TẬP ĐOÀN BƯU CHÍNH VIỄN THÔNG VIỆT NAM VNPT

CÔNG TY CÔNG NGHỆ THÔNG TIN VNPT-IT

----- ☙ ❧ -----

CÔNG TY CÔNG NGHỆ THÔNG TIN VNPT

VNPT INFORMATION TECHNOLOGY COMPANY

BÁO CÁO TIẾN ĐỘ TUẦN 1

(24/04 – 28/04)

Attention mechanism, Transformer (Positional embeddings, Layer normalization, Self-attention layer)

Hướng dẫn: Nguyễn Bình Minh

Thực hiện tiến độ: Đào Thành Mạnh

Hà Nội, Ngày 28 tháng 6 năm 2024

MỤC LỤC

1. Tiến độ công việc

1.1. Tìm hiểu về Attention mechanism (Cơ chế chú ý)
- Giới thiệu về Attention mechanism
- Tìm hiểu về Attention mechanism:
  + Đưa ra ví dụ về Attention mechanism
  + Tìm hiểu về Unified attention model
- Phân loại Attention mechanism
- Tìm hiểu về kiến trúc Attention mechanism
- Ứng dụng của Attention mechanism trong NLP

1.2. Tìm hiểu về Transformer (Positional embeddings, Layer normalization, Self-attention layer)
- Giới thiệu về Transformer
- Kiến trúc Transformer
- Positional embeddings
- Layer normalization
- Self-attention layer
- Ứng dụng của Transformer trong NLP

2. Nội dung chi tiết

2.1. Attention mechanism

2.1.1 Giới thiệu về Attention mechanism

Chú ý là một chức năng nhận thức phức tạp không thể thiếu đối với con người. Một đặc tính quan trọng của nhận thức là con người không có xu hướng xử lý toàn bộ thông tin cùng một lúc. Thay vào đó, con người có xu hướng tập trung có chọn lọc vào một phần thông tin khi nào và ở đâu cần thiết, nhưng đồng thời lại bỏ qua những thông tin có thể nhận biết được khác. Ví dụ, con người thường không nhìn thấy tất cả các cảnh từ đầu đến cuối khi nhận thức bằng mắt, mà thay vào đó, quan sát và chú ý đến những phần cụ thể khi cần thiết. Khi con người nhận thấy một cảnh thường có thứ họ muốn quan sát ở một phần nào đó, họ sẽ học cách tập trung vào phần đó khi những cảnh tương tự xuất hiện trở lại và tập trung chú ý hơn vào phần hữu ích. Đây là phương tiện để con người nhanh chóng lựa chọn thông tin có giá trị cao từ lượng thông tin khổng lồ sử dụng tài nguyên xử lý hạn chế. Cơ chế chú ý cải thiện đáng kể hiệu quả và độ chính xác của việc xử lý thông tin nhận thức.

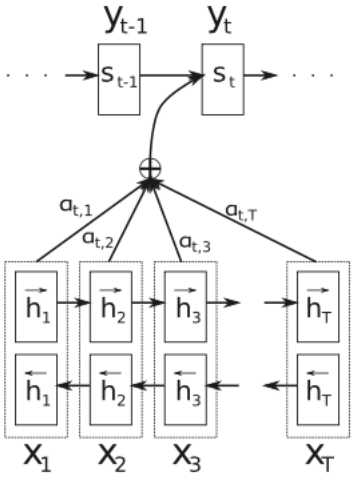
Cơ chế chú ý của con người có thể được chia thành hai phân loại theo cách thức tạo ra nó:

- Loại đầu tiên là sự chú ý vô thức từ dưới lên, được gọi là sự chú ý dựa trên độ mặn, được thúc đẩy bởi các kích thích bên ngoài. Ví dụ, mọi người có nhiều khả năng nghe thấy giọng nói lớn hơn trong một cuộc trò chuyện. Nó tương tự như cơ chế gộp tối đa và gating [4,5] trong học sâu, chuyển các giá trị phù hợp hơn (tức là các giá trị lớn hơn) sang bước tiếp theo.
- Loại thứ hai là sự chú ý có ý thức từ trên xuống, được gọi là sự chú ý tập trung. Sự chú ý tập trung đề cập đến sự chú ý có mục đích được xác định trước và dựa vào các nhiệm vụ cụ thể. Nó cho phép con người tập trung sự chú ý vào một đối tượng nhất định một cách có ý thức và tích cực. Hầu hết các cơ chế chú ý trong deep learning đều được thiết kế theo các nhiệm vụ cụ thể sao cho hầu hết đều là sự chú ý tập trung. Cơ chế chú ý được giới thiệu trong bài viết này thường đề cập đến sự chú ý tập trung ngoại trừ những trường hợp đặc biệt.

2.1.2. Tìm hiểu về Attention mechanism

2.1.2.1. Ví dụ về Attention mechanism: RNNsearch

RNNsearch lần đầu tiên áp dụng cơ chế chú ý vào tác vụ dịch máy. RNNsearch bao gồm mạng nơ- ron tái phát hai chiều (BiRNN) [58] làm bộ mã hóa và bộ giải mã mô phỏng việc tìm kiếm thông qua câu nguồn khi giải mã một bản dịch, như được minh họa trong Hình 1.



Hình 1. Illustration of a single step of decoding in attention-based neural machine translation

Một thiếu sót của RNN thông thường là chúng chỉ sử dụng bối cảnh trước đó. BiRNN có thể được huấn luyện bằng cách sử dụng tất cả các dữ liệu đầu vào có sẵn thông tin trong quá khứ và tương lai của một khung thời gian cụ thể.

Bộ giải mã bao gồm một khối chú ý và mạng thần kinh tái phát (RNN). Chức năng của khối chú ý là tính toán vectơ ngữ cảnh c đại diện cho mối quan hệ ngữ cảnh giữa ký hiệu đầu ra hiện tại và mỗi số hạng của toàn bộ trình tự đầu vào.

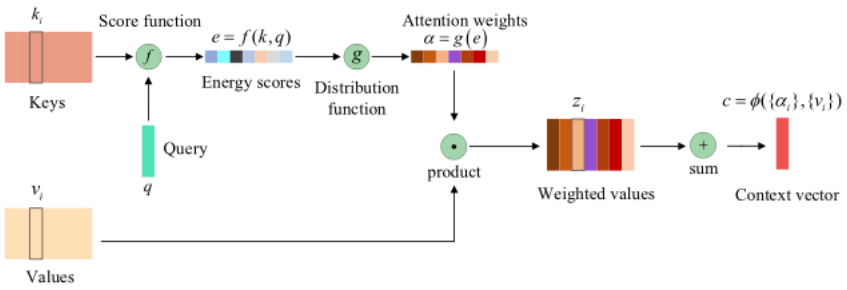
Bằng cách này, thông tin của câu nguồn có thể được phân bổ thành toàn bộ chuỗi thay vì mã hóa tất cả thông tin thành một vectơ có độ dài cố định thông qua bộ mã hóa, trong khi bộ giải mã có thể lấy nó một cách có chọn lọc ở mỗi bước thời gian. Công thức này cho phép mạng lưới thần kinh tập trung vào các yếu tố liên quan của đầu vào khác với những phần không liên quan.

2.1.2.2. Tìm hiểu về Unified attention model

Sau khi áp dụng cơ chế chú ý vào các tác vụ dịch máy, các biến thể mô hình chú ý được sử dụng trong các ứng dụng khác nhau tên miền đã phát triển nhanh chóng. Nói chung, quá trình thực hiện cơ chế chú ý có thể được chia thành hai bước:

- Một là tính toán sự phân bố sự chú ý trên thông tin đầu vào
- Hai là tính toán vectơ ngữ cảnh theo phân phối sự chú ý.

Hình 2 thể hiện mô hình chú ý thống nhất, bao gồm phần cốt lõi được chia sẻ bởi hầu hết mô hình chú ý được tìm thấy trong các tài liệu được khảo sát:



Hình 2. Kiến trúc của mô hình chú ý thống nhất.

Khi tính toán phân bổ sự chú ý, mạng lưới thần kinh đầu tiên mã hóa tính năng dữ liệu nguồn dưới dạng K, được gọi là khóa. K có thể được thể hiện dưới nhiều hình thức khác nhau tùy theo nhiệm vụ cụ thể và cấu trúc thần kinh. Ví dụ, K có thể là đặc điểm của một chứng chỉ vùng hình ảnh, phần nhúng từ của tài liệu hoặc trạng thái ẩn của RNN, như xảy ra với các chú thích trong RNNsearch.

Ngoài ra, thông thường cần phải giới thiệu một vectơ biểu diễn liên quan đến nhiệm vụ q, truy vấn, giống như trạng thái ẩn trước đó của đầu ra st1 trong RNNsearch.

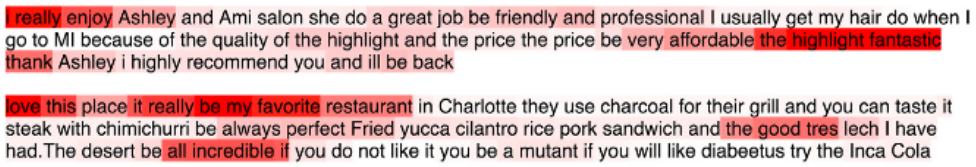
Trong một số trường hợp, q cũng có thể ở dạng ma trận hoặc hai vectơ tùy theo nhiệm vụ cụ thể. Sau đó, mạng lưới thần kinh tính toán mối tương quan giữa các truy vấn và khóa thông qua hàm tính điểm f (còn gọi là hàm năng lượng hoặc hàm tương thích) để

thu được năng lượng điểm e phản ánh tầm quan trọng của truy vấn liên quan đến khóa trong việc quyết định đầu ra tiếp theo.

Table 1
Summary of score function f. Here, $\boldsymbol{k}$ is an element of $\boldsymbol{K}$, $\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{W}, \boldsymbol{W_1}, \boldsymbol{W_2}$ are learnable parameters, d_k is the dimension of the input vector. The *act* is a nonlinear activation function, such as tanh and ReLU.

Name	Equation	Ref.
Additive	$f(q,k) = v^{\top} act(W_1 k + W_2 q + b)$	[15]
Multiplicative (dot-product)	$f(q,k) = q^{\top} k$	[15]
Scaled multiplicative	$f(q,k) = \frac{q^{\top} k}{\sqrt{d_k}}$	[16]
General	$f(q,k) = q^{\top} W k$	[14]
Concat	$f(q,k) = v^{\top} act(W[k;q] + b)$	[14]
Location-based	$f(q,k) = f(q)$	[14]
Similarity	$f(q,k) = \frac{q \cdot k}{\|q\| \cdot \|k\|}$	[60]

Nhìn chung, các cơ chế chú ý trong học sâu được gắn với các mô hình mạng lưới thần kinh để nâng cao khả năng xử lý thông tin của họ. Vì thế, thật khó để đánh giá hiệu suất của cơ chế chú ý mà không cần nghiên cứu sâu các mô hình học tập. Một cách tiếp cận phổ biến là nghiên cứu cắt bỏ có nghĩa là để phân tích khoảng cách hiệu suất giữa các mô hình có/không có cơ chế chú ý Ngoài ra, cơ chế chú ý có thể được đánh giá bằng cách hình dung mức độ chú ý (như thể hiện trong Hình 3), nhưng cách này không thể định lượng được.



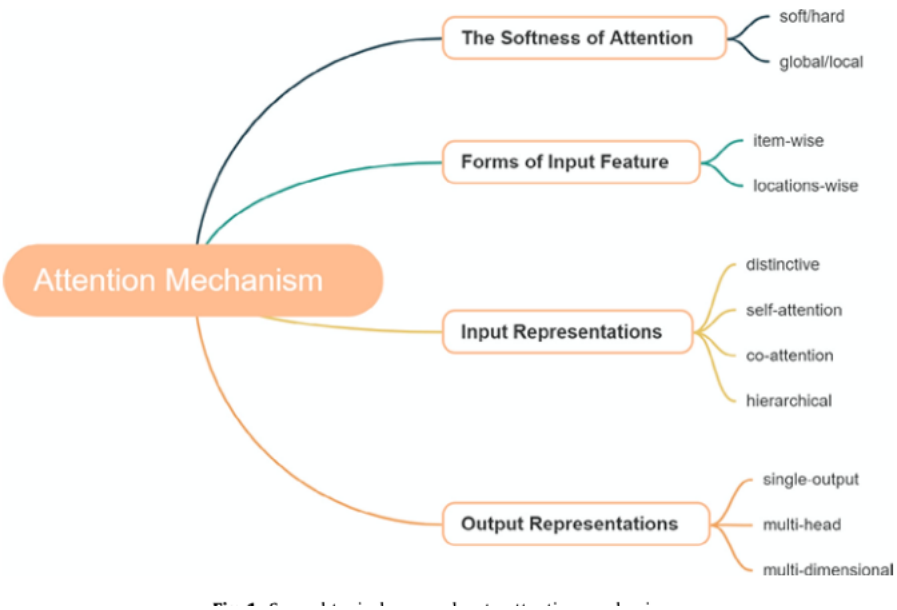
Hình 3. Sơ đồ nhiệt của các bài đánh giá, màu sắc đậm hơn cho thấy mức độ chú ý cao hơn.

2.1.3. Phân loại Attention mechanism

Trong phần trước, chúng tôi đã tóm tắt sự chú ý chung mô hình và giải thích chi tiết từng bước thực hiện cơ chế chú ý. Là một phương pháp để cải thiện việc xử lý thông tin khả năng của mạng lưới thần kinh, cơ chế chú ý có thể được áp dụng cho hầu hết các mô hình trong các lĩnh vực học sâu khác nhau. Mặc dù nguyên tắc của các mô hình chú ý là như nhau nhưng các nhà nghiên cứu đã đưa ra một số sửa đổi và cải tiến cơ chế chú ý trong để thích ứng tốt hơn với các nhiệm vụ cụ thể. Chúng tôi phân loại các cơ chế chú ý theo bốn tiêu chí như trong Bảng 2 và Hình 4:

Table 2
Four criteria for categorizing attention mechanism, and types of attention within each criterion.

Criterion	Type
The Softness of Attention	Aoft/hard, global/local
Forms of Input Feature	Item-wise, location-wise
Input Representations	Distinctive, self, co-attention, hierarchical
Output Representations	Single-output, multi-head, multi-dimensional



Hình 4. Một số cách tiếp cận điển hình đối với cơ chế chú ý.

Trong phần này, chúng tôi trình bày chi tiết về các loại cơ chế chú ý khác nhau trong từng tiêu chí thông qua việc xem xét một số bài báo chuyên đề.

Ngoài ra, chúng tôi muốn nhấn mạnh rằng các cơ chế chú ý ở các tiêu chí khác nhau không loại trừ lẫn nhau, vì vậy có có thể là sự kết hợp của nhiều tiêu chí trong một mô hình chú ý.

2.1.3.1. The Softness of Attention (Mức độ chú ý)

Sự chú ý được đề xuất bởi Bahdanau như đã đề cập ở trên thuộc về sự chú ý mềm (xác định), sử dụng mức trung bình có trọng số của tất cả các khóa để xây dựng vectơ ngữ cảnh. Đối với sự chú ý mềm, mô-đun chú ý có thể phân biệt được theo đầu vào, do đó toàn bộ hệ thống vẫn có thể được huấn luyện bằng các phương pháp truyền ngược tiêu chuẩn.

So với mô hình chú ý mềm, mô hình chú ý cứng là về mặt tính toán ít tốn kém hơn vì nó không cần tính toán trọng số chú ý của tất cả các phần tử tại mỗi thời điểm. Tuy

nhiên, việc đưa ra quyết định khó khăn ở mỗi vị trí của đặc điểm đầu vào sẽ khiến mô-đun không thể phân biệt và khó tối ưu hóa, vì vậy toàn bộ hệ thống có thể được huấn luyện bằng cách tối đa hóa giới hạn dưới biến thiên gần đúng hoặc tương đương bằng REINFORCE.

Trên cơ sở này, thu hút sự chú ý toàn cầu và cơ chế chú ý cục bộ cho dịch máy. Toàn cầu sự chú ý tương tự như mức độ chú ý. Sự chú ý của địa phương có thể được xem như một sự kết hợp thú vị giữa sự chú ý cứng rắn và mềm mỏng, trong đó chỉ một tập hợp con các từ nguồn được xem xét ở một mức độ nào đó. thời gian. Cách tiếp cận này ít tốn kém về mặt tính toán hơn so với toàn cầu chú ý hoặc chú ý nhẹ nhàng. Đồng thời, không giống như sự chú ý chăm chú, cách tiếp cận này có thể được phân biệt ở hầu hết mọi nơi, giúp việc này trở nên dễ dàng hơn để thực hiện và đào tạo.

2.1.3.2. Forms of Input Feature (Các thuộc tính đầu vào)

Các cơ chế chú ý có thể được chia thành từng mục và vị trí khôn ngoan tùy theo tính năng đầu vào có phải là một chuỗi hay không của các mặt hàng đó. Sự chú ý theo từng mục yêu cầu đầu vào là các mục rõ ràng hoặc một bước tiền xử lý bổ sung được thêm vào để tạo ra một chuỗi các mục từ dữ liệu nguồn. Ví dụ, mục có thể là một từ được nhúng trong RNNsearch hoặc một bản đồ đặc trưng trong SENet. Trong mô hình chú ý, bộ mã hóa mã hóa từng mục dưới dạng một mã riêng biệt và gán các trọng số khác nhau với chúng trong quá trình giải mã.

Ngược lại, sự chú ý về vị trí nhằm vào các nhiệm vụ khó có được các mục đầu vào riêng biệt, và nói chung cơ chế chú ý như vậy được sử dụng trong các nhiệm vụ trực quan. Ví dụ: bộ giải mã xử lý phần cắt đa độ phân giải của hình ảnh đầu vào ở mỗi bước hoặc chuyển đổi vùng liên quan đến nhiệm vụ thành một tư thế chuẩn, được mong đợi để đơn giản hóa việc suy luận ở các lớp tiếp theo.

Một điểm khác biệt nữa là cách tính toán khi kết hợp với cơ chế chú ý mềm/cứng. Sự chú ý mềm về mặt hàng tính toán trọng số cho từng mục rồi tạo tổ hợp tuyến tính của chúng. Sự chú ý mềm theo vị trí chấp nhận toàn bộ bản đồ đặc trưng làm đầu vào và tạo ra một phiên bản được chuyển đổi thông qua mô-đun chú ý. Thay vì kết hợp tuyến tính tất cả các mục, sự chú ý kỹ càng đến từng mục một cách ngẫu nhiên chọn một hoặc một số các mục dựa trên xác suất của chúng. Sự chú ý kỹ lưỡng về vị trí sẽ ngẫu nhiên chọn một tiểu vùng làm đầu vào và vị trí của tiểu vùng được chọn sẽ được mô-đun chú ý tính toán.

2.1.3.3. Input Representations (Biểu diễn đầu vào)

Có hai đặc điểm về biểu diễn đầu vào trong hầu hết các mô hình chú ý nêu trên:

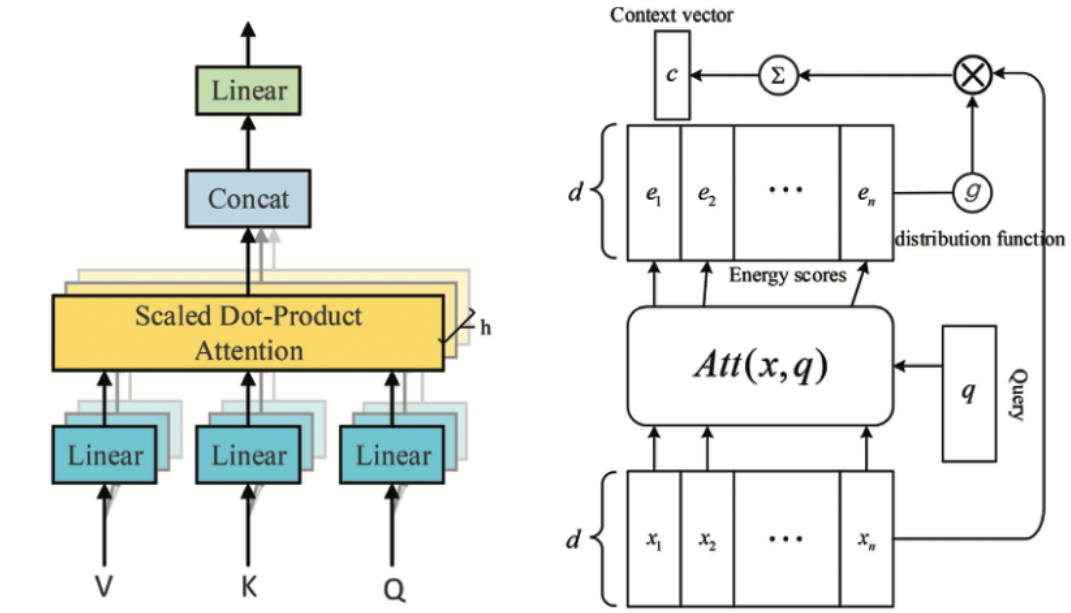
1) Các mô hình này bao gồm một đầu vào duy nhất và một chuỗi đầu ra tương ứng;

2) Các phím và các truy vấn thuộc về hai chuỗi độc lập.

Trường hợp này sự chú ý được gọi là sự chú ý đặc biệt [74]. Ngoài ra, cơ chế chú ý còn có nhiều dạng biểu diễn đầu vào khác nhau.

2.1.3.4. Output Representations (Biểu diễn đầu ra)

Trong phần này, chúng ta thảo luận về các loại biểu diễn đầu ra khác nhau trong các mô hình chú ý Trong số đó, cái phổ biến là đầu ra đơn sự chú ý đề cập đến một biểu diễn tính năng duy nhất trong mỗi lần. Cụ thể, điểm năng lượng được biểu thị bằng một và chỉ một vectơ tại mỗi bước thời gian. Tuy nhiên, trong một số trường hợp, việc sử dụng một biểu diễn đối tượng duy nhất có thể không thể hoàn thành tốt các nhiệm vụ tiếp theo. Tiếp theo, chúng tôi mô tả hai mô hình chú ý đa đầu ra: đa đầu và đa chiều như minh họa trong Hình 5:



(a) Multi-head attention [16] (b) Multi-dimensional attention [92]

Hình 5. Minh họa về biểu diễn nhiều đầu ra.

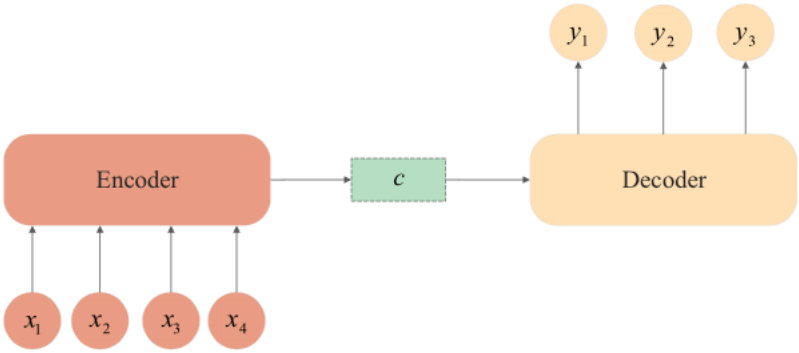
Trong nhiều ứng dụng của mạng nơ ron tích chập, người ta đã chứng minh rằng nhiều kênh có thể thể hiện dữ liệu đầu vào một cách toàn diện hơn so với một kênh. Ngoài ra, trong các mô hình chú ý, trong một số trường hợp, việc sử dụng phân phối chú ý duy nhất của chuỗi đầu vào có thể không đủ cho các tác vụ xuôi dòng đề xuất sự chú ý nhiều đầu chiếu tuyến tính chuỗi đầu vào (Q; K; V) tới nhiều không gian con dựa trên các tham số có thể học được, sau đó áp dụng sự chú ý tích số chấm theo tỷ lệ cho biểu diễn của nó trong mỗi không gian con và cuối cùng ghép nối đầu ra của chúng . Bằng cách này, nó cho phép mô hình cùng tham gia vào thông tin từ các không gian con biểu diễn khác nhau ở các vị trí khác nhau.

2.1.4. Tìm hiểu về kiến trúc Attention mechanism

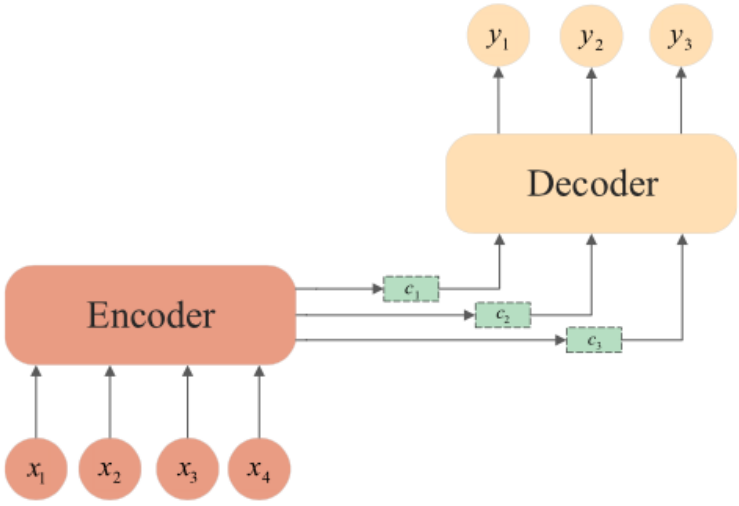
Trong phần này, chúng tôi mô tả ba kiến trúc mạng nơ-ron được sử dụng kết hợp với sự chú ý. Đầu tiên chúng tôi giải thích chi tiết về khung mã hóa-giải mã được sử dụng bởi hầu hết các mô hình chú ý. Sau đó chúng tôi mô tả kiến trúc mạng đặc biệt của các mô hình chú ý, mạng bộ nhớ, được trang bị bộ nhớ ngoài dài hạn ký ức. Cuối cùng, chúng tôi giới thiệu các cấu trúc mạng lưới thần kinh đặc biệt kết hợp với cơ chế chú ý, trong đó RNN không được sử dụng trong quá trình nắm bắt các phụ thuộc đường dài.

2.1.4.1. Encoder-decoder (Bộ mã hoá-giải mã)

Bộ mã hóa-giải mã (như trong Hình 6) là một khung chung dựa trên mạng lưới thần kinh, nhằm mục đích xử lý việc ánh xạ giữa đầu vào và đầu ra có cấu trúc cao:



Hình 6. Minh họa khung mã hóa-giải mã không có cơ chế chú ý.

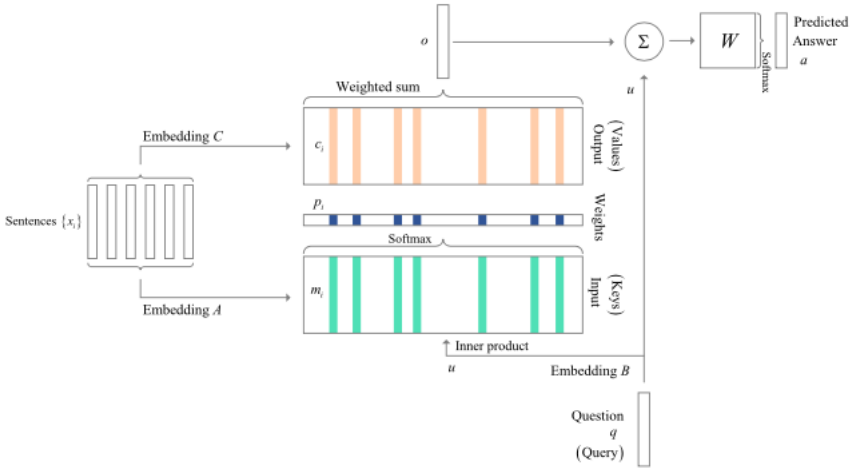


Hình 7. Minh họa khung mã hóa-giải mã bằng cách sử dụng chú ý cơ chế.

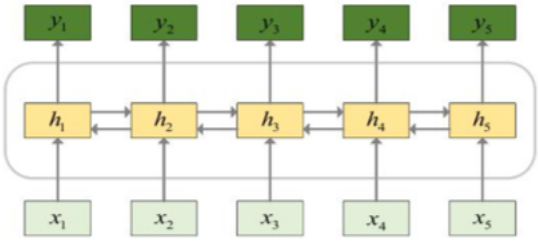
2.1.4.2. Memory networks (Mạng bộ nhớ)

Ngoài khung mã hóa-giải mã ở phần trước phần này, cơ chế chú ý cũng được sử dụng kết hợp với mạng bộ nhớ. Lấy cảm hứng từ cơ chế hoạt động của bộ não con người xử lý tình trạng quá tải thông tin, mạng bộ nhớ sẽ giới thiệu bộ nhớ ngoài bổ sung vào mạng lưới thần kinh. Cụ thể, các mạng bộ nhớ lưu một số thông tin liên quan đến tác vụ trong bộ nhớ phụ bằng cách đưa vào các phần phụ trợ bên ngoài đơn vị bộ nhớ và sau đó đọc nó khi cần thiết, điều này không chỉ tăng hiệu quả dung lượng mạng mà còn cải thiện hiệu quả tính toán mạng so với sự quan tâm chung cơ chế, mạng bộ nhớ thay thế khóa bằng bộ nhớ phụ dài hạn và sau đó khớp nội dung thông qua cơ chế chú ý.

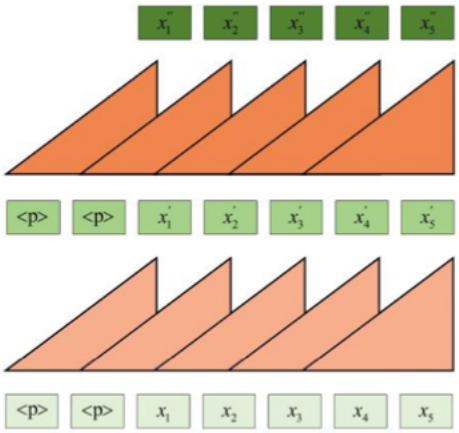
Một ví dụ nổi tiếng là bộ nhớ đầu cuối có khả năng phân biệt mạng, có thể đọc thông tin từ bên ngoài thông tin cuối cùng nhiều lần. Ý tưởng cốt lõi là chuyển đổi đầu vào được đặt thành hai đơn vị bộ nhớ ngoài, một đơn vị để đánh địa chỉ, và một cái khác cho đầu ra, như trong Hình 8 và 9:



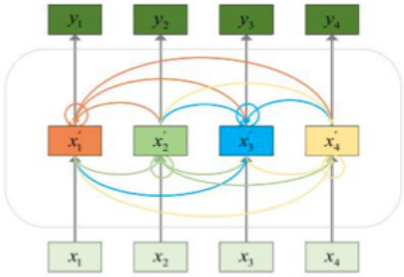
Hình 8. Phiên bản một lớp của mạng bộ nhớ đầu cuối. Ở đây, câu hỏi, đầu vào và đầu ra tương ứng với truy vấn, khóa và giá trị trong sự chú ý thống nhất mô hình tương ứng.



(a) Single-layer RNN model structure.



(b) Multi-layer CNN model structure.



(c) Single-layer self-attention model structure.

Hình 10. Minh họa ba cấu trúc được sử dụng để nắm bắt sự phụ thuộc khoảng cách xa.

Bộ nhớ đầu cuối mạng có thể được coi là một dạng của sự chú ý: cặp khóa-giá trị cơ chế chú ý Khác với sự chú ý thông thường, thay vì chỉ mô hình hóa sự chú ý trên một chuỗi duy nhất, họ sử dụng hai đơn vị bộ nhớ bên ngoài để mô hình hóa nó trên một cơ sở dữ liệu lớn về các chuỗi. Nói cách khác, chúng ta có thể coi cơ chế chú ý như một giao diện tách biệt việc lưu trữ thông tin khỏi việc tính toán, để dung lượng mạng có thể được tăng lên đáng kể chỉ với một lượng nhỏ tăng tham số mạng.

2.1.4.3. Networks without RNNs (Mạng không có RNN)

Như đã đề cập ở trên, cả bộ mã hóa và bộ giải mã trong khung mã hóa-giải mã có thể được triển khai theo nhiều cách như trong Hình 9. RNN dựa trên kiến trúc bộ mã hóa-giải mã thường tính toán nhân tử cùng với vị trí ký hiệu của trình tự đầu vào và đầu ra. Bản chất tuần tự vốn có này dẫn đến tính toán kém hiệu quả vì quá trình xử lý không thể song song. Mặt khác, việc nắm bắt các đối tượng phụ thuộc ở khoảng cách xa là điều cần thiết vì cơ chế chú ý trong bộ mã hóa- giải mã cần thu được thông tin theo ngữ cảnh. Tuy nhiên, độ phức tạp tính toán của việc thiết lập khoảng cách xa sự phụ thuộc của chuỗi có

độ dài n qua RNN là O(n). Trong phần này, chúng tôi mô tả các cách triển khai khác của bộ mã hóa-giải mã khung kết hợp với cơ chế chú ý, loại bỏ thẻ RNN. Đề xuất kiến trúc bộ mã hóa-giải mã hoàn toàn dựa vào mạng lưới thần kinh tích chập kết hợp với cơ chế chú ý. Ngược lại với thực tế là các mạng lặp lại duy trì trạng thái ẩn của toàn bộ quá khứ, các mạng tích chập không dựa vào các tính toán của quá khứ trước đó bước thời gian, sao cho nó cho phép thực hiện song song trên từng phần tử trong một sự liên tiếp. Kiến trúc này cho phép mạng nắm bắt được sự phụ thuộc ở khoảng cách xa bằng cách xếp chồng nhiều lớp CNN, độ phức tạp tính toán trở thành O(n/k) đối với CNN nhiều lớp với kích thước hạt tích chập là k. Hơn nữa, sự tích chập này phương pháp có thể khám phá cấu trúc thành phần trong trình tự dễ dàng hơn vì sự biểu diễn có thứ bậc của nó.

Vaswani và cộng sự đề xuất một kiến trúc mạng khác là Transformer hoạt động hoàn toàn dựa vào cơ chế tự chú ý để tính toán các biểu diễn đầu vào và đầu ra của nó mà không cần dùng đến RNN hoặc CNN. Transformer bao gồm hai thành phần: lớp mạng chuyển tiếp nguồn cấp dữ liệu theo vị trí (FFN) và lớp chú ý nhiều đầu. FFN theo vị trí là mạng chuyển tiếp nguồn cấp dữ liệu được kết nối đầy đủ, được áp dụng riêng cho từng vị trí và giống hệt nhau. Phương pháp này có thể đảm bảo thông tin vị trí của từng ký hiệu trong chuỗi đầu vào trong quá trình hoạt động. Sự chú ý của nhiều đầu cho phép mô hình tập trung vào thông tin từ các không gian con biểu diễn khác nhau từ các vị trí khác nhau bằng cách xếp chồng nhiều lớp tự chú ý, giống như nhiều kênh của CNN. Ngoài khả năng song song hóa hơn, tính phức tạp của việc thiết lập sự phụ thuộc đường dài thông qua cơ chế tự chú ý là O(1).

2.1.5. Ứng dụng của Attention mechanism trong NLP

Trong phần này, trước tiên chúng tôi giới thiệu một số phương pháp chú ý được sử dụng trong các nhiệm vụ khác nhau của NLP và sau đó mô tả một số cách biểu diễn từ đào tạo trước phổ biến được triển khai với cơ chế chú ý cho các nhiệm vụ NLP.

Dịch máy neural sử dụng mạng thần kinh để dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác. Trong quá trình dịch thuật, việc căn chỉnh các câu trong các ngôn ngữ khác nhau là một vấn đề quan trọng, đặc biệt đối với các câu dài. Bahdanau và cộng sự đã giới thiệu cơ chế chú ý vào mạng lưới thần kinh để cải thiện khả năng dịch máy thần kinh bằng cách tập trung có chọn lọc vào các phần của câu nguồn trong quá trình dịch. Sau đó, một số công trình đã được đề xuất cải tiến, chẳng hạn như sự chú ý của địa phương [14], sự chú ý có giám sát , sự chú ý theo cấp bậc và sự chú ý của bản thân. Họ đã sử dụng các kiến trúc chú ý khác nhau để cải thiện sự liên kết của các câu và nâng cao hiệu suất dịch thuật.

Phân loại văn bản nhằm mục đích gán nhãn cho văn bản và có các ứng dụng rộng rãi bao gồm ghi nhãn chủ đề, phân loại ý kiến và phát hiện thư rác. Trong các nhiệm vụ phân loại này, sự chú ý của bản thân chủ yếu được sử dụng để xây dựng cách trình bày tài liệu hiệu quả hơn. Trên cơ sở đó, một số công trình đã kết hợp cơ chế tự chú ý với các

phương pháp chú ý khác, chẳng hạn như tự chú ý theo cấp bậc và tự chú ý đa chiều. Ngoài ra, các tác vụ này còn áp dụng các kiến trúc mô hình chú ý như Transformer và mạng bộ nhớ.

So khớp văn bản cũng là một vấn đề nghiên cứu cốt lõi trong NLP và truy xuất thông tin, bao gồm trả lời câu hỏi, tìm kiếm tài liệu, phân loại kế thừa, nhận dạng diễn giải và đề xuất kèm theo các bài đánh giá. Nhiều nhà nghiên cứu đã đưa ra các phương pháp tiếp cận mới kết hợp với khả năng hiểu sự chú ý, chẳng hạn như mạng bộ nhớ, chú ý hơn chú ý, chú ý bên trong, chú ý có cấu trúc và đồng chú ý.

Biểu diễn từ được đào tạo trước là thành phần chính trong nhiều mô hình hiểu ngôn ngữ thần kinh. Tuy nhiên, các nghiên cứu trước đây chỉ xác định được một cách nhúng cho cùng một từ, điều này không thể đạt được việc nhúng từ theo ngữ cảnh. Peters và cộng sự đã giới thiệu một cách tiếp cận chung về biểu diễn phụ thuộc vào ngữ cảnh với Bi-LSTM để giải quyết vấn đề này. Lấy cảm hứng từ mô hình Transformer, các nhà nghiên cứu đã đề xuất các biểu diễn bộ mã hóa hai chiều từ Transformer (BERT) và phương pháp đào tạo trước tổng quát (GPT) theo các bộ phận mã hóa và giải mã. BERT là mô hình ngôn ngữ hai chiều và có hai nhiệm vụ đào tạo trước sau:

1) Mô hình ngôn ngữ mặt nạ (MLM). Nó chỉ đơn giản che giấu một số phần trăm mã thông báo đầu vào một cách ngẫu nhiên và sau đó dự đoán các mã thông báo bị che giấu đó.

2) Dự đoán câu tiếp theo. Nó sử dụng bộ phân loại nhị phân tuyến tính để xác định xem hai câu có được kết nối hay không. GPT là mô hình một chiều và phương pháp đào tạo của nó đại khái là sử dụng từ trước đó để dự đoán từ tiếp theo. Các thử nghiệm cho thấy những cải tiến lớn khi áp dụng chúng vào nhiều nhiệm vụ NLP.

2.2. Transformer

2.2.1. Giới thiệu về Transformer

Xử lý ngôn ngữ tự nhiên (NLP) là một lĩnh vực Học máy liên quan đến ngôn ngữ học nhằm xây dựng và phát triển Mô hình ngôn ngữ. Mô hình hóa ngôn ngữ (LM) xác định khả năng xảy ra chuỗi từ trong câu thông qua các kỹ thuật xác suất và thống kê. Vì ngôn ngữ của con người liên quan đến chuỗi từ nên các mô hình ngôn ngữ ban đầu dựa trên Mạng thần kinh tái phát (RNN).

Do RNN có thể dẫn đến sự biến mất và bùng nổ độ dốc trong các chuỗi dài nên các mạng tái phát được cải tiến như LSTM và GRU đã được sử dụng để cải thiện hiệu suất. Mặc dù đã được cải tiến nhưng LSTM vẫn thiếu khả năng hiểu khi có các chuỗi tương đối dài hơn. Điều này là do toàn bộ lịch sử được gọi là bối cảnh đang được xử lý bởi một vectơ trạng thái duy nhất. Tuy nhiên, tài nguyên tính toán lớn hơn dẫn đến sự xuất hiện của các kiến trúc mới gây ra sự gia tăng nhanh chóng của các mô hình NLP dựa trên Deep Learning.

Kiến trúc Transformer đột phá năm 2017 đã vượt qua giới hạn ngữ cảnh của LSTM thông qua cơ chế Chú ý. Ngoài ra, nó còn cung cấp thông lượng lớn hơn vì đầu vào được xử lý song song mà không phụ thuộc vào trình tự.

Những lần ra mắt tiếp theo của các mô hình dựa trên Transformer cải tiến như GPT-I và BERT vào năm 2018 hóa ra là một năm đỉnh cao đối với thế giới NLP. Những kiến trúc này được huấn luyện trên các tập dữ liệu lớn để tạo ra các mô hình được huấn luyện trước. Sau đó, việc học chuyển giao được sử dụng để tinh chỉnh các mô hình này cho các tính năng dành riêng cho nhiệm vụ, dẫn đến nâng cao hiệu suất đáng kể trên một số nhiệm vụ NLP. Những nhiệm vụ này bao gồm nhưng không giới hạn ở việc lập mô hình ngôn ngữ, phân tích tình cảm, trả lời câu hỏi và suy luận ngôn ngữ tự nhiên.

2.2.2. Kiến trúc Transformer

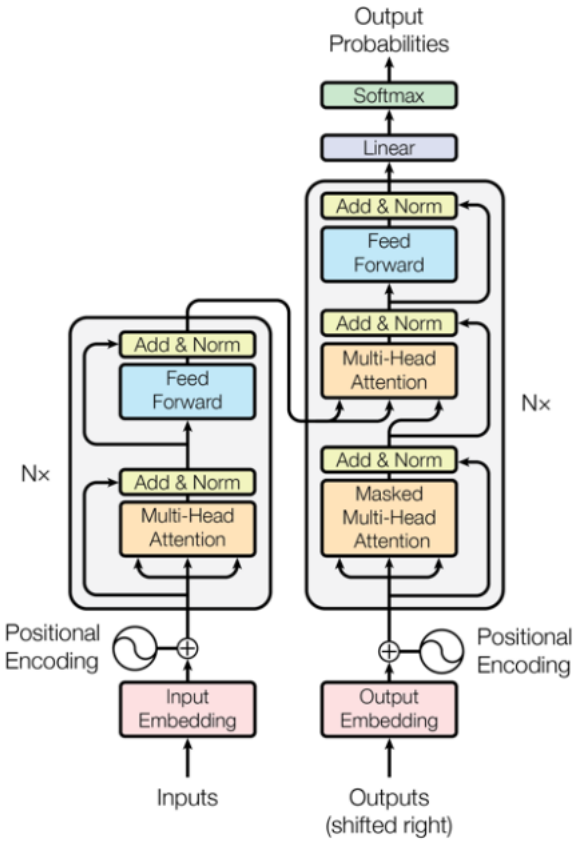


Figure 1: The Transformer - model architecture.

Hầu hết các mô hình truyền tải chuỗi thần kinh cạnh tranh đều có cấu trúc bộ mã hóa-giải mã. Ở đây, bộ mã hóa ánh xạ chuỗi đầu vào của các biểu diễn ký hiệu (x1, ...,

xn) thành một chuỗi các biểu diễn liên tục z = (z1, ..., zn). Cho z, bộ giải mã sau đó tạo ra một chuỗi đầu ra (y1, ..., ym) gồm các ký hiệu, mỗi phần tử một.

Ở mỗi bước, mô hình sẽ tự động hồi quy, sử dụng các ký hiệu được tạo trước đó làm đầu vào bổ sung khi tạo ký hiệu tiếp theo.

Transformer tuân theo kiến trúc tổng thể này bằng cách sử dụng các lớp tự chú ý và điểm thông minh xếp chồng lên nhau, được kết nối đầy đủ cho cả bộ mã hóa và bộ giải mã, tương ứng được hiển thị ở nửa bên trái và bên phải của Hình 1 .

2.2.2.1. Encoder

Bộ mã hóa bao gồm một chồng N = 6 lớp giống hệt nhau. Mỗi lớp có hai lớp con. Đầu tiên là cơ chế tự chú ý nhiều đầu, và thứ hai là cơ chế đơn giản, định vị- mạng chuyển tiếp nguồn cấp dữ liệu được kết nối đầy đủ khôn ngoan. Chúng tôi sử dụng kết nối dư [10] xung quanh mỗi lớp trong số hai lớp con, sau đó là chuẩn hóa lớp [1]. Nghĩa là, đầu ra của mỗi lớp con là LayerNorm(x + Sublayer(x)), trong đó Sublayer(x) là chức năng do chính lớp con đó thực hiện . Để tạo điều kiện thuận lợi cho các kết nối còn lại này, tất cả các lớp con trong mô hình cũng như các lớp nhúng tạo ra kết quả đầu ra có kích thước dmodel = 512.

2.2.2.2. Decoder

Bộ giải mã cũng bao gồm một chồng gồm N = 6 lớp giống hệt nhau. Ngoài hai lớp con trong mỗi lớp bộ mã hóa, bộ giải mã còn chèn một lớp con thứ ba, lớp này thực hiện sự chú ý nhiều đầu đối với đầu ra của ngăn xếp bộ mã hóa. Tương tự như bộ mã hóa, chúng tôi sử dụng các kết nối còn lại xung quanh mỗi lớp con, sau đó là chuẩn hóa lớp. Chúng tôi cũng sửa đổi lớp con tự chú ý trong ngăn xếp bộ giải mã để ngăn các vị trí tham gia vào các vị trí tiếp theo. Việc che giấu này , kết hợp với thực tế là các phần nhúng đầu ra được bù bởi một vị trí, đảm bảo rằng các dự đoán cho vị trí i chỉ có thể phụ thuộc vào các đầu ra đã biết ở các vị trí nhỏ hơn i.

2.2.3. Positional embeddings

Positional Embeddings là một kỹ thuật dùng để thêm thông tin vị trí vào các embeddings của từ trong chuỗi đầu vào. Do mô hình Transformer không sử dụng cấu trúc tuần tự như RNN hay LSTM, nên cần một cách để mô hình hiểu được vị trí tương đối của các từ trong câu.

Để đạt được mục đích này, chúng tôi thêm "mã hóa vị trí" vào phần nhúng đầu vào tại đáy của ngăn xếp bộ mã hóa và bộ giải mã. Các mã hóa vị trí có cùng chiều d_{model} như các phần nhúng, để cả hai có thể được tóm tắt. Có nhiều lựa chọn về mã hóa vị trí, đã học và sửa lỗi.

Trong công việc này, chúng tôi sử dụng các hàm sin và cosin có tần số khác nhau:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Trong đó pos là vị trí và i là kích thước. Nghĩa là, mỗi chiều của mã hóa vị trí tương ứng với một hình sin. Các bước sóng tạo thành một cấp số nhân từ 2π đến 10000 · 2π. Chúng tôi đã chọn chức năng này vì chúng tôi đưa ra giả thuyết rằng nó sẽ cho phép mô hình dễ dàng học cách tham dự các vị trí tương đối, vì đối với bất kỳ độ lệch k cố định nào, P Epos+k có thể được biểu diễn dưới dạng hàm tuyến tính của P Epos.

Thay vào đó , chúng tôi cũng đã thử nghiệm bằng cách sử dụng các phần nhúng vị trí đã học và nhận thấy rằng cả hai các phiên bản tạo ra kết quả gần như giống hệt nhau. Chúng tôi chọn phiên bản hình sin bởi vì nó có thể cho phép mô hình ngoại suy theo độ dài chuỗi dài hơn độ dài chuỗi gặp phải trong quá trình huấn luyện.

2.2.4. Layer normalization

Layer normalization là một phương pháp chuẩn hóa đơn giản để cải thiện tốc độ huấn luyện cho các mô hình mạng nơ-ron khác nhau. Không giống như chuẩn hóa hàng loạt, phương pháp được đề xuất ước tính trực tiếp số liệu thống kê chuẩn hóa từ các đầu vào được tổng hợp đến các nơ-ron trong một lớp ẩn để việc chuẩn hóa không đưa ra bất kỳ sự phụ thuộc mới nào giữa các trường hợp huấn luyện. Cho thấy rằng chuẩn hóa lớp hoạt động tốt cho RNN và cải thiện cả thời gian đào tạo cũng như hiệu suất tổng quát hóa của một số mô hình RNN hiện có.

Bây giờ chúng ta xem xét phương pháp chuẩn hóa lớp được thiết kế để khắc phục những hạn chế của chuẩn hóa hàng loạt.

Lưu ý rằng những thay đổi về đầu ra của một lớp sẽ có xu hướng gây ra những thay đổi tương quan cao về tổng đầu vào của lớp tiếp theo, đặc biệt là với các đơn vị ReLU có đầu ra có thể thay đổi rất nhiều. Điều này cho thấy vấn đề "dịch chuyển hiệp phương sai" có thể được giảm bớt bằng cách sửa giá trị trung bình và phương sai của tổng các đầu vào trong mỗi lớp. Do đó, chúng tôi tính toán số liệu thống kê chuẩn hóa lớp trên tất cả các đơn vị ẩn trong cùng một lớp như sau:

$$\mu^l = \frac{1}{H}\sum_{i=1}^{H} a_i^l \qquad \sigma^l = \sqrt{\frac{1}{H}\sum_{i=1}^{H}\left(a_i^l - \mu^l\right)^2}$$

Trong đó H biểu thị số lượng đơn vị ẩn trong một lớp. Sự khác biệt giữa phương trình. (2) và phương trình. (3) là trong quá trình chuẩn hóa lớp, tất cả các đơn vị ẩn trong một lớp có chung các thuật ngữ chuẩn hóa μ và σ, nhưng các trường hợp huấn luyện khác

nhau có các thuật ngữ chuẩn hóa khác nhau. Không giống như chuẩn hóa hàng loạt, chuẩn hóa lớp không áp đặt bất kỳ ràng buộc nào đối với kích thước của lô nhỏ và nó có thể được sử dụng trong chế độ trực tuyến thuần túy với kích thước lô 1.

2.2.5. Self-attention layer

Self-Attention (hay còn gọi là Intra-Attention) là một cơ chế chú ý mà trong đó mỗi phần tử của chuỗi đầu vào tương tác với tất cả các phần tử khác trong chuỗi để tạo ra một biểu diễn mới của chuỗi. Mục đích để tìm hiểu và nắm bắt các mối quan hệ ngữ nghĩa giữa các phần tử trong cùng một chuỗi, giúp mô hình hiểu rõ ngữ cảnh toàn cầu.

Vậy tại sao phải sử dụng Self-attention layer?

Trong phần này, chúng tôi so sánh các khía cạnh khác nhau của các lớp tự chú ý với các lớp lặp lại và lớp chập thường được sử dụng để ánh xạ một chuỗi biểu diễn ký hiệu có độ dài thay đổi (x1, ..., xn) sang một dãy khác có độ dài bằng nhau (z1, ..., zn), với xi ,zi ,Rd, chẳng hạn như một ẩn lớp trong bộ mã hóa hoặc bộ giải mã truyền dẫn trình tự điển hình. Thúc đẩy việc sử dụng sự chú ý của chúng ta hãy xem xét ba mong muốn.

- Một là tổng độ phức tạp tính toán trên mỗi lớp.
- Hai là số lượng tính toán có thể được song song hóa, được đo bằng số lượng hoạt động tuần tự tối thiểu được yêu cầu.
- Thứ ba là độ dài đường dẫn giữa các phần phụ thuộc tầm xa trong mạng.

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Như đã lưu ý trong Bảng 1, lớp tự chú ý kết nối tất cả các vị trí với số lượng liên tục không đổi các hoạt động được thực hiện, trong khi lớp lặp lại yêu cầu các hoạt động tuần tự O(n) . Về mặt độ phức tạp tính toán, các lớp tự chú ý sẽ nhanh hơn các lớp lặp lại khi chuỗi chiều dài n nhỏ hơn chiều biểu diễn d, điều này thường xảy ra với cách trình bày câu được sử dụng bởi các mô hình tiên tiến nhất trong các bản dịch máy, chẳng hạn

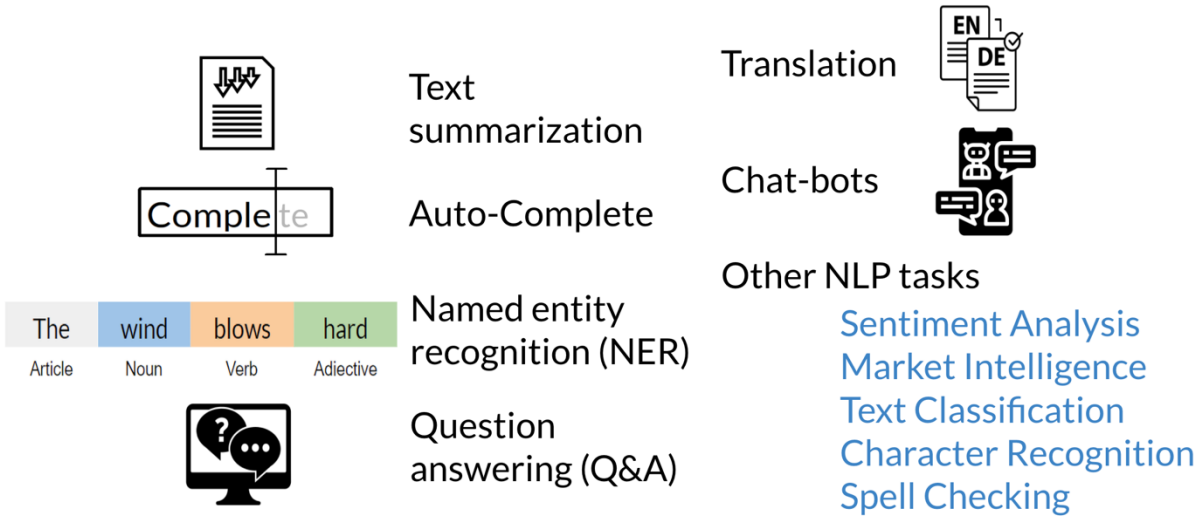
như từng từ và biểu diễn cặp byte. Để cải thiện hiệu suất tính toán cho các nhiệm vụ liên quan đến các chuỗi rất dài, sự tự chú ý có thể bị hạn chế khi chỉ xem xét một vùng lân cận có kích thước r trong trình tự đầu vào tập trung quanh vị trí đầu ra tương ứng. Điều này sẽ tăng độ dài đường dẫn tối đa lên O(n/r).

Một lớp chập đơn có độ rộng hạt nhân k < n không kết nối tất cả các cặp vị trí đầu vào và đầu ra. Làm như vậy đòi hỏi một chồng các lớp chập O(n/k) trong trường hợp các hạt nhân liền kề hoặc O(logk(n)) trong trường hợp các lớp chập bị giãn, tăng độ dài của các đường đi dài nhất giữa hai vị trí bất kỳ trong mạng. Các lớp tích chập thường đắt hơn các lớp hồi quy, theo hệ số k. Tuy nhiên, các tích chập có thể tách rời làm giảm độ phức tạp đáng kể, đến O(k · n · d + n · d^2) tích chập tương đương với sự kết hợp giữa lớp tự chú ý và lớp chuyển tiếp theo điểm, cách tiếp cận mà chúng tôi tham gia. Tuy nhiên, ngay cả với k = n, độ phức tạp của một hàm phân tách được vào mô hình của chúng tôi.

Về lợi ích phụ, Self-Attention có thể mang lại những mô hình dễ hiểu hơn. Chúng tôi kiểm tra sự phân bổ sự chú ý từ các mô hình của mình, đồng thời trình bày và thảo luận các ví dụ trong phần phụ lục. Self-Attention không chỉ học cách thực hiện các nhiệm vụ khác nhau một cách rõ ràng mà nhiều người còn thể hiện hành vi liên quan đến cấu trúc cú pháp và ngữ nghĩa của câu.

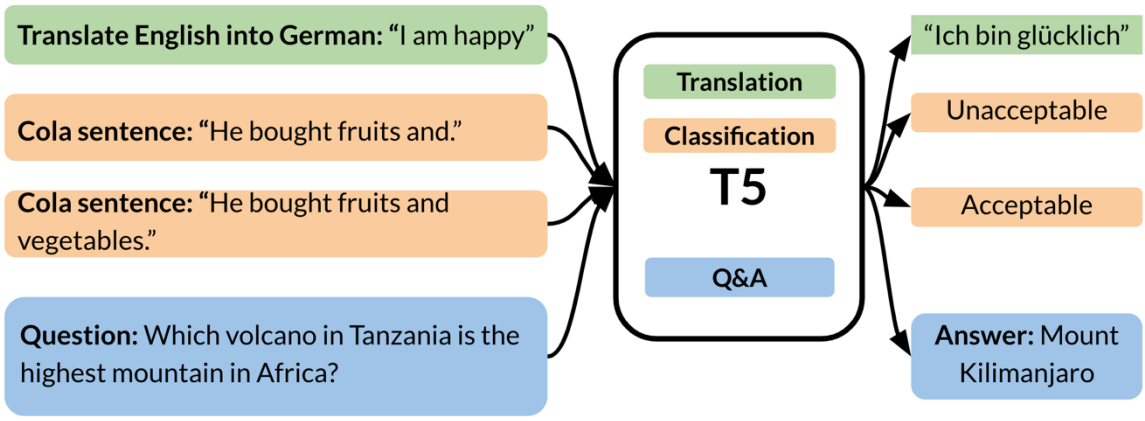
2.2.6. Ứng dụng của Transformer trong NLP

Dưới đây là bản tóm tắt ngắn gọn về tất cả các ứng dụng khác nhau mà bạn có thể xây dựng bằng Transformer:



Một lĩnh vực nghiên cứu thú vị khác là sử dụng phương pháp học chuyển giao với Transformer. Ví dụ: để đào tạo một mô hình sẽ dịch tiếng Anh sang tiếng Đức, bạn chỉ cần thêm văn bản "dịch tiếng Anh sang tiếng Đức" vào đầu vào mà bạn sắp cung cấp cho

mô hình. Sau đó, bạn có thể giữ lại mô hình đó để phát hiện cảm tính bằng cách thêm một thẻ khác vào trước. Hình ảnh sau đây tóm tắt mẫu T5 sử dụng khái niệm này:



3. Khó khăn trong quá trình thực hiện

- Theo như tiến độ công việc thì cơ bản nắm được Attention mechanism, Transformer (Positional embeddings, Layer normalization, Self-attention layer)

4. Đề xuất tiến độ công việc tiếp theo (Tuần 2)

- Tìm hiểu về Transfer Learning (Encoder-only - Model Bert và các biến thể Roberta, Electra)

5. Tài liệu tham khảo

[1] Ashish Vaswani, Llion Jones, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin (2017) Attention Is All You Need. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[2] Zhaoyang Niu, Guoqiang Zhong, Hui Yu (2021) A review on the attention mechanism of deep learning. https://www.sciencedirect.com/science/article/pii/S092523122100477X?casa_token=XiMWvQCA8bwAAAAA:4dQLLufR5o7qsnOuVK75zn8-mzXdiTNOQ_Ss6GB2kfEADYi3-LcfDjeJUK0K3H3vPO92zQi9oNk

[3] Sanghyuk Roy Choi and Minhyeok Lee (2023) Transformer Architecture and Attention Mechanisms in Genome Data Analysis: A Comprehensive Review. https://www.mdpi.com/2079-7737/12/7/1033

[4] SUSHANT SINGH , (Member, IEEE), AND AUSIF MAHMOOD , (Member, IEEE) (2021) The NLP Cookbook: Modern Recipes for Transformer Based Deep Learning Architectures. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9422763

[5] Jimmy Lei Ba, Jamie Ryan Kiros, Geoffrey E. Hinton (2016) Layer Normalization. https://arxiv.org/pdf/1607.06450

[6] Pu-Chin Chen∗, Henry Tsai∗, Srinadh Bhojanapalli∗, Hyung Won Chung, Yin-Wen Chang, Chun-Sung Ferng (2021). A Simple and Effective Positional Encoding for Transformers. https://arxiv.org/pdf/2104.08698