**Bộ giáo dục và đào tạo**

**Trường Đại học Ngoại ngữ - Tin học TP.HCM**

Ảnh có chứa văn bản, Phông chữ, vòng tròn, biểu tượng

Mô tả được tạo tự động

**BÀI BÁO CÁO KẾT THÚC HỌC PHẦN**

**XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**Nghiên cứu bài báo “Attention is all you need”**

Giảng viên hướng dẫn : TS. Trần Khải Thiện

**Sinh viên thực hiện:**

Phùng Tấn Phúc – 21DH111460

**Tháng 7 Năm 2024**

**Mục lục**

1. GIỚI THIỆU 5

2. CƠ SỞ LÝ THUYẾT 7

2.1 Mạng Nơ-ron và Mạng Nơ-ron Tích Chập (CNN): 7

2.2 Mạng Nơ-ron Hồi Quy (RNN) và Long Short-Term Memory (LSTM): 7

2.3 Attention và Self-Attention: 8

2.4 Mask Multi-Head Attention 8

2.6 Feed-Forward Networks và Residual Connections: 9

2.7 End-to-End Learning: 10

2.8 Các phương pháp huấn luyện và tối ưu hóa: 10

3. CÔNG TRÌNH LIÊN QUAN 11

3.1 Mô hình Recurrent Neural Networks (RNN) và Long Short-Term Memory (LSTM) 11

3.2 Gated Recurrent Units (GRU) 11

3.3 Attention Mechanism 12

3.4 Transformer 12

3.5. Các mô hình tiên tiến dựa trên Transformer 13

4. Cấu Trúc Mô Hình 14

4.1 Encoder 15

4.2 Decoder 16

4.3 Multi-head Attention 17

4.4 Position-wise Feed-Forward Network 18

4.5 Embeddings và Softmax 19

4.6 Positional Encoding 19

5. THỰC NGHIỆM: 20

6. KẾT LUẬN 35

TÀI LIỆU THAM KHẢO (IEEE) 36

**Danh mục hình ảnh**

Hình 1. Positional Encoding 9

Hình 2. Mô hình Transformer 14

Hình 3. Encoder 15

Hình 4. Decoder 16

Hình 5. Multi-head Attention 17

Hình 6. Position-wise Feed-Forward Network 18

Hình 7. Embeddings và Softmax 19

Hình 8. Positional Encoding 19

**Nghiên cứu bài báo “Attention is all you need”**

Lê Đào Sỷ Quỳnh, Châu Phan Nguyên Quốc, Lương Thịnh Phát, Phùng Tấn Phúc, GVHD: Trần Khải Thiện

Khoa Công nghệ Thông tin, Trường Đại học Ngoại ngữ - Tin học Thành phố Hồ Chí Minh

Tác giả liên hệ: Email: 21DH111460@st.huflit.edu.vn | Điện thoại: 0947482105

**Tóm tắt**

Bài báo "Attention Is All You Need" đã giới thiệu mô hình Transformer, một bước đột phá trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), đặc biệt là trong các nhiệm vụ dịch chuỗi. Transformer đã thay thế hoàn toàn mạng nơ-ron hồi quy (RNN) truyền thống bằng cơ chế chú ý (attention), mang lại những lợi ích vượt trội về hiệu quả và độ chính xác. Điểm khác biệt chính của Transformer là việc loại bỏ tính tuần tự trong xử lý chuỗi, cho phép mô hình thực hiện song song hóa, từ đó giảm đáng kể thời gian huấn luyện. Kiến trúc của Transformer bao gồm hai phần chính: bộ mã hóa (Encoder) và bộ giải mã (Decoder). Cả hai phần đều sử dụng cơ chế multi-head self-attention, cho phép mô hình học được các mối quan hệ phức tạp giữa các từ trong câu, đồng thời tập trung vào các phần quan trọng của dữ liệu đầu vào. Kết quả thực nghiệm cho thấy Transformer đạt hiệu suất vượt trội trong dịch máy, đạt 28.4 BLEU cho dịch Anh-Đức và 41.8 BLEU cho dịch Anh-Pháp trên bộ dữ liệu WMT 2014. Điều này chứng tỏ khả năng dịch chính xác và mượt mà của mô hình. Ngoài dịch máy, Transformer còn thể hiện khả năng tổng quát hóa tốt khi áp dụng vào các nhiệm vụ NLP khác như phân tích cú pháp tiếng Anh. Với những ưu điểm vượt trội, Transformer đã mở ra một hướng phát triển mới cho lĩnh vực NLP, tạo nền tảng cho nhiều mô hình tiên tiến sau này như BERT, GPT-2, và GPT-3

**Từ khoá:** Transformer; RNN; attention; Encoder; Decoder; multi-head self-attention

1. GIỚI THIỆU

Trong những năm gần đây, mô hình Transformer đã trở thành một đột phá quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Được giới thiệu lần đầu tiên bởi Ashish Vaswani. trong bài báo nổi tiếng "Attention Is All You Need", Transformer đã thay đổi cách mà chúng ta tiếp cận các vấn đề liên quan đến ngôn ngữ. Trước khi xuất hiện mô hình này, các kiến trúc mạng nơ-ron tuần tự như RNN (Recurrent Neural Networks) và LSTM (Long Short-Term Memory) là những công cụ chính cho các nhiệm vụ như dịch tự động và tạo văn bản. Tuy nhiên, những mô hình này gặp phải nhiều hạn chế như khó khăn trong việc xử lý các chuỗi dữ liệu dài và vấn đề về gradient vanishing.

Transformer giải quyết những hạn chế này bằng cách loại bỏ hoàn toàn sự tuần tự trong việc xử lý chuỗi, thay vào đó sử dụng hoàn toàn cơ chế Attention. Điểm đột phá của Transformer nằm ở việc sử dụng Multi-head Attention, cho phép mô hình học được các mối quan hệ phức tạp giữa các từ trong câu mà không bị ràng buộc bởi thứ tự tuần tự. Điều này giúp tăng cường hiệu quả huấn luyện và cải thiện độ chính xác của mô hình trên nhiều nhiệm vụ khác nhau trong NLP.

Ngoài ra, Transformer còn giới thiệu các khái niệm mới như Positional Encoding để giúp mô hình nhận biết vị trí của các từ trong câu, và các kỹ thuật như Residual Connection và Layer Normalization để cải thiện quá trình huấn luyện. Những cải tiến này đã khiến Transformer trở thành nền tảng cho nhiều mô hình tiên tiến sau này như BERT, GPT-2, và GPT-3, tiếp tục mở rộng khả năng của cơ chế Attention trong việc giải quyết các nhiệm vụ phức tạp trong NLP.

BERT (Bidirectional Encoder Representations from Transformers), được phát triển bởi Jacob Devlin., là một trong những mô hình dựa trên Transformer nổi bật nhất. BERT cải tiến cơ chế Attention bằng cách áp dụng Attention hai chiều (bidirectional), cho phép mô hình hiểu được ngữ cảnh của từ từ cả hai phía trái và phải trong câu. Điều này mang lại hiệu quả vượt trội trong nhiều nhiệm vụ NLP như phân loại văn bản, trả lời câu hỏi và nhận dạng thực thể.

GPT (Generative Pre-trained Transformer), được phát triển bởi OpenAI, là một dòng mô hình khác dựa trên Transformer. GPT nổi bật với khả năng tạo văn bản tự nhiên và mạch lạc, đã được sử dụng rộng rãi trong các ứng dụng như chatbot và trợ lý ảo. GPT-3, phiên bản mới nhất, có hơn 175 tỷ tham số, thể hiện khả năng mạnh mẽ trong việc hiểu và tạo ra ngôn ngữ tự nhiên.

Mô hình T5 (Text-to-Text Transfer Transformer) của Google cũng là một ví dụ đáng chú ý khác. T5 tiếp cận các nhiệm vụ NLP dưới dạng các bài toán sinh văn bản, giúp thống nhất các nhiệm vụ khác nhau dưới một mô hình duy nhất. Cách tiếp cận này cho phép T5 thực hiện hiệu quả các nhiệm vụ như dịch máy, tóm tắt văn bản và trả lời câu hỏi.

Những tiến bộ này không chỉ cải thiện hiệu suất của các hệ thống NLP hiện tại mà còn mở ra những khả năng mới cho việc ứng dụng công nghệ ngôn ngữ trong các lĩnh vực khác nhau. Transformer và các mô hình dựa trên nó đã trở thành công cụ không thể thiếu trong hộp công cụ của các nhà nghiên cứu và kỹ sư NLP, thúc đẩy sự phát triển nhanh chóng của lĩnh vực này.

Trong tương lai, có thể mong đợi thấy nhiều cải tiến hơn nữa dựa trên kiến trúc Transformer. Các nghiên cứu hiện tại đang khám phá các phương pháp tối ưu hóa mô hình, giảm thiểu chi phí tính toán và năng lượng, cũng như cải thiện khả năng giải thích và kiểm soát của các mô hình ngôn ngữ. Với tiềm năng to lớn và khả năng thích ứng mạnh mẽ, Transformer hứa hẹn sẽ tiếp tục dẫn đầu trong việc giải quyết các thách thức trong xử lý ngôn ngữ tự nhiên và trí tuệ nhân tạo.

1. CƠ SỞ LÝ THUYẾT

Bài báo "Attention Is All You Need" đã giới thiệu mô hình Transformer, một mô hình transduction đầu tiên hoàn toàn dựa vào cơ chế tự chú ý (self-attention) để tính toán biểu diễn của đầu vào và đầu ra mà không sử dụng các mạng neural hồi quy theo chuỗi (sequence-aligned RNNs) hoặc mạng neural tích chập (convolution).

## **2.1 Mạng Nơ-ron và Mạng Nơ-ron Tích Chập (CNN):**

Mạng nơ-ron (Neural Networks) là nền tảng của học sâu, mô phỏng cách thức hoạt động của não bộ con người với các nơ-ron được kết nối với nhau. Mỗi nơ-ron nhận đầu vào, thực hiện một phép tính đơn giản và truyền kết quả cho các nơ-ron khác.

Mạng nơ-ron tích chập (CNN) là một loại mạng nơ-ron đặc biệt hiệu quả trong việc xử lý dữ liệu có cấu trúc không gian, ví dụ như hình ảnh. CNN sử dụng các lớp tích chập để trích xuất đặc trưng từ các vùng cục bộ trong dữ liệu đầu vào. Trong xử lý ngôn ngữ tự nhiên, CNN có thể được sử dụng để trích xuất đặc trưng ngữ nghĩa từ các cụm từ hoặc câu ngắn. Tuy nhiên, CNN gặp hạn chế khi xử lý các phụ thuộc dài hạn trong chuỗi, vốn rất quan trọng trong nhiều tác vụ NLP như dịch máy.

Tại sao Transformer không sử dụng CNN: Transformer dựa trên cơ chế self-attention, cho phép mô hình nắm bắt được mối quan hệ giữa các từ bất kể khoảng cách giữa chúng trong chuỗi, vượt qua hạn chế của CNN trong việc xử lý phụ thuộc dài hạn.

## **2.2 Mạng Nơ-ron Hồi Quy (RNN) và Long Short-Term Memory (LSTM):**

Mạng nơ-ron hồi quy (RNN) là một loại mạng nơ-ron được thiết kế để xử lý dữ liệu chuỗi. RNN có cấu trúc đặc biệt cho phép chúng lưu trữ thông tin từ các bước thời gian trước đó, giúp chúng học được các mẫu từ dữ liệu chuỗi.

Long Short-Term Memory (LSTM) là một biến thể phức tạp hơn của RNN, được thiết kế để giải quyết vấn đề "vanishing gradient" thường gặp trong RNN truyền thống. LSTM có khả năng học phụ thuộc dài hạn trong chuỗi tốt hơn RNN, nhưng vẫn gặp hạn chế về khả năng tính toán song song do tính chất tuần tự của nó.

Tại sao Transformer không sử dụng RNN hay LSTM: Mặc dù RNN và LSTM đã đạt được nhiều thành công trong NLP, chúng xử lý chuỗi theo tuần tự, dẫn đến hạn chế về tốc độ huấn luyện và khả năng mở rộng cho dữ liệu lớn. Transformer loại bỏ hoàn toàn cấu trúc tuần tự, thay vào đó sử dụng self-attention để tính toán song song mối quan hệ giữa các từ, giúp tăng tốc độ huấn luyện và mở rộng mô hình lên dữ liệu lớn hơn.

## **2.3 Attention và Self-Attention:**

Attention: Là một kỹ thuật đột phá trong NLP, attention cho phép mô hình tập trung vào các phần quan trọng của dữ liệu đầu vào khi xử lý thông tin. Thay vì xử lý toàn bộ chuỗi đầu vào một cách đồng đều như RNN và LSTM, attention tính toán một "trọng số chú ý" cho mỗi phần tử trong chuỗi đầu vào, thể hiện mức độ quan trọng của phần tử đó đối với nhiệm vụ hiện tại.

Ví dụ, trong dịch máy, khi dịch từ "con mèo" trong câu "Con mèo đang ngủ trên ghế sofa", mô hình có thể chú ý nhiều hơn đến các từ "mèo", "ngủ" và "ghế sofa" trong câu nguồn, vì chúng chứa thông tin quan trọng để dịch chính xác từ "con mèo".

Self-Attention: Là một biến thể đặc biệt của attention, self-attention cho phép mô hình học được mối quan hệ giữa các từ trong cùng một chuỗi.

Ví dụ, trong câu "Mặc dù trời mưa, nhưng tôi vẫn cảm thấy rất vui", self-attention cho phép mô hình học được mối quan hệ giữa "trời mưa" và "vui", từ đó hiểu được ý nghĩa tương phản của câu.

## **2.4 Mask Multi-Head Attention**

Mask Multi-Head Attention là một mở rộng của cơ chế tự chú ý đơn đầu (single-head self-attention). Thay vì chỉ tính toán trọng số chú ý một lần, cơ chế này tính toán nhiều bộ trọng số chú ý song song (multi-head) và sau đó kết hợp kết quả lại. Điều này giúp mô hình học được nhiều khía cạnh khác nhau của dữ liệu cùng một lúc.

Quá trình hoạt động của Mask Multi-Head Attention như sau:

1. Tạo nhiều bộ Query, Key và Value: Đầu tiên, embedding của từ được nhân với nhiều ma trận trọng số khác nhau để tạo ra nhiều bộ query, key và value.

2. Tính toán tự chú ý: Với mỗi bộ query, key và value, cơ chế tự chú ý được áp dụng để tính toán các điểm số chú ý và biểu diễn mới cho mỗi từ.

3. Kết hợp các đầu ra: Kết quả từ mỗi bộ chú ý sau đó được kết hợp lại bằng cách nối các vector và nhân với một ma trận trọng số cuối cùng.

Masking được sử dụng để ngăn mô hình tại một vị trí cụ thể nhìn thấy các vị trí tương lai trong chuỗi. Ví dụ, trong dịch máy, khi dịch từ "the" trong câu đầu vào "the cat sat", mô hình không nên xem xét từ "sat" vì nó chưa được dịch. Masking đảm bảo rằng dự đoán tại mỗi vị trí chỉ phụ thuộc vào các vị trí trước đó, duy trì tính tuần tự của quá trình dịch.

**2.5 Positional Encoding**

Trong mô hình Transformer, do không có thông tin thứ tự vị trí được cung cấp bởi các kiến trúc tuần tự như RNN, positional encoding được thêm vào các embedding để cung cấp thông tin về vị trí của từ trong chuỗi.

Positional encoding sử dụng các hàm sin và cosin của các tần số khác nhau để tạo ra các vector vị trí duy nhất cho mỗi từ. Các vector này sau đó được cộng vào các embedding của từ để giữ thông tin thứ tự vị trí.

Công thức của positional encoding là:

Ảnh có chứa văn bản, Phông chữ, hàng, biểu đồ

Mô tả được tạo tự động

### Hình 1. Positional Encoding

Trong đó, pos là vị trí của từ trong chuỗi, i là chỉ số chiều của vector, và  là kích thước của embedding.

Ví dụ, với từ "the" ở vị trí thứ nhất và từ "cat" ở vị trí thứ hai trong câu "the cat sat", vector positional encoding của "the" và "cat" sẽ khác nhau, giúp mô hình phân biệt được vị trí của chúng trong chuỗi.

Positional encoding giúp mô hình Transformer duy trì thông tin thứ tự mà không cần sử dụng các cơ chế tuần tự như RNN, đồng thời tận dụng được khả năng tính toán song song của các GPU hiện đại, từ đó cải thiện hiệu suất và tốc độ huấn luyện mô hình.

## **2.6 Feed-Forward Networks và Residual Connections:**

Feed-Forward Networks: Sau mỗi lớp Multi-Head Attention trong Transformer, dữ liệu được đưa qua một mạng feed-forward. Mạng feed-forward là một mạng nơ-ron truyền thống, nhận đầu vào từ lớp trước đó và áp dụng một hàm phi tuyến (ví dụ như ReLU) để tạo ra đầu ra. Mạng này giúp Transformer học được các biểu diễn phức tạp hơn từ dữ liệu.

Residual Connections: Residual Connections là một kỹ thuật quan trọng được sử dụng trong Transformer (và nhiều kiến trúc mạng nơ-ron sâu khác) để giải quyết vấn đề "vanishing gradient". Residual connections tạo ra các “đường tắt” cho phép gradient lan truyền trực tiếp từ các lớp sau đến các lớp trước, giúp quá trình huấn luyện ổn định hơn và mô hình có thể học được từ dữ liệu phức tạp hơn.

## **2.7 End-to-End Learning:**

End-to-End Learning: Trong bối cảnh của Transformer, end-to-end learning có nghĩa là chúng ta huấn luyện toàn bộ mô hình Transformer từ đầu vào (chuỗi nguồn) đến đầu ra (chuỗi đích) một cách trực tiếp, thay vì huấn luyện riêng biệt các thành phần như encoder và decoder. Điều này cho phép Transformer tự động học cách tối ưu hóa toàn bộ quá trình dịch, từ đó đạt được hiệu suất cao hơn so với các phương pháp huấn luyện truyền thống.

## **2.8 Các phương pháp huấn luyện và tối ưu hóa:**

Gradient Descent: Là một thuật toán tối ưu hóa lặp, được sử dụng để tìm giá trị tối ưu của các tham số của mô hình bằng cách di chuyển theo hướng ngược với gradient của hàm mất mát.

Các biến thể: Có nhiều biến thể của Gradient Descent được sử dụng trong thực tế, ví dụ như Stochastic Gradient Descent (SGD), Mini-batch Gradient Descent, Momentum, Adagrad, RMSprop, và Adam. Mỗi biến thể có ưu nhược điểm riêng và được lựa chọn dựa trên đặc thù của bài toán và dữ liệu.

Adam Optimizer: Là một thuật toán tối ưu hóa dựa trên Gradient Descent, được sử dụng rộng rãi trong học sâu vì khả năng hội tụ nhanh và hiệu quả. Adam kết hợp các ưu điểm của AdaGrad và RMSProp, cho phép nó tự động điều chỉnh tốc độ học cho từng tham số của mô hình.

1. CÔNG TRÌNH LIÊN QUAN

**3.1 Mô hình Recurrent Neural Networks (RNN) và Long Short-Term Memory (LSTM)**

Recurrent Neural Networks (RNN) là một loại mạng nơ-ron đặc biệt được thiết kế để xử lý dữ liệu tuần tự. RNN duy trì một trạng thái ẩn (hidden state) giúp ghi nhớ thông tin từ các bước trước đó trong chuỗi. Tuy nhiên, RNN gặp phải vấn đề "vanishing gradient" khi chuỗi quá dài, khiến mô hình khó học và ghi nhớ thông tin từ các bước đầu tiên trong chuỗi. Hiện tượng này xảy ra khi gradient trở nên quá nhỏ trong quá trình huấn luyện, dẫn đến việc mô hình không cập nhật được trọng số một cách hiệu quả. Điều này làm giảm khả năng ghi nhớ và xử lý thông tin dài hạn của RNN, khiến nó kém hiệu quả trong các nhiệm vụ yêu cầu xử lý chuỗi dài.

Để khắc phục hạn chế này, Long Short-Term Memory (LSTM) được giới thiệu bởi Hochreiter và Schmidhuber (1997). LSTM bao gồm các cổng đặc biệt để kiểm soát luồng thông tin vào và ra khỏi các tế bào bộ nhớ (memory cells). Cụ thể, LSTM sử dụng ba cổng chính: cổng quên (forget gate), cổng đầu vào (input gate), và cổng đầu ra (output gate). Cổng quên quyết định thông tin nào cần quên từ trạng thái bộ nhớ trước đó, cổng đầu vào điều chỉnh thông tin mới nào cần thêm vào trạng thái bộ nhớ, và cổng đầu ra quyết định phần nào của trạng thái bộ nhớ sẽ được sử dụng để tạo đầu ra. Điều này giúp LSTM có khả năng duy trì thông tin qua các chuỗi dài hơn, cải thiện hiệu quả trong các nhiệm vụ như dịch máy, nhận dạng giọng nói, và phân tích chuỗi thời gian. LSTM đã trở thành một công cụ quan trọng trong xử lý ngôn ngữ tự nhiên (NLP) và các lĩnh vực liên quan, nhờ vào khả năng xử lý chuỗi dài và phức tạp.

**3.2 Gated Recurrent Units (GRU)**

Gated Recurrent Units (GRU) là một biến thể của RNN, được giới thiệu bởi Cho et al. (2014), nhằm đơn giản hóa cấu trúc của LSTM trong khi vẫn giữ lại những ưu điểm chính. GRU kết hợp các cổng đầu vào và cổng quên thành một cổng duy nhất, được gọi là cổng cập nhật (update gate). Cấu trúc này giúp giảm số lượng tham số và tăng tốc độ huấn luyện. GRU cũng sử dụng một cổng thứ hai gọi là cổng đặt lại (reset gate) để quyết định thông tin nào từ trạng thái ẩn trước đó sẽ được sử dụng để tính toán trạng thái hiện tại. Nhờ vào sự đơn giản hóa này, GRU có thể huấn luyện nhanh hơn và yêu cầu ít tài nguyên tính toán hơn so với LSTM.

Mặc dù GRU đơn giản hơn, nhưng nó vẫn hiệu quả trong nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên như dịch máy, tóm tắt văn bản, và phân loại cảm xúc. Một số nghiên cứu cho thấy GRU có thể đạt hiệu suất tương đương hoặc thậm chí vượt trội hơn so với LSTM trong một số trường hợp. Điều này khiến GRU trở thành một lựa chọn hấp dẫn cho các ứng dụng yêu cầu sự hiệu quả về mặt tính toán mà vẫn duy trì được hiệu suất cao.

**3.3 Attention Mechanism**

Attention Mechanism là một bước đột phá trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), cho phép mô hình tập trung vào các phần quan trọng của dữ liệu đầu vào khi tạo ra từng phần của đầu ra. Cơ chế chú ý được Bahdanau et al. (2014) giới thiệu đầu tiên trong mô hình dịch máy seq2seq. Thay vì chỉ dựa vào trạng thái ẩn cuối cùng của RNN, cơ chế chú ý tính toán một trọng số chú ý cho mỗi phần tử đầu vào và sử dụng các trọng số này để tổng hợp thông tin liên quan nhất.

Ví dụ, trong dịch máy, khi dịch từ một ngôn ngữ sang ngôn ngữ khác, mô hình có thể chú ý nhiều hơn đến các từ quan trọng hoặc có ngữ nghĩa tương tự trong câu nguồn. Điều này giúp cải thiện đáng kể độ chính xác của các mô hình seq2seq với Attention trong dịch máy và các nhiệm vụ NLP khác như tóm tắt văn bản, trả lời câu hỏi, và nhận dạng thực thể. Attention Mechanism đã trở thành một thành phần không thể thiếu trong các mô hình hiện đại, giúp chúng xử lý thông tin hiệu quả hơn và đạt được kết quả xuất sắc trong nhiều nhiệm vụ khác nhau.

**3.4 Transformer**

Transformer, được giới thiệu trong bài báo "Attention Is All You Need" của Vaswani et al. (2017), sử dụng hoàn toàn cơ chế Attention mà không cần đến các cấu trúc tuần tự như RNN hay LSTM. Cấu trúc của Transformer bao gồm hai phần chính: Bộ mã hóa (Encoder) và Bộ giải mã (Decoder), mỗi phần có nhiều lớp của cơ chế Multi-head Attention và Position-wise Feed-Forward Networks.

- Encoder: Gồm nhiều lớp, mỗi lớp có một cơ chế Multi-head Self-Attention và một mạng neural truyền thẳng (Feed-Forward Network). Encoder nhận đầu vào là các vector embedding của các từ trong câu và tạo ra các biểu diễn ngữ nghĩa sâu hơn cho mỗi từ.

- Decoder: Cũng gồm nhiều lớp, mỗi lớp có một cơ chế Multi-head Self-Attention, một cơ chế Multi-head Attention với đầu ra của Encoder, và một mạng Feed-Forward. Decoder sử dụng các biểu diễn từ Encoder và các từ đã được dịch để tạo ra từ tiếp theo trong chuỗi đầu ra.

Transformer đã chứng minh hiệu quả vượt trội trên nhiều nhiệm vụ NLP như dịch máy, tóm tắt văn bản, và nhận dạng thực thể, mở ra hướng phát triển mới cho các mô hình ngôn ngữ tiên tiến. Khả năng song song hóa của Transformer cũng giúp tăng tốc độ huấn luyện và xử lý, khiến nó trở thành một lựa chọn phổ biến trong các hệ thống NLP hiện đại.

**3.5. Các mô hình tiên tiến dựa trên Transformer**

**BERT (Bidirectional Encoder Representations from Transformers)**

BERT, được giới thiệu bởi Devlin et al. (2018), học biểu diễn ngữ nghĩa của từ trong ngữ cảnh hai chiều (bidirectional). Điều này có nghĩa là BERT xem xét cả trái và phải của một từ trong chuỗi khi tính toán biểu diễn của từ đó. BERT đã đạt kết quả xuất sắc trên nhiều nhiệm vụ NLP như phân loại văn bản, trả lời câu hỏi, và nhận dạng thực thể. BERT sử dụng một phương pháp tiền huấn luyện trên các tập dữ liệu lớn và sau đó tinh chỉnh trên các nhiệm vụ cụ thể, giúp cải thiện hiệu suất trên nhiều bài toán khác nhau. Sự thành công của BERT đã thúc đẩy sự phát triển của nhiều biến thể và mô hình tiếp nối như RoBERTa và ALBERT, tiếp tục cải thiện hiệu suất và hiệu quả của các hệ thống NLP.

**GPT (Generative Pre-trained Transformer)**

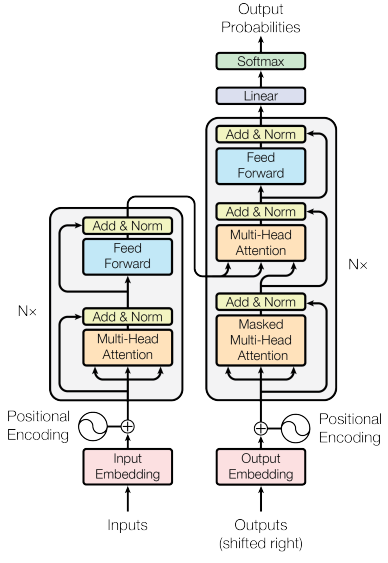
GPT, được giới thiệu bởi Radford et al., chuyên về việc tạo văn bản tự nhiên và sáng tạo. GPT-2 và GPT-3 là các phiên bản cải tiến với số lượng tham số lớn hơn, cho phép mô hình tạo ra văn bản tự nhiên, logic và mạch lạc hơn. GPT đã mở ra nhiều ứng dụng mới trong NLP như viết văn bản tự động, chatbot, và trợ lý ảo. Mô hình GPT hoạt động theo cơ chế chỉ sử dụng Decoder của Transformer và được huấn luyện trên một lượng lớn văn bản để dự đoán từ tiếp theo trong chuỗi. Khả năng của GPT trong việc hiểu ngữ cảnh và tạo ra văn bản giống con người đã được chứng minh qua nhiều ứng dụng thực tế, làm nổi bật tiềm năng của các mô hình ngôn ngữ lớn trong việc thay đổi cách chúng ta tương tác với công nghệ.

**T5 (Text-to-Text Transfer Transformer)**

T5, được giới thiệu bởi Raffel et al. (2019), chuyển đổi nhiều nhiệm vụ NLP về dạng bài toán sinh văn bản. Điều này có nghĩa là cả đầu vào và đầu ra của T5 đều được biểu diễn dưới dạng văn bản, giúp mô hình dễ dàng xử lý nhiều nhiệm vụ khác nhau như dịch máy, tóm tắt văn bản, và trả lời câu hỏi. T5 sử dụng cả Encoder và Decoder của Transformer và được huấn luyện trên một tập dữ liệu lớn với nhiều nhiệm vụ khác nhau, chứng minh hiệu quả cao trong các bài toán chuyển đổi văn bản. Khả năng của T5 trong việc đồng nhất hóa các nhiệm vụ NLP thành một khuôn khổ thống nhất đã giúp đơn giản hóa quy trình phát triển và triển khai các ứng dụng NLP, đồng thời đạt được hiệu suất cao trên nhiều loại nhiệm vụ khác nhau.

1. Cấu Trúc Mô Hình

Mô hình Transformer được cấu trúc theo hai phần chính: Encoder và Decoder, mỗi phần bao gồm một số lớp (stacks) của cơ chế Attention và các mạng fully connected. Dưới đây là mô tả chi tiết về các thành phần chính của mô hình:



### Hình 2. Mô hình Transformer

## **4.1 Encoder**

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

### Hình 3. Encoder

Encoder của Transformer bao gồm một loạt các lớp Multi-head Self-Attention và Position-wise Feed-Forward Networks. Mỗi lớp Encoder bao gồm:

- Multi-head Self-Attention: Thành phần này cho phép mô hình học được các mối quan hệ khác nhau giữa các từ trong câu thông qua việc áp dụng cơ chế Attention nhiều lần với các trọng số khác nhau (heads). Bằng cách này, mô hình có thể tập trung vào các khía cạnh khác nhau của dữ liệu đầu vào đồng thời.

- Position-wise Feed-Forward Network: Sau lớp Attention, các dữ liệu được truyền qua một mạng fully connected với hai lớp và hàm kích hoạt ReLU. Điều này giúp tăng cường khả năng biểu diễn của mô hình.

- Residual Connection và Layer Normalization: Để giúp việc huấn luyện trở nên ổn định và hiệu quả hơn, mỗi lớp trong Encoder sử dụng các kết nối residual và áp dụng normalization. Residual Connection giúp giảm thiểu vấn đề gradient vanishing bằng cách cung cấp các đường dẫn trực tiếp cho gradient khi lan truyền ngược.

## **4.2 Decoder**

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

### Hình 4. Decoder

Phần Decoder cũng bao gồm một loạt các lớp tương tự như Encoder, nhưng có thêm một số đặc điểm để phù hợp với việc sinh chuỗi đầu ra:

- Masked Multi-head Attention: Thành phần này đảm bảo rằng mỗi từ trong chuỗi đầu ra chỉ phụ thuộc vào các từ đã biết trước đó và không nhìn thấy các từ ở tương lai. Điều này được thực hiện bằng cách áp dụng một mặt nạ (mask) để ẩn đi các vị trí tương lai trong quá trình tính toán Attention.

- Encoder-Decoder Attention: Lớp Attention này sử dụng đầu ra của Encoder làm Key và Value, trong khi các Query đến từ lớp Decoder trước đó. Điều này cho phép mô hình tập trung vào các phần quan trọng của chuỗi đầu vào khi sinh ra từng từ của chuỗi đầu ra.

- Position-wise Feed-Forward Network: Giống như trong Encoder, phần Decoder cũng sử dụng một mạng fully connected để tăng cường khả năng biểu diễn.

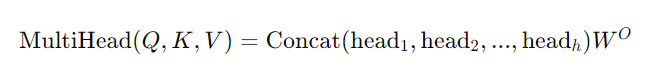
## **4.3 Multi-head Attention**

Ảnh có chứa văn bản, ảnh chụp màn hình, bản phác thảo, biểu đồ

Mô tả được tạo tự động

### Hình 5. Multi-head Attention

Multi-head Attention là một trong những thành phần quan trọng nhất của Transformer. Nó cho phép mô hình tập trung vào nhiều phần khác nhau của chuỗi đầu vào song song, cải thiện khả năng học các mối quan hệ phức tạp giữa các từ. Công thức của Multi-head Attention như sau:



Trong đó mỗi head được tính bằng cách áp dụng Attention một cách song song trên các phần nhỏ của đầu vào:



Các đầu ra của các head sau đó được nối lại và chiếu tuyến tính thêm một lần nữa để tạo thành đầu ra cuối cùng.

## **4.4 Position-wise Feed-Forward Network**

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

### Hình 6. Position-wise Feed-Forward Network

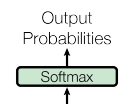
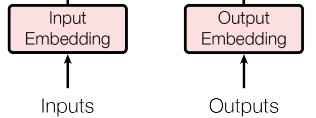
Mạng này bao gồm hai lớp fully connected với hàm kích hoạt ReLU giữa chúng. Công thức của nó như sau:

Ảnh có chứa văn bản, Phông chữ, màu trắng, thuật in máy

Mô tả được tạo tự động

Mạng này được áp dụng một cách độc lập cho từng vị trí trong chuỗi đầu vào.

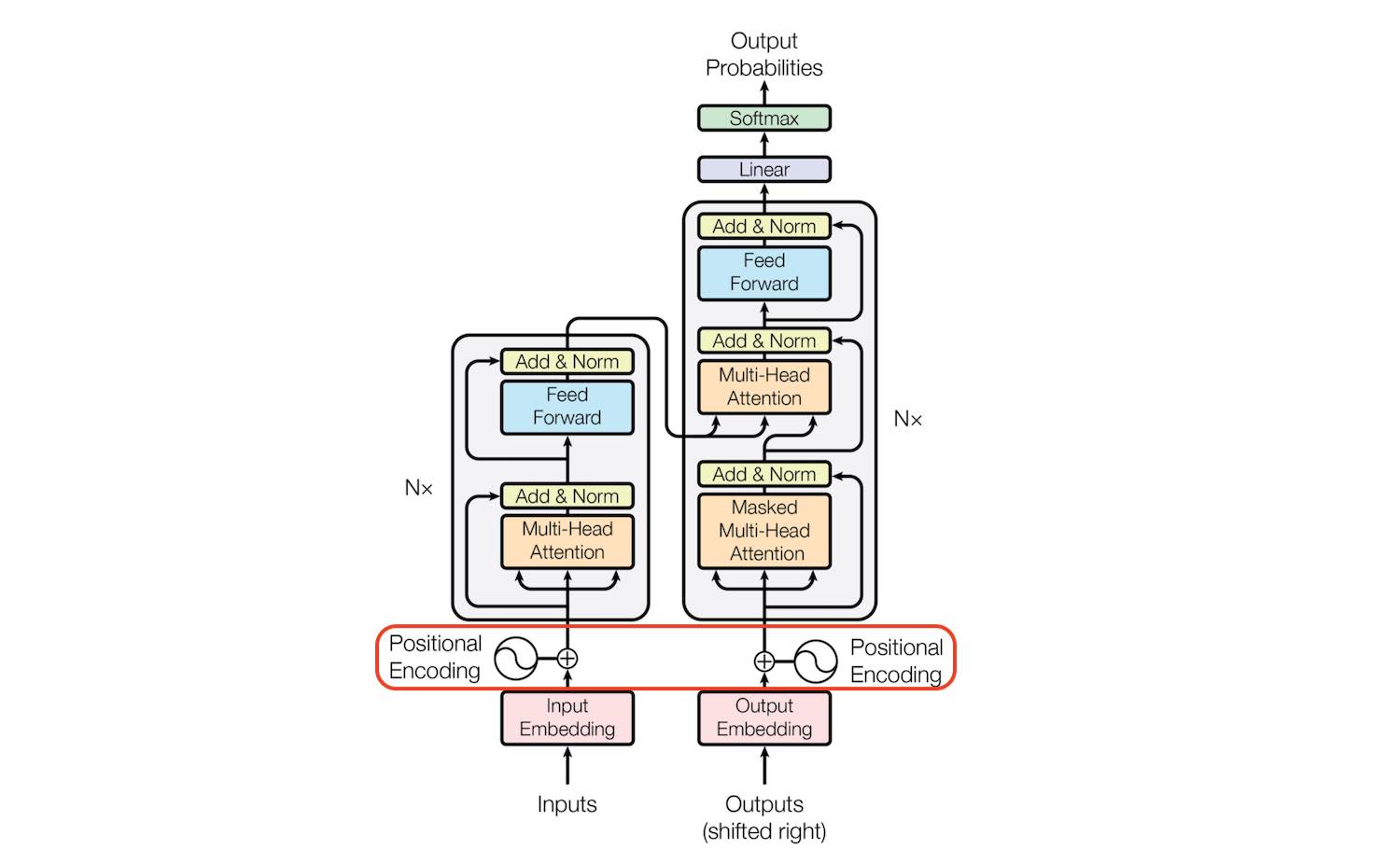
## **4.5 Embeddings và Softmax**



### Hình 7. Embeddings và Softmax

Transformer sử dụng các lớp embedding để chuyển đổi các token đầu vào và đầu ra thành các vector. Các vector này sau đó được truyền qua mô hình và cuối cùng được chuyển đổi lại thành xác suất dự đoán cho từ tiếp theo bằng hàm softmax.

## **4.6 Positional Encoding**



### Hình 8. Positional Encoding

Vì Transformer không sử dụng kiến trúc tuần tự, nó cần một cách để nhận biết vị trí của các từ trong câu. Điều này được thực hiện bằng cách thêm thông tin vị trí vào các vector embedding thông qua Positional Encoding. Công thức của Positional Encoding như sau:

Ảnh có chứa văn bản, Phông chữ, hàng, biểu đồ

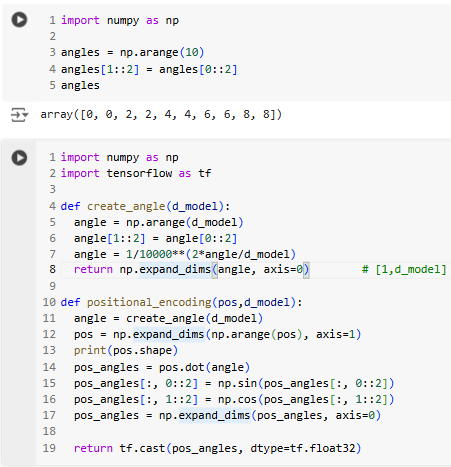
Mô tả được tạo tự động

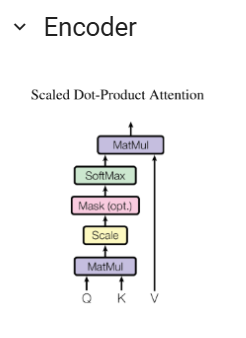
Ở đây, pos là vị trí của từ và i là chiều của vector. Các giá trị sin và cos giúp mô hình phân biệt được vị trí của các từ trong chuỗi.

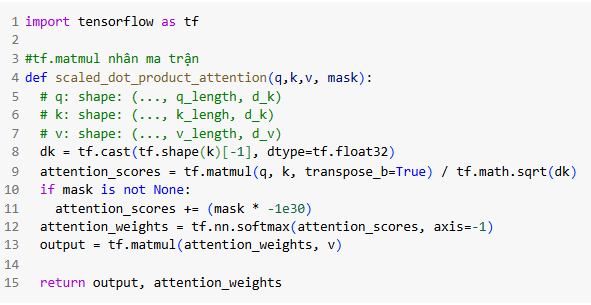
1. THỰC NGHIỆM:

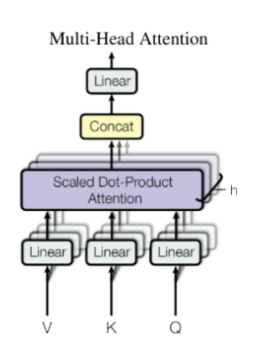
Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

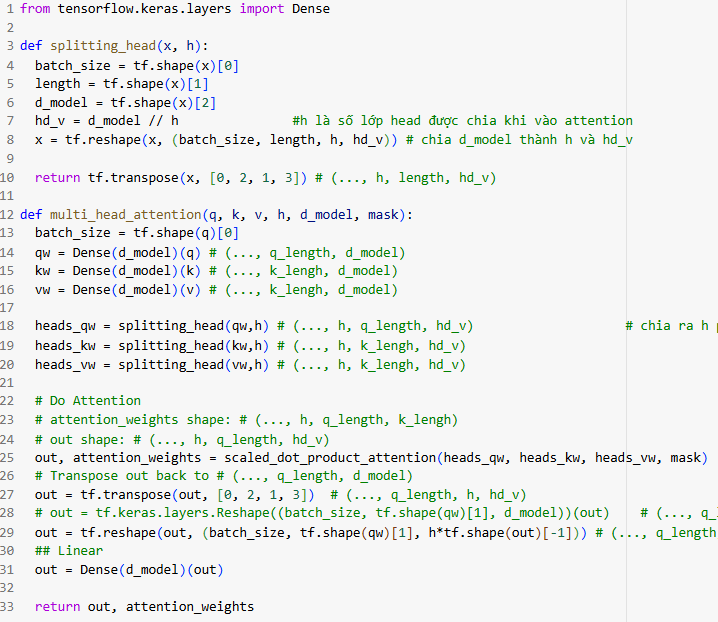
Mô tả được tạo tự động

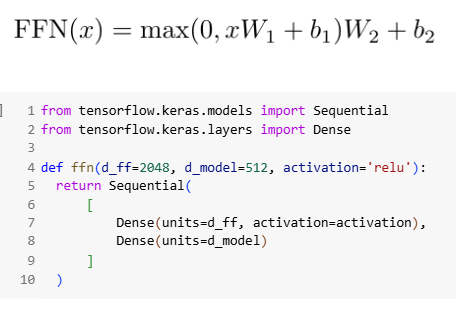






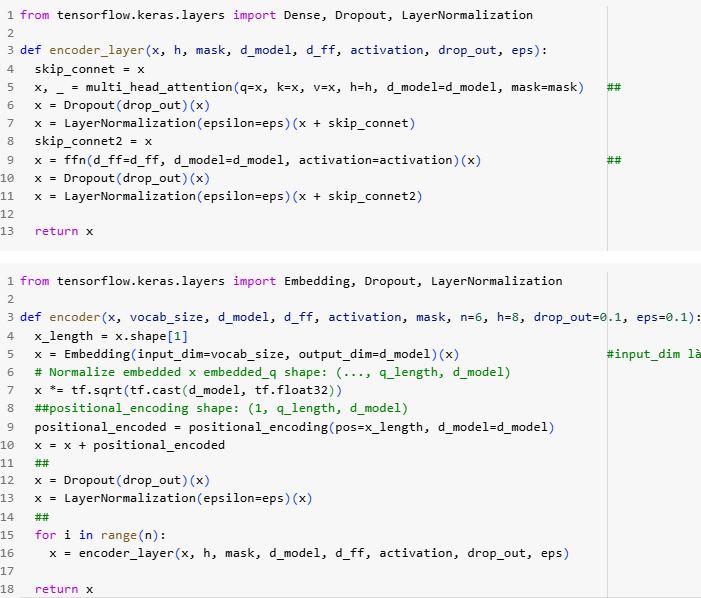


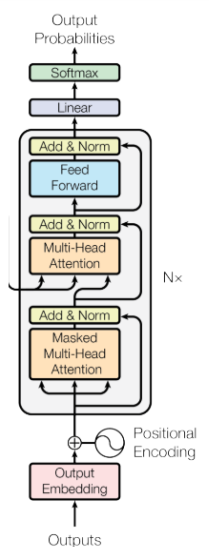


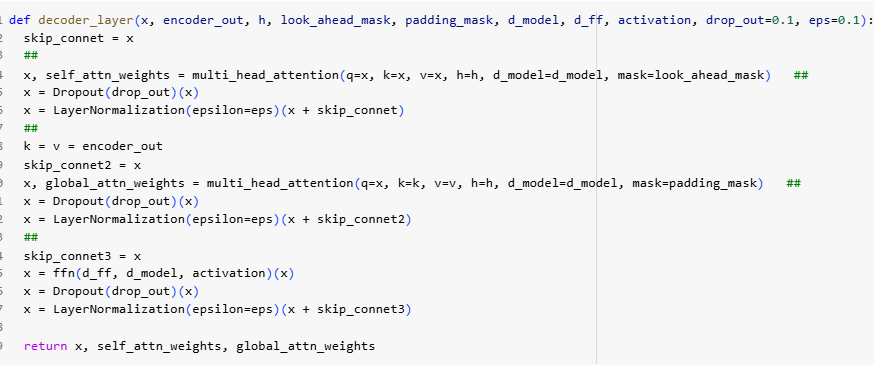


Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự động







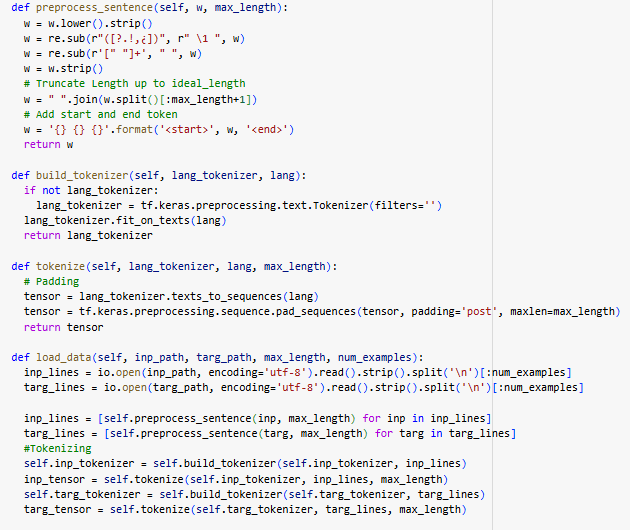
Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

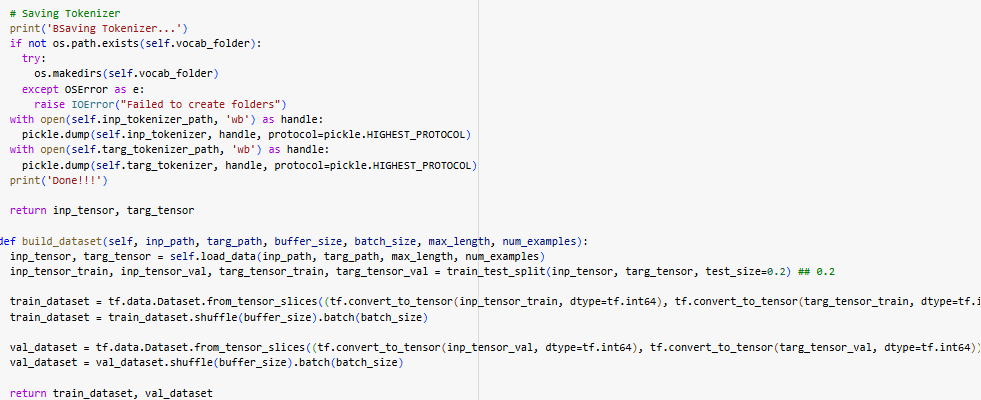
Mô tả được tạo tự động



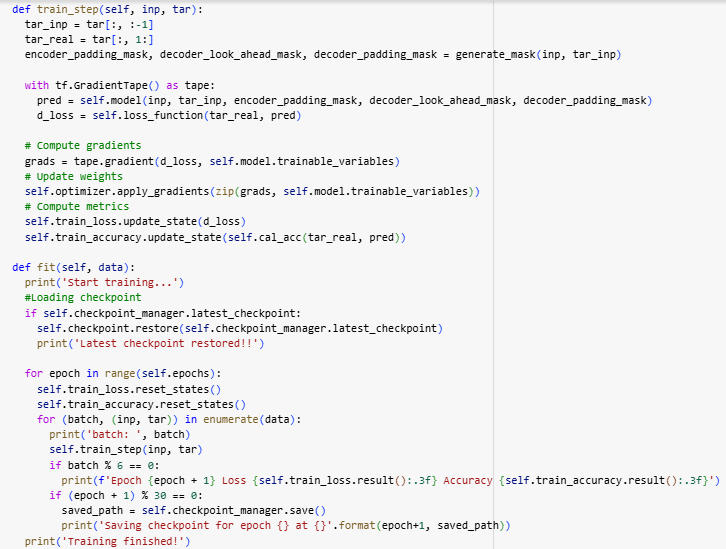


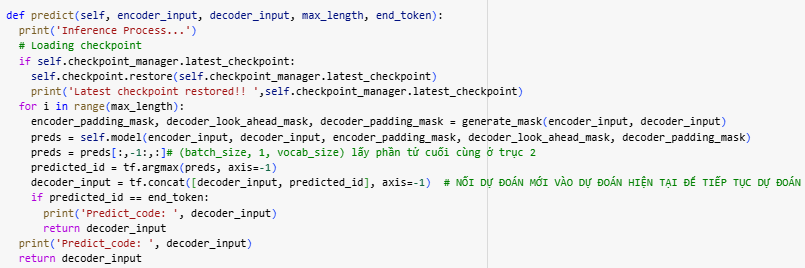


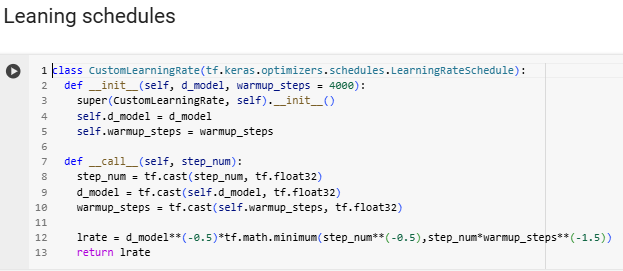


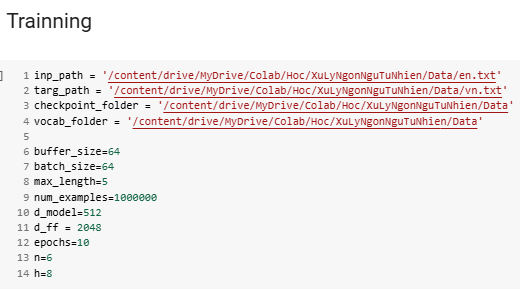


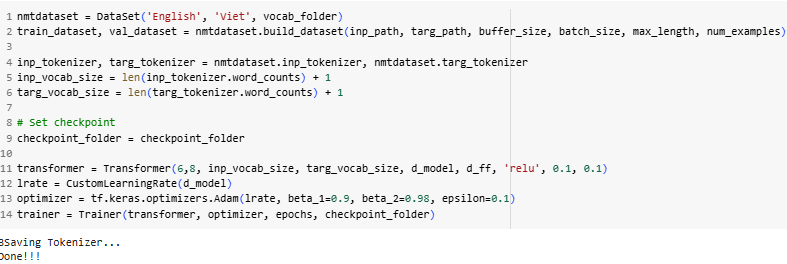






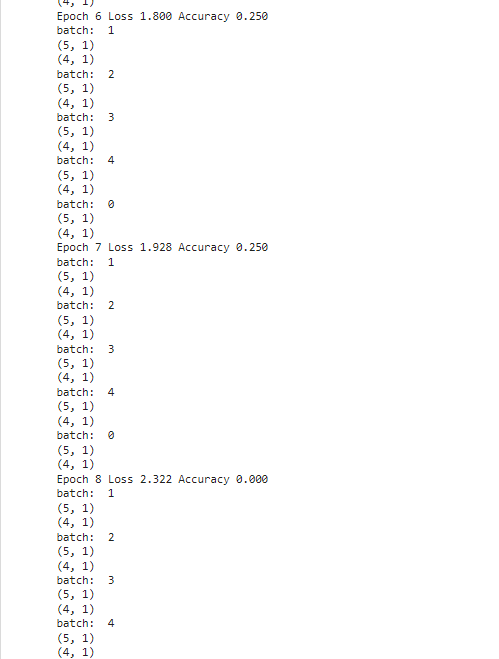




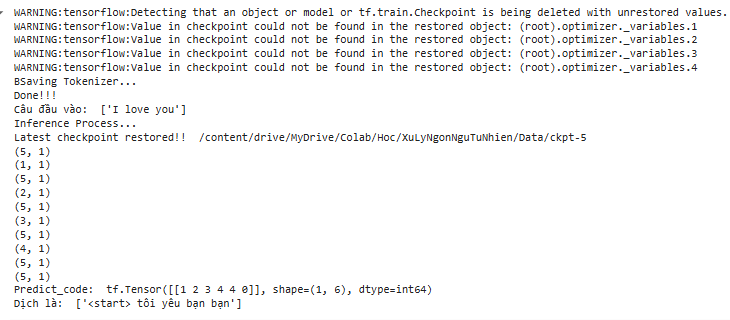


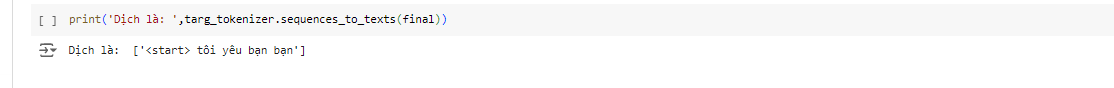
Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động









1. KẾT LUẬN

Trong công trình này, Transformer đã được giới thiệu như là mô hình chuyển đổi chuỗi đầu tiên dựa hoàn toàn vào cơ chế attention. Bằng cách thay thế các lớp tuần tự thường thấy trong kiến trúc mã hóa-giải mã bằng multi-headed self-attention, Transformer đã mở ra một hướng tiếp cận mới, hiệu quả hơn trong việc xử lý dữ liệu.

Đối với các nhiệm vụ dịch, Transformer không chỉ đơn giản hóa kiến trúc mà còn cải thiện đáng kể hiệu suất huấn luyện. Mô hình này có thể được huấn luyện nhanh hơn so với các kiến trúc dựa trên các lớp tuần tự hoặc tích chập, giúp giảm thiểu thời gian và tài nguyên cần thiết. Trên cả hai nhiệm vụ dịch từ tiếng Anh sang tiếng Đức và từ tiếng Anh sang tiếng Pháp trên bộ dữ liệu WMT 2014, Transformer đã đạt được một state-of-the-art mới. Đặc biệt, trong nhiệm vụ dịch từ tiếng Anh sang tiếng Đức, mô hình tốt nhất đã vượt qua tất cả các bộ sưu tập trước đây được báo cáo, chứng minh khả năng vượt trội của Transformer trong việc xử lý và dịch ngôn ngữ tự nhiên.

Những thành tựu này đặt nền móng cho tương lai của các mô hình dựa trên attention và mở ra nhiều hướng nghiên cứu mới. Có kế hoạch áp dụng Transformer vào các nhiệm vụ khác ngoài dịch ngôn ngữ, như tổng hợp văn bản, phân loại cảm xúc, và trả lời câu hỏi. Hơn nữa, việc mở rộng Transformer để giải quyết các vấn đề liên quan đến các dạng đầu vào và đầu ra khác ngoài văn bản như ảnh, âm thanh và video cũng đang được nghiên cứu. Điều này bao gồm việc phát triển các cơ chế attention hạn chế địa phương để xử lý hiệu quả các đầu vào và đầu ra lớn, đáp ứng nhu cầu của các ứng dụng đa phương tiện phức tạp.

Nhìn chung, Transformer đã chứng minh được tiềm năng to lớn và khả năng thay đổi cách tiếp cận truyền thống trong lĩnh vực xử lý ngôn ngữ tự nhiên. Các kết quả đạt được cho thấy sự ưu việt của mô hình này không chỉ trong các nhiệm vụ hiện tại mà còn trong khả năng mở rộng và ứng dụng vào nhiều lĩnh vực khác nhau. Với sự phát triển liên tục và những cải tiến mới, Transformer hứa hẹn sẽ tiếp tục đóng vai trò quan trọng trong việc giải quyết các thách thức trong NLP và trí tuệ nhân tạo.

Nghiên cứu này đã chỉ ra rằng việc sử dụng attention có thể cải thiện hiệu suất của các mô hình học sâu trong nhiều ứng dụng khác nhau. Sự thành công của Transformer là minh chứng rõ ràng cho sự tiến bộ trong việc thiết kế các kiến trúc mô hình, mở ra một kỷ nguyên mới cho nghiên cứu và phát triển trong lĩnh vực này. Hy vọng rằng những khám phá này sẽ thúc đẩy sự đổi mới và cải tiến liên tục trong các phương pháp và ứng dụng của xử lý ngôn ngữ tự nhiên, mang lại nhiều lợi ích thực tiễn và rộng lớn cho cộng đồng khoa học và công nghệ.

TÀI LIỆU THAM KHẢO (IEEE)

1. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," \*arXiv preprint arXiv:1706.03762\*, 2017. DOI: 10.48550/arXiv.1706.03762.
2. K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," \*arXiv preprint arXiv:1412.6596\*, 2014. DOI: 10.48550/arXiv.1412.6596.
3. P. D. Khanh, "Attention Layer," \*Pham Dinh Khanh's Blog\*, 2019. [Online]. Available: <https://phamdinhkhanh.github.io/2019/06/18/AttentionLayer.html>. [Accessed: 07-Jul-2024].
4. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in \*Proceedings of the 3rd International Conference on Learning Representations (ICLR)\*, San Diego, CA, 2015. [Online]. Available: <http://machinelearning.org/archive/icml2008/papers/391.pdf>. [Accessed: 07-Jul-2024].
5. A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," \*arXiv preprint arXiv:1801.06146\*, 2018. DOI: 10.48550/arXiv.1801.06146.
6. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," \*OpenAI\*, 2019. [Online]. Available:<https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.> [Accessed: 07-Jul-2024].
7. A. Radford, M. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Better Language Models and Their Implications," \*OpenAI\*, 2019. [Online]. Available: <https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.> [Accessed: 07-Jul-2024].
8. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," \*arXiv preprint arXiv:2005.14165\*, 2020. DOI: 10.48550/arXiv.2005.14165.