. Điều này giúp duy trì ngữ nghĩa và ngữ pháp của văn bản.
- Bằng cách chuẩn hóa từ về dạng cơ bản nhưng vẫn giữ ngữ nghĩa, lemmatization giúp mô hình NLP hoạt động chính xác hơn, đặc biệt trong các bài toán phân tích ngữ nghĩa và phân loại văn bản.

2.1.3 Vectorization

Các mô hình ngôn ngữ tự nhiên về bản chất là các mô hình toán học. Vì vậy, nó không thể hoạt động khi dữ liệu đưa vào là dạng chữ. Cho nên việc chuyển dạng từ về vector là rất cần thiết trước khi đưa chúng vào mô hình ngôn ngữ tự

nhiên.

Một vài mô hình chuyển đổi sang vector:

- Bag of Words (BoW): là một phương pháp đơn giản để vector hóa văn bản bằng cách biến mỗi văn bản thành một vector dựa trên sự xuất hiện của các từ trong tài liệu.
- TF-IDF (Term Frequency-Inverse Document Frequency): là một phương pháp cải tiến từ Bag of Words nhằm đánh giá tầm quan trọng của các từ trong tài liệu bằng cách điều chỉnh tần suất của từ theo mức độ phổ biến của nó trong tập hợp tài liệu.
- Word Embeddings (Word2Vec, GloVe, FastText): là các phương pháp hiện đại sử dụng mô hình học sâu để biểu diễn các từ dưới dạng các vector số trong không gian nhiều chiều. Các từ có ngữ nghĩa tương tự nhau sẽ có các vector gần nhau trong không gian này.

So sánh và lựa chọn phương pháp:

- BoW và TF-IDF phù hợp cho các bài toán đơn giản, như phân loại văn bản và phân tích sơ bộ dữ liệu văn bản.
- Word Embeddings phù hợp cho các bài toán phức tạp hơn, yêu cầu hiểu biết ngữ nghĩa và ngữ cảnh, như dịch máy, phân tích ngữ nghĩa và xây dựng các mô hình ngôn ngữ mạnh mẽ.

Ở bài tập lớn này, thay vì việc sử dụng thuật toán phức tạp như word embedding, GloVe ta sử dụng một cách thức đơn giản hơn chính là sử dụng bảng tra cứu (look up table).

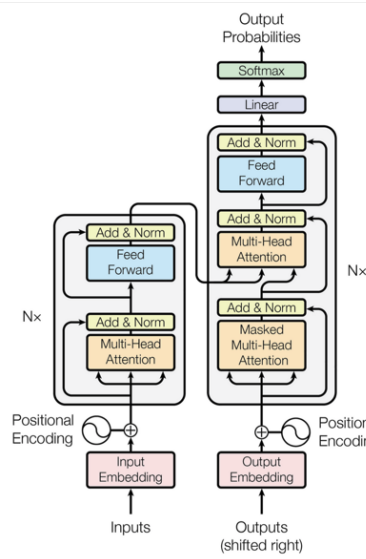
Look up table hoạt động như sau:

- Khởi tạo ngẫu nhiên các embedding vector cho mỗi từ trong tập vocab
- Trong quá trình huấn luyện, ta cập nhật các giá trị embedding vector thông qua lan truyền ngược cùng với các ma trận trọng số khác trong mô hình.

2.2 Mô hình Transformer

Transformer là một mô hình mạng nơ-ron được giới thiệu bởi các nhà nghiên cứu của Google trong bài báo nổi tiếng "Attention Is All You Need" vào năm 2017. Đây là một mô hình tiên tiến trong xử lý ngôn ngữ tự nhiên (NLP), được thiết kế để xử lý hiệu quả các chuỗi dữ liệu tuần tự như văn bản, mà không cần dùng đến các kiến trúc truyền thống như mạng nơ-ron hồi quy (Recurrent Neural Networks - RNNs). Thay vào đó, Transformer sử dụng một cơ chế mới gọi là Attention (chính xác hơn là Self-Attention) để học cách tập trung vào các phần quan trọng của câu và duy trì ngữ cảnh của từ trong một đoạn văn bản dài.

Mô hình Transformer gồm hai thành phần chính là encoder và decoder. Thông thường, một mô hình chuẩn của transformer (được xây dựng trong bài báo "attention is all you need") có 6 lớp encoder và 6 lớp decoder.



Hình 2.2: Sơ đồ hoạt động của mô hình Transformer

2.2.1 Chuẩn bị dữ liệu đầu vào cho khối Encoder

Positional Encodeing

Để xử lý vấn đề toàn bộ dữ liệu đi vào mạng cùng một lúc, vì vậy cần một cơ chế để note lại vị trí các từ ở trong câu. Ta tạo ra một vector positional embedding (PE) để tham gia vào bài toán. Vector PE có kích thước bằng với kích thước của word embedding.

$$p_{i,j} = \begin{cases} \sin\left(\frac{pos}{10000^{\frac{j}{d}}}\right) & \text{nếu } j \text{ là chẵn} \\ \cos\left(\frac{pos}{10000^{\frac{j-1}{d}}}\right) & \text{nếu } j \text{ là lẻ} \end{cases}$$

Trong đó :

- pos: vị trí của từ trong câu
- j vị trí trong vector PE
- d: là kích thước của vector embedding

Sau khi thực hiện quá trình tính ra các vector PE. ta tiến hành cộng vector PE và vector từ ban đầu thu được kết quả chứa cả nghĩa của từ và vị trí của từ:

$$\vec{x}' = \vec{x} + \overrightarrow{PE}$$

Trong đó:

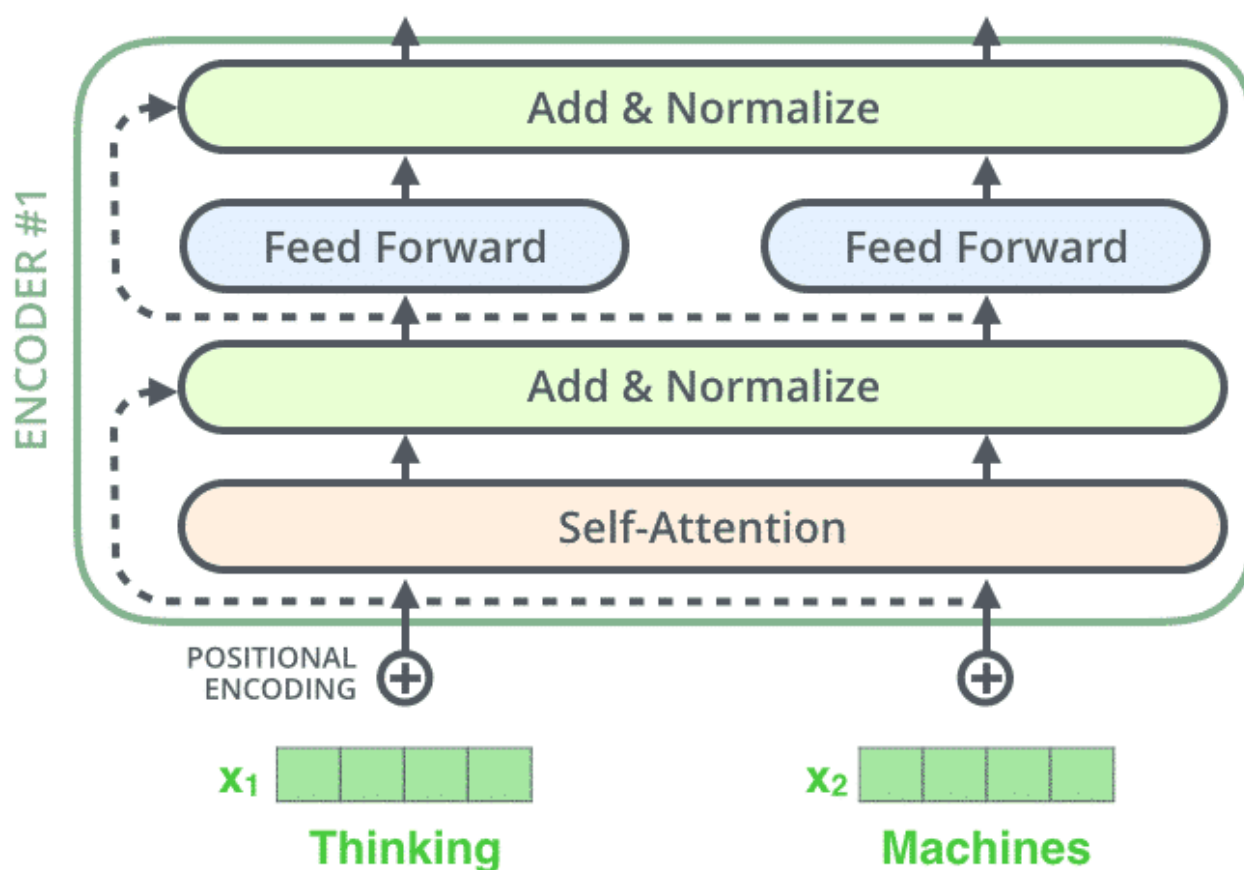
- x' : đầu vào của encoder
- x : word embedding ban đầu
- PE: vector tạo ra từ quá trình positional encoding

2.2.2 Quá trình encoder

Self-attention

Self-attention là nền tảng của multi-head attention. Self-attention cho phép mỗi từ trong chuỗi đầu vào "chú ý" đến các từ khác trong cùng một chuỗi để tự động tính toán các mối liên hệ ngữ nghĩa, giúp mô hình hiểu ngữ cảnh toàn cục. Nó là cơ chế cốt lõi để tính toán sự liên quan giữa các từ, từ đó giúp biểu diễn từ theo cách có ngữ cảnh.

Khi được mã hóa, kết quả của self attention sẽ mang thêm thông tin của các từ liên quan.



Hình 2.3: Cấu trúc của encoder

Quá trình của một self attention diễn ra như sau:

- Bước 1 khởi tạo trọng số: Trước khi bắt đầu huấn luyện, các ma trận trọng số W_q , W_k , và W_v thường được khởi tạo ngẫu nhiên. Các trọng số này sẽ học được trong suốt quá trình huấn luyện.
- Bước 2 tính Toán Các Đầu Vào: Đầu vào cho mô hình Transformer là một chuỗi các vector từ (embedding vectors) của các từ trong câu. Mỗi từ được chuyển đổi thành ba vector khác nhau:
 - **Query (Q)**: Tính bằng $Q = X \cdot W_q$
 - **Key (K)**: Tính bằng $K = X \cdot W_k$
 - **Value (V)**: Tính bằng $V = X \cdot W_v$

Ở đây, X là ma trận đầu vào.

- Bước 3 tính Attention Score: Các scores của attention được tính bằng cách sử dụng Q và K :

$$\text{Attention Scores} = \frac{Q \cdot K^T}{\sqrt{d_k}}$$

trong đó d_k là kích thước của vector khóa.

- Bước 5 Softmax và Trọng số Attention: Kết quả từ bước trước được áp dụng hàm softmax để tạo ra các trọng số attention:

$$\text{Attention Weights} = \text{softmax}(\text{Attention Scores})$$

$$\text{Attention output} = \sum_{i=1}^n \text{AttentionWeights}_i V_i$$

- Bước 6 Tính Loss: Mô hình sẽ so sánh output đầu ra với nhãn thực tế (ground truth) để tính toán loss (thường sử dụng hàm mất mát như cross-entropy).
- Bước 7 Cập nhật trọng số: Sử dụng thuật toán tối ưu như Adam hoặc SGD, các trọng số W_q , W_k , và W_v được cập nhật thông qua quá trình backpropagation. Các gradient được tính toán và sau đó được sử dụng để điều chỉnh các ma trận trọng số:

$$W \leftarrow W - \eta \cdot \frac{\partial \text{Loss}}{\partial W}$$

trong đó η là tốc độ học (learning rate).

Quá trình từ bước 2 đến bước 7 được lặp lại cho đến khi các ma trận trọng số hội tụ.

Multi-head Attention

Bản chất của multi-head attention là việc thực hiện nhiều self-attention, từ đó tạo ra nhiều các ma trận trọng số khác nhau. Qua đó việc hiểu ngữ cảnh có nhiều góc nhìn khác nhau. Kết quả của mỗi self-attention là một ma trận Z_i khác nhau. Một multi-head attention thông thường có chứa 8 self-attention.

Quá trình thực hiện multi-head attention diễn ra như sau (giả sử như số lượng head giống như số lượn được đề cập trong paper xuất bản của thuật toán (Attention is all you need)):

- Bước 1 Khởi tạo một ma trận trọng số W_0 có kích thước phụ thuộc vào output của các self-attention. Giả sử kích thước của các self-attention là $m \times n$ thì kích thước của ma trận trọng số W_0 là $8n \times n$
- Bước 2 Kết hợp các head: : Sau khi tính toán xong các output cho tất cả các head, mô hình sẽ kết hợp các output này lại thành một ma trận duy nhất bằng cách nối dài chúng lại từ mỗi head có kích thước $m \times n$ tạo thành ma trận mới có kích thước $m \times 8n$.
- Bước 3 Đưa ra kết quả: Kết quả output được tính toán thêm công thức sau:

$$W \leftarrow W - \eta \cdot \frac{\partial \text{Loss}}{\partial W}$$

trong đó η là tốc độ học (learning rate).

- Cập nhật ma trận trọng số: Tương tự như các ma trận trọng số khác, thông thường ma trận trọng số được sử dụng phương pháp Gradient Decent để cập nhật giá trị

Để chuẩn hóa đầu ra và làm cho các thông tin ở lớp trước dễ dàng đến với các lớp sau, ta thực hiện Add và Normalization với công thức như sau:

$$Y = \text{LayerNorm}(X + Z)$$

Trong đó:

- X: Ma trận của word embedding sau khi được positional embedding
- Z : ma trận kết của của multi-head attention

Việc này giúp cho việc training dễ dàng hơn và các ma trận trọng số hội tụ nhanh hơn.

Mạng Feed-forward (FNN)

Mạng Feed-Forward Network (FNN) trong mỗi lớp encoder đóng vai trò quan trọng trong việc xử lý và biến đổi thông tin ở từng lớp của mô hình. FNN này là một lớp mạng tuyến tính hai tầng, thường bao gồm:

- Lớp Dense đầu tiên: Áp dụng phép biến đổi tuyến tính để mở rộng số chiều của vector đầu vào. Chiều rộng thường được mở rộng gấp 4 lần chiều của đầu vào.
- Hàm kích hoạt (thường là ReLU): Tạo ra phi tuyến để giúp mô hình học các mối quan hệ phức tạp trong dữ liệu.
- Lớp Dense thứ hai: Đưa số chiều về lại kích thước ban đầu của đầu vào.

Sau khi kết thúc FNN, tiếp tục thực hiện Add và normalization. kết quả được đưa sang decoder.

2.2.3 Chuẩn bị cho quá trình decoder

Thực hiện các bước vectorization và Positional embedding với output(được shiftright và thêm một kí tự <sos> vào phía trước).

2.2.4 Quá trình decoder

Masked multi-head attention

Trong mô hình Transformer, quy trình mask multi-head attention được sử dụng chủ yếu trong phần decoder để đảm bảo rằng tại mỗi vị trí từ đang xét, mô hình chỉ có thể "nhìn thấy" những từ phía trước hoặc vị trí hiện tại, mà không thể nhìn các từ trong tương lai. Điều này rất quan trọng cho các bài toán dịch máy, giúp đảm bảo tính tuần tự của ngôn ngữ.

Nhìn chung thuật toán khá giống với Multi-head attention tuy nhiên có một điểm đặc biệt là một mask được sử dụng đối với các attention score để che đi vị trí những từ phía sau. Trong decoder, mask này là causal mask hoặc look-ahead mask, thường là một ma trận tam giác trên, che đi các từ ở vị trí tương lai.

Các giá trị score được gán giá trị âm vô cùng để đảm bảo sau khi thực hiện phép nhân và softmax thì các giá trị bị che đi sẽ quay về giá trị 0 và không được tính đến.

Quy trình này giúp đảm bảo rằng tại mỗi bước, decoder chỉ sử dụng thông tin từ các từ trước và vị trí hiện tại, giúp mô hình sinh ra các từ tiếp theo một cách hợp lý và tuần tự.

Multi-head attention

Multi-head attention ở decoder hoạt động tương tự giống với multi-head attention ở encoder. Tuy nhiên, thay vì việc khởi tạo một giá trị bất kì cho các ma trận trọng số W_q, W_k, W_v thì các ma trận đó được khởi tạo như sau:

$$Q = X_{\text{dec}} \cdot W_q^{\text{enc-dec}}, \quad K = X_{\text{enc}} \cdot W_k^{\text{enc-dec}}, \quad V = X_{\text{enc}} \cdot W_v^{\text{enc-dec}}$$

Trong đó:

- X_{enc} : là output của encoder
- $W_k^{\text{enc-dec}}, W_v^{\text{enc-dec}}, W_q^{\text{enc-dec}}$ là ma trận hệ số W_k, W_v và W_q trong encoder
- X_{dec} : là output của masked multi-head attention

Kết quả ta thu được một tập các vector W_q, W_k, W_v . Sau khi có các vector trọng số ta thực hiện multi-head attention như trong phần encoder.

FNN

Trong phần này ta thực hiện mạng FNN tương tự như phần encoder.

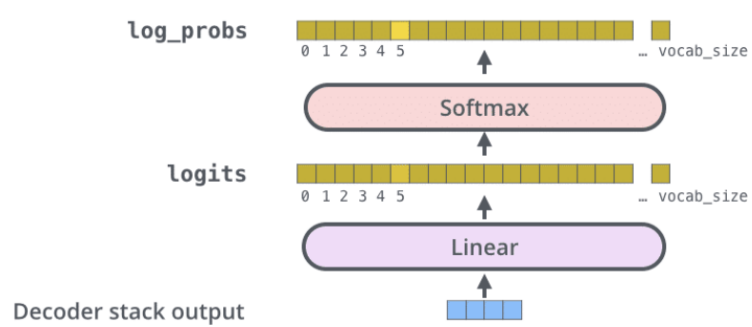
Linear

Khởi tạo một ma trận trọng số W_{out} có kích thước (X_{model}, V) . Trong đó X_{model} là kích thước input đầu vào, thường là 512; V là kích thước tập từ vựng.

Cập nhật qua hàm mất mát: Mỗi lần mô hình dự đoán một từ, hàm mất mát (chẳng hạn Cross-Entropy) sẽ so sánh xác suất dự đoán với nhãn thực tế. Dựa trên sự khác biệt này, các trọng số của W_{out} sẽ được điều chỉnh để dự đoán chính xác hơn trong các lần lặp tiếp theo. Qua nhiều vòng lặp, W_{out} được cập nhật để có thể ánh xạ chính xác hơn tổng những lần tiếp theo.

Softmax

Sử dụng hàm softmax để đưa ra xác suất và ánh xạ lên từ tương ứng.



Hình 2.4: Mô tả quá trình thực hiện linear và softmax

Chương 3

Triển khai mô hình, khảo nghiệm độ chính xác của mô hình

3.1 Bài toán

Như đã trình bày ở phần đầu của đề tài, bài toán lần này được nhóm tập trung ứng dụng kiến trúc Transformer để huấn luyện một mô hình NMT(Neural Machine Translation)

3.2 Tổng quan về dữ liệu

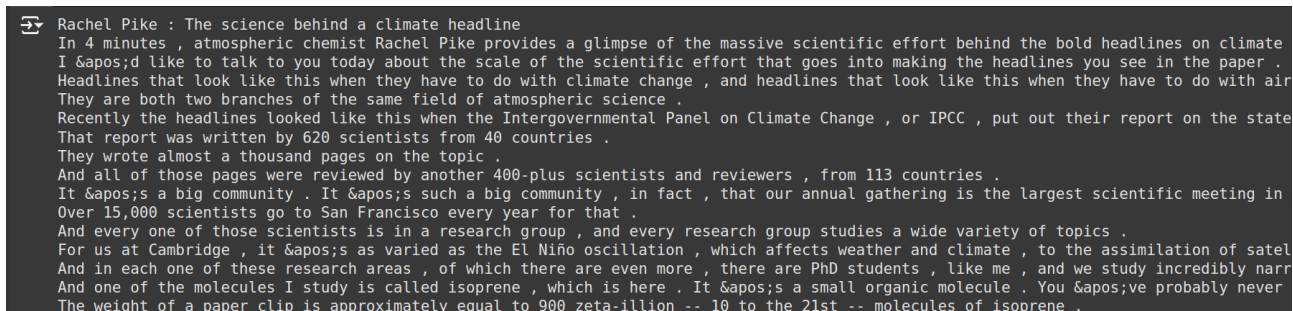
3.2.1 Mô tả chung về dữ liệu

Dữ liệu sử dụng trong bài được lấy từ "The Stanford Natural Language Processing Group", dữ liệu nhóm đã được sử dụng gồm 4 file "train.en", "train.vi", "tst2013.en", "tst2013.vi".

- Link data: https://drive.google.com/drive/folders/1KptjyBk6TV9nYsBePka4O5h0-TCDXGe0?usp=drive_link
- Data train gồm 133.317 câu English-Vietnamese
- Data validation gồm 1.553 câu English-Vietnamese

Data Train

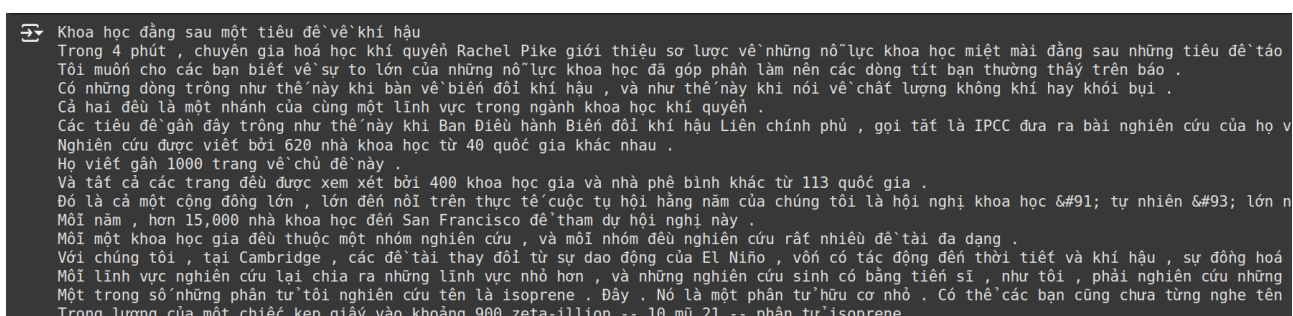
- File "train.en" chứa các câu ngôn ngữ Tiếng Anh



Rachel Pike : The science behind a climate headline
 In 4 minutes , atmospheric chemist Rachel Pike provides a glimpse of the massive scientific effort behind the bold headlines on climate .
 I 'd like to talk to you today about the scale of the scientific effort that goes into making the headlines you see in the paper .
 Headlines that look like this when they have to do with climate change , and headlines that look like this when they have to do with air .
 They are both two branches of the same field of atmospheric science .
 Recently the headlines looked like this when the Intergovernmental Panel on Climate Change , or IPCC , put out their report on the state .
 That report was written by 620 scientists from 40 countries .
 They wrote almost a thousand pages on the topic .
 And all of those pages were reviewed by another 400-plus scientists and reviewers , from 113 countries .
 It 's a big community . It 's such a big community , in fact , that our annual gathering is the largest scientific meeting in .
 Over 15,000 scientists go to San Francisco every year for that .
 And every one of those scientists is in a research group , and every research group studies a wide variety of topics .
 For us at Cambridge , it 's as varied as the El Niño oscillation , which affects weather and climate , to the assimilation of satel .
 And in each one of these research areas , of which there are even more , there are PhD students , like me , and we study incredibly narr .
 And one of the molecules I study is called isoprene , which is here . It 's a small organic molecule . You 've probably never .
 The weight of a paper clip is approximately equal to 900 zeta-illion -- 10 to the 21st -- molecules of isoprene .

Hình 3.1: File "train.en"

- File "train.vi" chứa các câu ngôn ngữ Tiếng Việt được dịch tương ứng với các câu Tiếng Anh trong file "train.en"

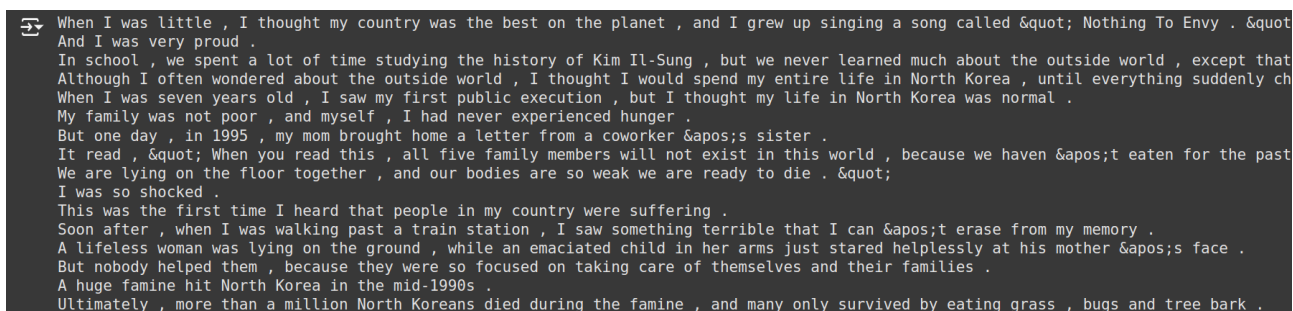


Khoa học đằng sau một tiêu đề về khí hậu
 Trong 4 phút , chuyên gia hoá học khí quyển Rachel Pike giới thiệu sơ lược về những nỗ lực khoa học miệt mài đằng sau những tiêu đề táo .
 Tôi muốn cho các bạn biết về sự to lớn của những nỗ lực khoa học đã góp phần làm nên các dòng tít bạn thường thấy trên báo .
 Có những dòng trông như thế này khi bạn về biến đổi khí hậu , và như thế này khi nói về chất lượng không khí hay khói bụi .
 Cả hai đều là một nhánh của cùng một lĩnh vực trong ngành khoa học khí quyển .
 Các tiêu đề gần đây trông như thế này khi Ban Điều hành Biến đổi khí hậu Liên chính phủ , gọi tắt là IPCC đưa ra bài nghiên cứu của họ v .
 Nghiên cứu được viết bởi 620 nhà khoa học từ 40 quốc gia khác nhau .
 Họ viết gần 1000 trang về chủ đề này .
 Và tất cả các trang đều được xem xét bởi 400 khoa học gia và nhà phê bình khác từ 113 quốc gia .
 Đó là cả một cộng đồng lớn , lớn đến nỗi trên thực tế cuộc tụ hội hằng năm của chúng tôi là hội nghị khoa học [tự nhiên] lớn n .
 Mỗi năm , hơn 15,000 nhà khoa học đến San Francisco để tham dự hội nghị này .
 Mỗi một khoa học gia đều thuộc một nhóm nghiên cứu , và mỗi nhóm đều nghiên cứu rất nhiều đề tài đa dạng .
 Với chúng tôi , tại Cambridge , các đề tài thay đổi từ sự dao động của El Niño , vốn có tác động đến thời tiết và khí hậu , sự đồng hoá .
 Mỗi lĩnh vực nghiên cứu lại chia ra những lĩnh vực nhỏ hơn , và những nghiên cứu sinh có bằng tiến sĩ , như tôi , phải nghiên cứu những .
 Một trong số những phân tử tôi nghiên cứu tên là isoprene . Đây . Nó là một phân tử hữu cơ nhỏ . Có thể các bạn cũng chưa từng nghe tên .
 Trọng lượng của một chiếc kẹp giấy vào khoảng 900 zeta-illion -- 10 mũ 21 -- phân tử isoprene .

Hình 3.2: File "train.vi"

Data Validation

- File "tst2013.en" chứa các câu ngôn ngữ Tiếng Anh



When I was little , I thought my country was the best on the planet , and I grew up singing a song called " Nothing To Envy . "
 And I was very proud .
 In school , we spent a lot of time studying the history of Kim Il-Sung , but we never learned much about the outside world , except that .
 Although I often wondered about the outside world , I thought I would spend my entire life in North Korea , until everything suddenly ch .
 When I was seven years old , I saw my first public execution , but I thought my life in North Korea was normal .
 My family was not poor , and myself , I had never experienced hunger .
 But one day , in 1995 , my mom brought home a letter from a coworker 's sister .
 It read , " When you read this , all five family members will not exist in this world , because we haven 't eaten for the past .
 We are lying on the floor together , and our bodies are so weak we are ready to die . "
 I was so shocked .
 This was the first time I heard that people in my country were suffering .
 Soon after , when I was walking past a train station , I saw something terrible that I can 't erase from my memory .
 A lifeless woman was lying on the ground , while an emaciated child in her arms just stared helplessly at his mother 's face .
 But nobody helped them , because they were so focused on taking care of themselves and their families .
 A huge famine hit North Korea in the mid-1990s .
 Ultimately , more than a million North Koreans died during the famine , and many only survived by eating grass , bugs and tree bark .

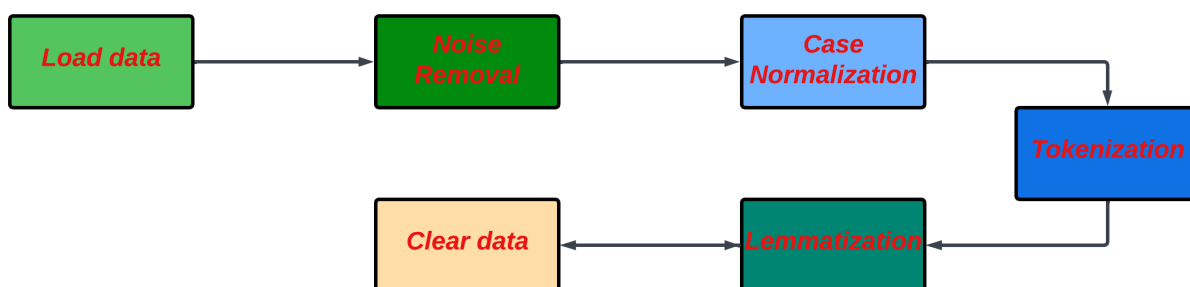
Hình 3.3: File "tst2013.en"

- File "tst2013.vi" chứa các câu ngôn ngữ Tiếng Việt được dịch tương ứng với các câu Tiếng Anh trong file "tst2013.en"

Khi tôi còn nhỏ , Tôi nghĩ rằng BắcTriều Tiên là đất nước tốt nhất trên thế giới và tôi thường hát bài " Chúng ta chẳng có gì phải
 Tôi đã rất tự hào về đất nước tôi .
 Ở trường , chúng tôi dành rất nhiều thời gian để học về cuộc đời của chủ tịch Kim II- Sung , nhưng lại không học nhiều về thế giới bên n
 Mặc dù tôi đã từng tự hỏi không biết thế giới bên ngoài kia như thế nào , nhưng tôi vẫn nghĩ rằng mình sẽ sống cả cuộc đời ở BắcTriều Ti
 Khi tôi lên 7 , tôi chứng kiến cảnh người ta xử bắn công khai lãnh đầu tiên trong đời , nhưng tôi vẫn nghĩ cuộc sống của mình ở đây là ho
 Gia đình của tôi không nghèo , và bản thân tôi thì chưa từng phải chịu đói .
 Nhưng vào một ngày của năm 1995 , mẹ tôi mang về nhà một lá thư từ một người chị em cùng chỗ làm với mẹ .
 Trong đó có viết : Khi chị đọc được những dòng này thì cả gia đình 5 người của em đã không còn trên cõi đời này nữa , bởi vì cả nhà em đ
 Tất cả cùng nằm trên sàn , và cơ thể chúng tôi yếu đến có thể cảm thấy như cái chết đang đến rất gần .
 Tôi đã bị sốc .
 Vì đó là lần đầu tiên tôi biết rằng đồng bào của tôi đang phải chịu đựng như vậy .
 Không lâu sau đó , khi tôi đi qua một nhà ga , tôi nhìn thấy một cảnh tượng kinh hoàng mà tôi không bao giờ có thể quên
 Trên nền nhà ga là xác chết của một người đàn bà hai tay vẫn đang ôm một đứa bé hóc hác và đứa bé chỉ biết nhìn chăm chăm vào khuôn mặt
 Nhưng không có ai giúp họ , bởi vì tất cả đều đang phải lo cho chính mình và cả gia đình .
 Vào giữa những năm 90 , Bắc Triều Tiên trải qua một nạn đói trầm trọng .
 Nó khiến hơn một triệu người Triều Tiên chết trong nạn đói , và nhiều người chỉ sống sót phải ăn cỏ , sâu bọ và vỏ cây .

Hình 3.4: File "tst2013.vi"

3.2.2 Tiền xử lý dữ liệu



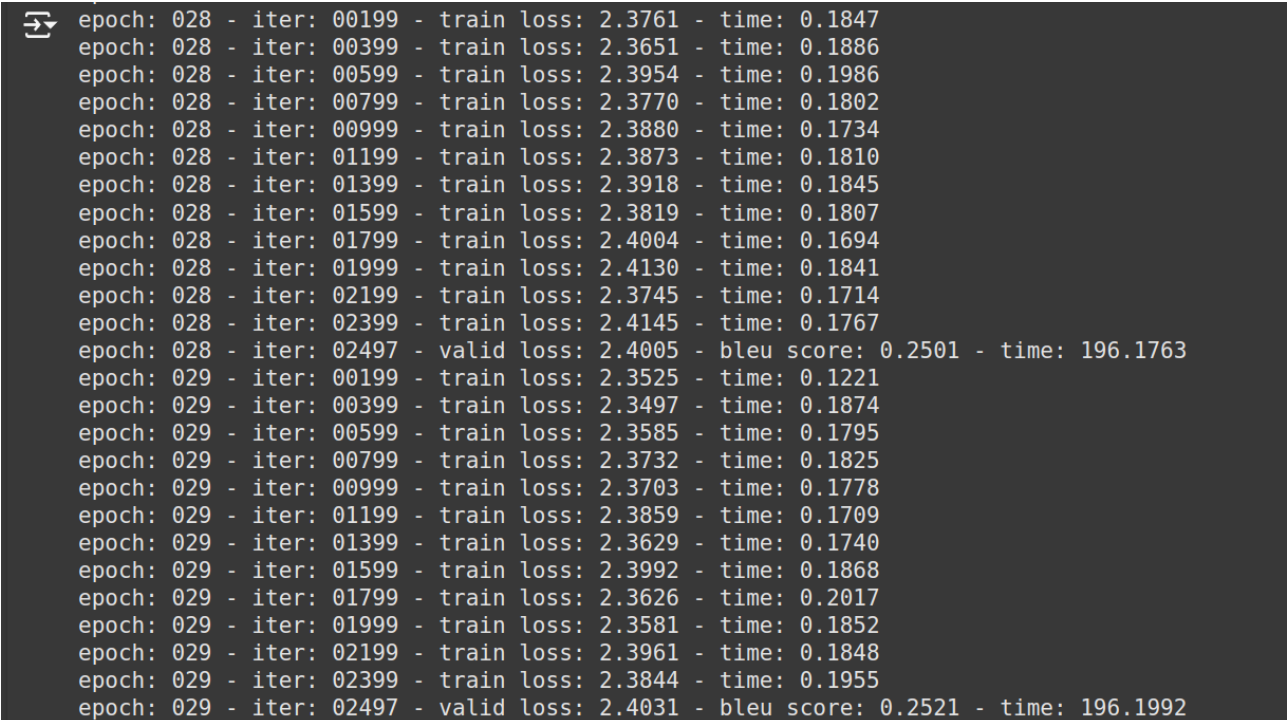
Hình 3.5: Quá trình xử lý dữ liệu

Ngoài việc áp dụng các kiến thức đã được trình bày ở mục 2.1.2, thực hiện quy trình tiền xử lý dữ liệu ngôn ngữ tự nhiên để xử lý dữ liệu ở mục text. Bên cạnh đó nhóm em sử dụng mô hình ngôn ngữ "en_core_web_sm" của thư viện Spacy trong công việc xử lý như phân tích các cú pháp ngữ pháp, tokenization,...

3.2.3 Look up table

Sau tiền xử lý dữ liệu, ta sử dụng look up table để chuyển hóa từ thành dạng vector.

3.3 Triển khai mô hình



```
epoch: 028 - iter: 00199 - train loss: 2.3761 - time: 0.1847
epoch: 028 - iter: 00399 - train loss: 2.3651 - time: 0.1886
epoch: 028 - iter: 00599 - train loss: 2.3954 - time: 0.1986
epoch: 028 - iter: 00799 - train loss: 2.3770 - time: 0.1802
epoch: 028 - iter: 00999 - train loss: 2.3880 - time: 0.1734
epoch: 028 - iter: 01199 - train loss: 2.3873 - time: 0.1810
epoch: 028 - iter: 01399 - train loss: 2.3918 - time: 0.1845
epoch: 028 - iter: 01599 - train loss: 2.3819 - time: 0.1807
epoch: 028 - iter: 01799 - train loss: 2.4004 - time: 0.1694
epoch: 028 - iter: 01999 - train loss: 2.4130 - time: 0.1841
epoch: 028 - iter: 02199 - train loss: 2.3745 - time: 0.1714
epoch: 028 - iter: 02399 - train loss: 2.4145 - time: 0.1767
epoch: 028 - iter: 02497 - valid loss: 2.4005 - bleu score: 0.2501 - time: 196.1763
epoch: 029 - iter: 00199 - train loss: 2.3525 - time: 0.1221
epoch: 029 - iter: 00399 - train loss: 2.3497 - time: 0.1874
epoch: 029 - iter: 00599 - train loss: 2.3585 - time: 0.1795
epoch: 029 - iter: 00799 - train loss: 2.3732 - time: 0.1825
epoch: 029 - iter: 00999 - train loss: 2.3703 - time: 0.1778
epoch: 029 - iter: 01199 - train loss: 2.3859 - time: 0.1709
epoch: 029 - iter: 01399 - train loss: 2.3629 - time: 0.1740
epoch: 029 - iter: 01599 - train loss: 2.3992 - time: 0.1868
epoch: 029 - iter: 01799 - train loss: 2.3626 - time: 0.2017
epoch: 029 - iter: 01999 - train loss: 2.3581 - time: 0.1852
epoch: 029 - iter: 02199 - train loss: 2.3961 - time: 0.1848
epoch: 029 - iter: 02399 - train loss: 2.3844 - time: 0.1955
epoch: 029 - iter: 02497 - valid loss: 2.4031 - bleu score: 0.2521 - time: 196.1992
```

Hình 3.6: Kết quả khi sau khi train với epochs=30

Đánh giá Train&Valid Loss

- Nhận thấy rằng trong epoch 28 train loss dao động từ khoảng 2.3761 đến 2.4145. Sang đến epoch 29 train loss giảm nhẹ với mức giao động từ 2.3525 đến 2.3961, điều này cho thấy mô hình vẫn đang có xu hướng hội tụ dần.
- Đối với valid loss tại cuối epoch 28 và epoch 29 là 2.4005 và 2.4031, ta thấy rằng không có sự thay đổi nào quá lớn cho thấy sự ổn định trong quá trình đánh giá. Tuy nhiên valid loss còn khá cao cần được điều chỉnh và cải thiện thêm.
- Việc train loss đang có xu hướng hội tụ cũng như valid loss có sự ổn định và không quá cao cho thấy mô hình không bị overfitting hoặc underfitting.

Đánh giá về BLEU Score

- Sau epoch 29, BLEU Score tăng nhẹ từ 0.2501 lên 0.2521. BLEU Score ở mức ổn đối với một mô hình NMT nhỏ cùng với một tập dữ liệu không quá lớn.

Đánh giá về thời gian huấn luyện

- Thời gian huấn luyện mỗi epoch là khá hợp lý và không quá dài đối với một mô hình NMT nhỏ.

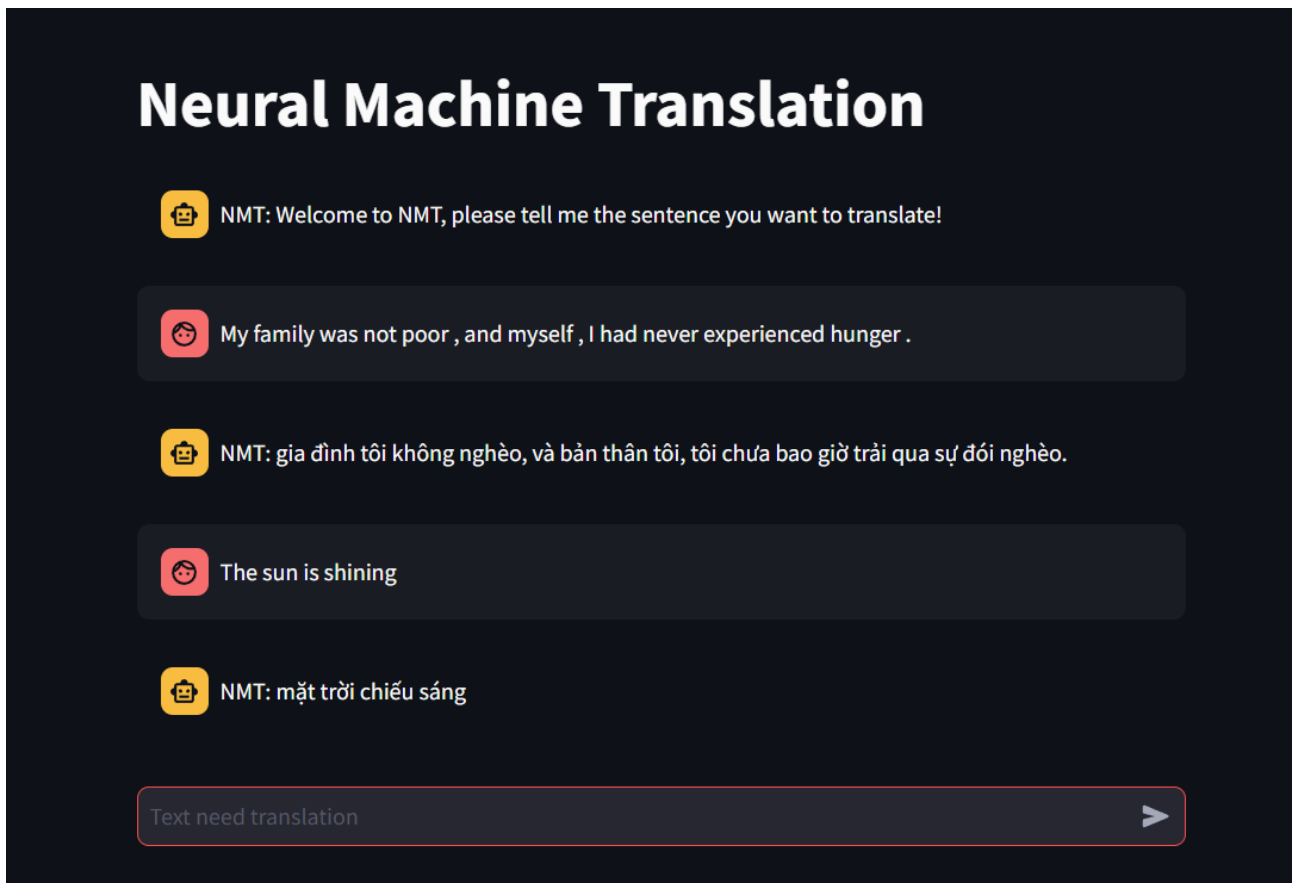
Nhìn chung, mô hình có chất lượng ổn định với một mô hình NMT nhỏ và tập dữ liệu không quá lớn.

Chương 4

Triển khai mô hình chạy trên thời gian thực

Nhóm em tiến hành triển khai một hệ thống dịch máy (Neural Machine Translation - NMT) chạy trên thời gian thực, với mục tiêu cung cấp khả năng dịch văn bản một cách nhanh chóng và chính xác cho người dùng.

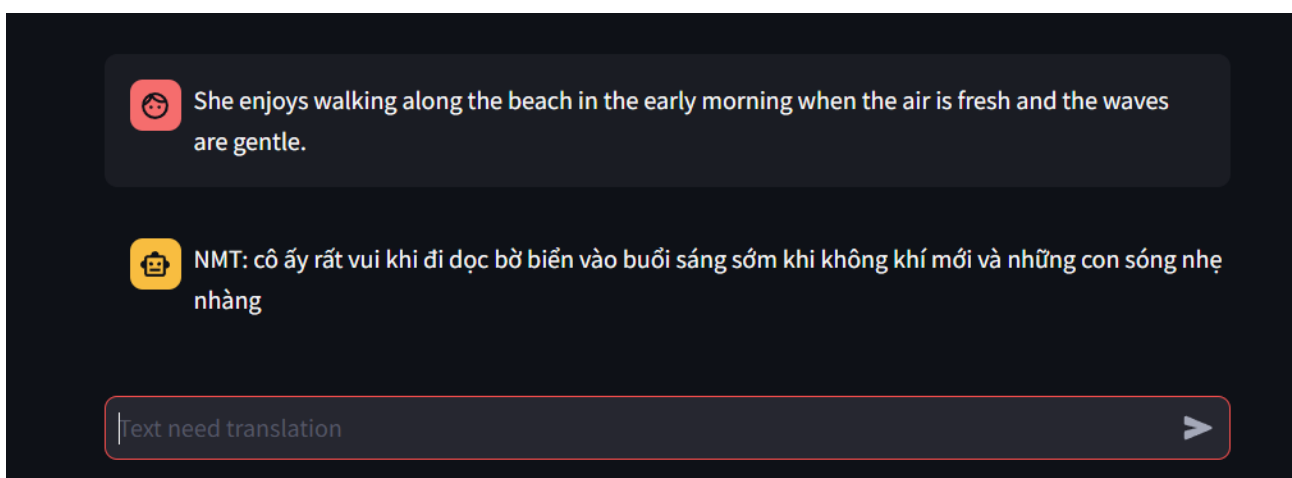
Bằng cách sử dụng thư viện Streamlit để thiết kế giao diện, nhóm em thiết kế giao diện dựa trên ý tưởng của giao diện chatbot. Dữ liệu người dùng được nhập vào từ thanh trò chuyện và chatbot sẽ trả về văn bản đã được dịch. Ứng dụng trả về kết quả trung bình 1-2s cho một câu trả lời, đáp ứng được nhu cầu sử dụng thời gian thực.



Hình 4.1: Giao diện cơ bản

4.1 Đánh giá mô hình

Mô hình dịch của nhóm chúng em dịch tương đối ổn nếu so với một mô hình dịch nhỏ với thời gian huấn luyện hạn chế. Khả năng hoạt động của mô hình khá tốt khi được triển khai thời gian thực, độ trễ mô hình không cao, hoàn toàn đáp ứng được nhu cầu sử dụng.



Hình 4.2: Chất lượng bản dịch

4.2 Phát triển trong tương lai

Qua tìm hiểu nhóm chúng em thấy rằng, kiến trúc Transformer đã chứng minh được tính hiệu quả vượt trội trong dịch máy, trong tương lai nhóm chúng em muốn tiếp tục tối ưu hiệu suất của mô hình cũng như tăng cường chất lượng bản dịch với các phương pháp khác như Fine-tuning,...Giúp cho mô hình hoàn thiện nhất có thể.

Chương 5

Tổng Kết

Trong học phần NLP, nhóm chúng em đã nghiên cứu và triển khai mô hình dịch máy (Neural Machine Translation - NMT) sử dụng kiến trúc Transformer, một bước tiến lớn trong lĩnh vực xử lý ngôn ngữ tự nhiên. Qua quá trình tìm hiểu, chúng tôi nhận thấy Transformer đã khắc phục được nhiều hạn chế của các mô hình dịch máy trước đó như Recurrent Neural Networks (RNN) và Long Short-Term Memory (LSTM) nhờ cơ chế Self-Attention và khả năng xử lý song song dữ liệu hiệu quả. Điều này giúp Transformer đạt độ chính xác cao hơn và thời gian huấn luyện ngắn hơn, đặc biệt khi làm việc với các tập dữ liệu lớn.

Thực nghiệm của chúng tôi trên tập dữ liệu gồm 133,000 câu đã cho thấy mô hình Transformer có khả năng xử lý và dịch thuật nhanh chóng, đồng thời duy trì chất lượng dịch ổn định và phù hợp ngữ cảnh. Nhờ khả năng học được các quan hệ ngữ nghĩa và cú pháp phức tạp trong ngôn ngữ, mô hình Transformer đã tạo ra các bản dịch có tính chính xác cao và mượt mà.

Tóm lại, mô hình Transformer đã chứng tỏ vai trò quan trọng trong dịch máy hiện đại, mở ra tiềm năng cải thiện chất lượng dịch máy tự động và khả năng áp dụng rộng rãi trong nhiều ngôn ngữ và lĩnh vực khác nhau.

Tài liệu tham khảo :

1. <https://trituenhantao.io/tin-tuc/minh-hoa-transformer/>
2. <https://www.youtube.com/watch?v=DVouDUMNFDY>
3. [Attention is all you need](#)
4. [OPTIMAL FEED-FORWARD NEURAL NETWORK ARCHITECTURES](#)