



Nhận ngày 21 tháng 4 năm 2021, chấp nhận ngày 30 tháng 4 năm 2021, ngày xuất bản ngày 4 tháng 5 năm 2021, ngày của phiên bản hiện tại là ngày 14 tháng 5 năm 2021.

Mã định danh đối tượng kỹ thuật số 10.1109/ACCESS.2021.3077350

Sách dạy nấu ăn NLP: Bí quyết hiện đại cho Kiến trúc học sâu dựa trên máy biến áp

SUSHANT SINGH , (Thành viên, IEEE), VÀ AUSIF MAHMOOD , (Thành viên, IEEE)
 Khoa Khoa học và Kỹ thuật Máy tính, Đại học Bridgeport, Bridgeport, CT 06604, Hoa Kỳ
 Tác giả tương ứng: Sushant Singh (sushants@my.bridgeport.edu)

TÓM TẮT Trong những năm gần đây, các mô hình Xử lý ngôn ngữ tự nhiên (NLP) đã đạt được thành công vượt bậc trong các nhiệm vụ ngôn ngữ và ngữ nghĩa như phân loại văn bản, dịch máy, hệ thống đối thoại nhận thức, truy xuất thông tin thông qua Hiểu ngôn ngữ tự nhiên (NLU) và Tạo ngôn ngữ tự nhiên (NLG).

Thành tích này chủ yếu là do kiến trúc Transformer nổi tiếng, dẫn đến các thiết kế như BERT, GPT (I, II, III), v.v. Mặc dù các mô hình kích thước lớn này đã đạt được hiệu suất chưa từng có nhưng chúng lại có chi phí tính toán cao. Do đó, một số kiến trúc NLP gần đây đã sử dụng các khái niệm về học chuyển giao, cắt tỉa, lượng tử hóa và chắt lọc kiến thức để đạt được kích thước mô hình vừa phải trong khi vẫn giữ được hiệu suất gần như tương tự như những kiến trúc trước đó của chúng. Ngoài ra, để giảm thiểu thách thức về kích thước dữ liệu do các mô hình ngôn ngữ đặt ra từ góc độ trích xuất kiến thức, Công cụ truy xuất kiến thức đã được xây dựng để trích xuất các tài liệu dữ liệu rõ ràng từ một kho cơ sở dữ liệu lớn với hiệu quả và độ chính xác cao hơn. Nghiên cứu gần đây cũng tập trung vào khả năng suy luận vượt trội bằng cách chú ý hiệu quả đến các chuỗi đầu vào dài hơn. Trong bài viết này, chúng tôi tóm tắt và kiểm tra công nghệ tiên tiến nhất hiện nay (SOTA)

Các mô hình NLP đã được sử dụng cho nhiều nhiệm vụ NLP để có hiệu suất và hiệu quả tối ưu.


Chúng tôi cung cấp sự hiểu biết chi tiết và hoạt động của các kiến trúc khác nhau, phân loại thiết kế NLP, đánh giá so sánh và định hướng tương lai trong NLP.

CHỈ SỐ ĐIỀU KHOẢN Học sâu, xử lý ngôn ngữ tự nhiên (NLP), hiểu ngôn ngữ tự nhiên (NLU), tạo ngôn ngữ tự nhiên (NLG), truy xuất thông tin (IR), chắt lọc kiến thức (KD), cắt tỉa, lượng tử hóa.

I. GIỚI THIỆU

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

Phó biên tập viên điều phối việc xem xét bản thảo này và người phê duyệt nó để xuất bản là Ali Shariq Imran .

gây ra sự gia tăng nhanh chóng của các mô hình NLP dựa trên Deep Learning [1].

Kiến trúc Transformer [2] 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 [3] và BERT [4] 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 [5] - [10]. 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.

Thành tựu này thiếu mục tiêu chính của việc học chuyển giao là đạt được độ chính xác cao của mô hình với các mẫu tinh chỉnh tối thiểu. Ngoài ra, hiệu suất của mô hình cần phải được khái quát hóa trên một số bộ dữ liệu và không được

nhệm vụ hoặc tập dữ liệu cụ thể [11]–[13]. Tuy nhiên, mục tiêu cao việc học tổng quát hóa và chuyển giao đang bị tổn hại khi lượng dữ liệu ngày càng tăng được sử dụng cho cả hai mục đích đào tạo trước và tinh chỉnh. Điều này làm lu mờ quyết định liệu dữ liệu huấn luyện lớn hơn hay kiến trúc được cải tiến nên được kết hợp để xây dựng một ngôn ngữ SOTA tốt hơn người mẫu. Ví dụ, kiến trúc XLNet [14] tiếp theo sở hữu mô hình ngôn ngữ mới lạ nhưng phức tạp, cung cấp một cải tiến nhỏ so với BERT đơn giản kiến trúc được đào tạo chỉ trên 10% dữ liệu của XLNet (113GB). Sau đó, với sự kích thích của RoBERTa [15], một mô hình dựa trên BERT lớn được đào tạo trên nhiều dữ liệu hơn đáng kể hơn BERT (160GB), vượt trội hơn XLNet. Do đó, một kiến trúc có tính tổng quát hơn và được đào tạo sâu hơn về dữ liệu lớn hơn, dẫn đến điểm chuẩn NLP.

Các kiến trúc nêu trên chủ yếu là ngôn ngữ hiểu các mô hình, trong đó một phương ngữ tự nhiên được ánh xạ tới một sự giải thích hình thức. Ở đây mục tiêu ban đầu là dịch của cách phát âm của người dùng đầu vào thành cách diễn đạt cụm từ thông thường. Để hiểu ngôn ngữ tự nhiên (NLU), biểu diễn trung gian cho mục tiêu cuối cùng của các mô hình trên là được quyết định bởi các nhiệm vụ xuôi dòng.

Trong khi đó, việc tinh chỉnh đang trở thành thách thức ngày càng lớn đối với các vai trò có nhiệm vụ cụ thể trong các mô hình NLU. Vì nó yêu cầu cỡ mẫu lớn hơn để tìm hiểu một nhiệm vụ cụ thể, làm mất đi những mô hình như vậy khỏi sự khái quát hóa [16]. Cái này đã kích hoạt sự ra đời của Tạo ngôn ngữ tự nhiên (NLG) các mô hình trái ngược với đào tạo của NLU, tạo ra các cách nói phương ngữ học được từ các phương ngữ bị che giấu hoặc bị sửa lỗi tương ứng của chúng. ngữ nghĩa đầu vào. Các mô hình như vậy hoạt động khác với cách tiếp cận xuôi dòng thông thường về hiểu ngôn ngữ chữ thảo và tối ưu cho việc tạo chuỗi theo chuỗi nhiệm vụ, chẳng hạn như dịch ngôn ngữ. Các mẫu như T5 [17], BART [18], mBART [19], T-NLG [20] đã được đào tạo trước trên một kho văn bản lớn bị hỏng và tạo ra văn bản được làm sạch tương ứng thông qua mục tiêu khử nhiễu [21]. Cái này quá trình chuyển đổi rất hữu ích vì lớp tinh chỉnh bổ sung cho Nhiệm vụ của NLU không bắt buộc cho mục đích NLG. Điều này hơn nữa nâng cao khả năng dự đoán thông qua việc học bằng 0 hoặc vài lần cho phép tạo chuỗi với sự tinh chỉnh tối thiểu hoặc không cần tinh chỉnh. Ví dụ: nếu không gian nhúng ngữ nghĩa của mô hình được huấn luyện trước khả năng nhận dạng động vật 'mèo', 'sư tử' và 'tinh tinh', nó vẫn có thể dự đoán chính xác 'con chó' mà không cần tinh chỉnh. Mặc dù việc tạo ra trình tự vượt trội khả năng, kích thước mô hình NLG tăng theo cấp số nhân với bản phát hành tiếp theo của GPT-III [22], bản phát hành lớn nhất mô hình trước khi phát hành GShard [23].

Do các mô hình có kích thước đặc biệt lớn của NLU và NLG cần nhiều GPU để tải, điều này hóa ra tốn kém và nguồn lực bị hạn chế trong hầu hết các tình huống thực tế. Hơn nữa, khi được đào tạo trong vài ngày hoặc vài tuần trên cụm GPU, những mô hình khổng lồ này có chi phí năng lượng cắt cổ. Để giảm thiểu chi phí tính toán như vậy [24], các mô hình dựa trên Phân tích kiến trúc (KD) [25] như DistilBERT [26], Tiny-BERT [27], MobileBERT [28] đã được giới thiệu ở mức giá thấp hơn. chi phí suy luận và kích thước. Những mô hình sinh viên nhỏ hơn này

tận dụng khuynh hướng quy nạp của các mô hình giáo viên lớn hơn (BERT) để đạt được thời gian đào tạo nhanh hơn. Tương tự, kỹ thuật cắt tỉa và lượng tử hóa [29] trở nên phổ biến để xây dựng mô hình có quy mô kinh tế. Cắt tỉa có thể được phân loại thành 3 loại: cắt tỉa theo trọng lượng, cắt tỉa theo lớp và cắt tỉa phần đầu trong đó các trọng lượng, lớp và lớp đóng góp tối thiểu nhất định đầu chú ý được loại bỏ khỏi mô hình. Giống như việc cắt tỉa, lượng tử hóa nhận thức đào tạo được thực hiện để đạt được ít hơn Định dạng chính xác 32-bit do đó làm giảm kích thước mô hình.

Để có hiệu suất cao hơn, cần phải học tập nhiều hơn dẫn đến lưu trữ dữ liệu và kích thước mô hình lớn hơn. Quá hạn tới mức độ không rõ và khả năng lưu trữ kiến thức tiềm ẩn của mô hình, khả năng học tập của nó có những hạn chế về khả năng truy cập thông tin hiệu quả. Các mô hình Truy xuất Tri thức hiện tại như ORQA [30], REALM [31], RAG [32], DPR [33] cố gắng giảm bớt mối lo ngại về việc lưu trữ tiềm ẩn của các mô hình ngôn ngữ bằng cách cung cấp quyền truy cập bên ngoài vào kiến thức mô-đun có thể giải thích được. Điều này đạt được bằng cách bổ sung ngôn ngữ đào tạo trước mô hình với 'công cụ truy xuất kiến thức' giúp mô hình truy xuất và tham dự một cách hiệu quả các hoạt động rõ ràng nhằm mục tiêu các tài liệu từ một kho tài liệu lớn như Wikipedia.

Hơn nữa, mô hình Transformer không có khả năng xử lý đầu vào các trình tự vượt quá một khoảng mã thông báo cố định đã ngăn cản họ hiểu được các nội dung văn bản lớn một cách toàn diện. Điều này đặc biệt rõ ràng khi các từ liên quan cách xa nhau hơn so với các từ liên quan. chiều dài đầu vào. Vì vậy, để nâng cao sự hiểu biết theo ngữ cảnh, các kiến trúc như Transformer-XL [34], Longformer [35], ETC [36], Big Bird [37], được giới thiệu với các sửa đổi cơ chế chú ý để xử lý các chuỗi dài hơn.

Ngoài ra, do nhu cầu về các mô hình NLP tăng cao để có hiệu quả kinh tế và sẵn có trên các thiết bị biên, các mô hình nén tiên tiến đã được ra mắt dựa trên nền tảng chung.

Kỹ thuật. Đây là ngoài việc chứng cất, cắt tỉa, và Kỹ thuật lượng tử hóa được mô tả trước đó. Những mô hình như vậy triển khai một loạt các thủ tục tối ưu hóa máy tính từ băm [38], chú ý thừa thớt [39], tham số hóa nhúng theo yếu tố [40], phát hiện mã thông báo được thay thế [41], chia sẻ tham số giữa các lớp [42] hoặc kết hợp trong số những điều đã đề cập ở trên.

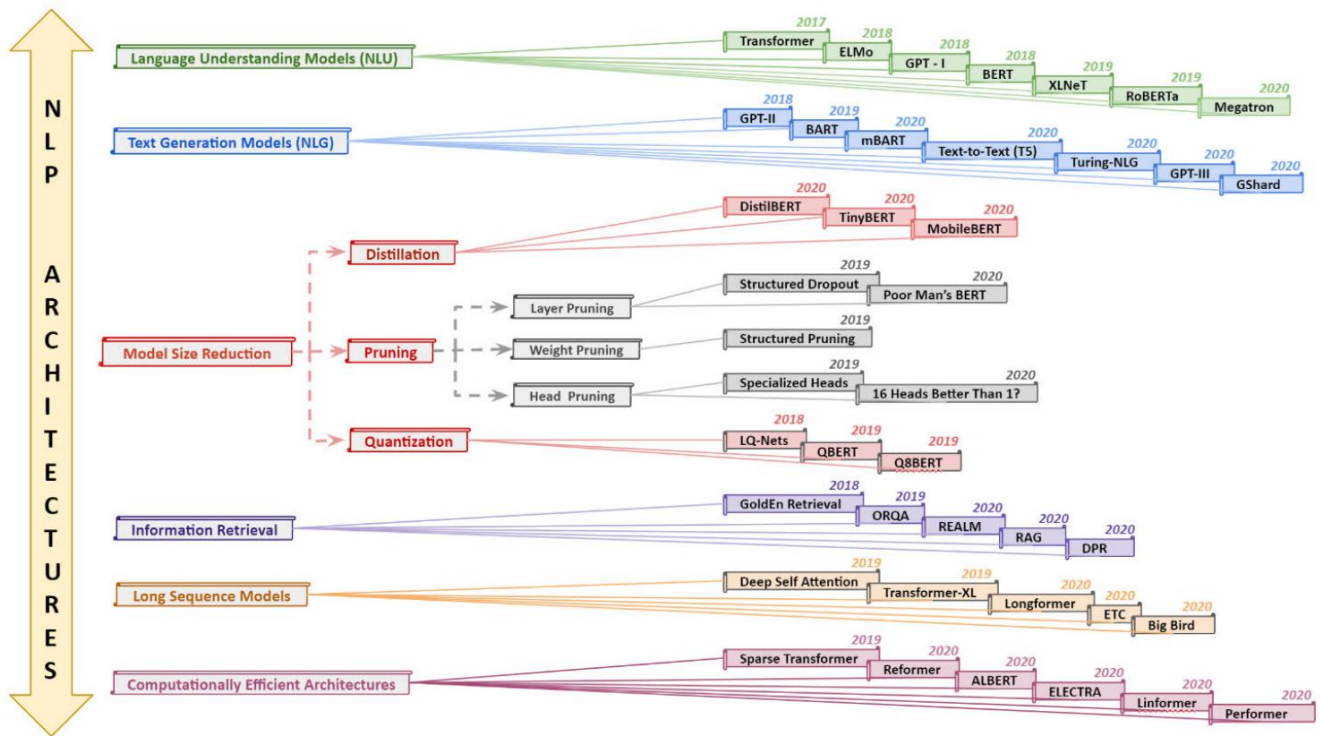
II. ĐÁNH GIÁ/PHÂN TÍCH LIÊN QUAN

Chúng tôi đề xuất một phân loại dựa trên NLP mới cung cấp một cách phân loại độc đáo phân loại các mô hình NLP hiện tại từ sáu khía cạnh khác nhau:

Mô hình NLU: Mô hình NLU vượt trội trong các nhiệm vụ phân loại, dự đoán có cấu trúc và/hoặc tạo truy vấn. Đây là được thực hiện thông qua đào tạo trước và tinh chỉnh được thúc đẩy bởi nhiệm vụ tiếp theo.

Các mô hình NLG: Ngược lại với các mô hình NLU, đây là những mô hình nổi bật trong các nhiệm vụ tạo tuần tự. Họ tạo ra văn bản rõ ràng thông qua việc học một vài lần và một lần từ những câu nói bị hỏng tương ứng.

Giảm kích thước mô hình: Sử dụng các kỹ thuật dựa trên nén như KD, Pruning và Quantization để tạo mô hình lớn kinh tế và thực dụng. Nó hữu ích cho



HÌNH 1. Phân loại kiến trúc NLP.

triển khai theo thời gian thực các mô hình ngôn ngữ lớn để hoạt động trên các thiết bị biên. Truy xuất thông tin (IR): Việc trả lời câu hỏi trong miền mở theo ngữ cảnh (QA) phụ thuộc vào việc truy xuất tài liệu hiệu quả và hiệu quả. Do đó, các hệ thống IR thông qua việc trích xuất từ vựng và ngữ nghĩa vượt trội của các tài liệu vật lý từ một kho văn bản lớn tạo ra SOTA trong miền QA trên nhiều điểm chuẩn vượt trội so với các mô hình ngôn ngữ đương thời. Mô hình chuỗi dài: Độ phức tạp tính toán dựa trên sự chú ý trong Transformers có tỷ

lệ bậc hai với độ dài đầu vào, do đó nó thường được cố định ở mức 512 mã thông báo. Điều này có thể được chấp nhận đối với các tác vụ giải quyết đồng tham chiếu được hưởng lợi từ độ dài đầu vào nhỏ hơn [43], tuy nhiên, không đủ cho các tác vụ Trả lời câu hỏi (QA) trong đó yêu cầu lý luận trên nhiều tài liệu dài, ví dụ: bộ dữ liệu HotpotQA [44].

Kiến trúc hiệu quả về mặt tính toán: Kiến trúc hiệu quả về bộ nhớ với độ chính xác tương đương với các mô hình ngôn ngữ lớn được xây dựng để giảm thời gian đào tạo cao của các mô hình đó.

Phần được đề cập ở trên là sự phân loại tổng quát chứ không phải là sự phân loại cứng, một số mô hình có thể được sử dụng thay thế cho nhau có thể phục vụ các mục đích kép, tuy nhiên, có sự phân định rõ ràng mặc dù tính phổ biến không đáng kể.

Hình 1 mô tả cách phân loại này cung cấp thông tin phân tích trực quan về các mẫu xe quan trọng thuộc các danh mục khác nhau cùng với năm ra mắt của chúng.

III. GIỚI THIỆU CHO KIẾN TRÚC NLP HIỆN ĐẠI

Mô hình Bộ mã hóa-Giải mã RNN truyền thống [45] bao gồm hai mạng thần kinh tái phát (RNN), trong đó một mạng tạo ra phiên bản được mã hóa của chuỗi đầu vào và mạng kia tạo phiên bản được giải mã của nó thành một chuỗi khác.

Để tối đa hóa xác suất có điều kiện của mục tiêu cho chuỗi đầu vào, mô hình được đào tạo cùng với mô hình ngôn ngữ sau,

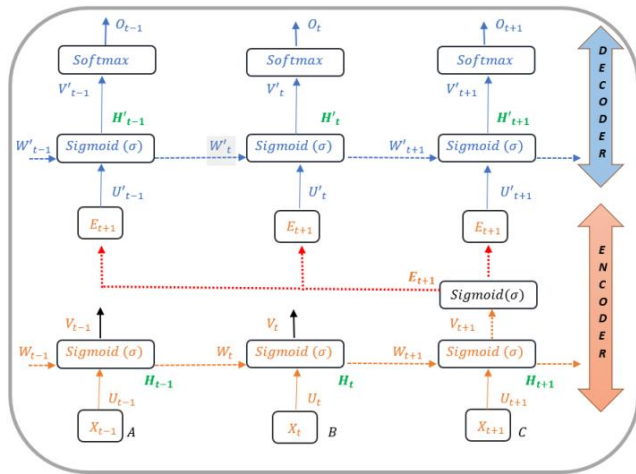
$$y = \operatorname{argmax}_P (y_t | y_1, y_2, y_3, \dots, y_{t-1}) \quad (1)$$

$$P(y_t | y_1, y_2, y_3, \dots, y_{t-1}) = P(y_t | y_1^{t-1}) \quad (2)$$

Một hệ thống như vậy được chứng minh bằng thực nghiệm là mang lại kết quả vượt trội so với RNN thông thường, LSTM [46] hoặc GRU [47] bằng cách thực hiện các xác suất có điều kiện của các cặp pha trong dịch máy, ánh xạ trình tự sang trình tự hoặc các tác vụ tóm tắt văn bản.

Trong kiến trúc trên (Hình 2), lớp cuối cùng E_{t+1} của Bộ mã hóa truyền thông tin đến bộ giải mã từ lớp V_{t+1} ở đầu cuối của nó, lớp chứa toàn bộ hiểu biết theo ngữ cảnh của tất cả các từ trước đó thông qua phân bố xác suất.

Sự biểu diễn trừu tượng kết hợp của tất cả các từ này được đưa đến bộ giải mã để tính toán tác vụ dựa trên ngôn ngữ mong muốn. Giống như các lớp trước của nó, các tham số có thể học tương ứng của lớp cuối cùng lần lượt là U_{t+1} và V_{t+1} ở đầu vào và đầu ra tại Bộ mã hóa và UV $t+1$ ở Bộ giải mã. $t+1$, Việc kết hợp các ma trận trọng số với trạng thái ẩn và độ lệch có thể



HÌNH 2. Kiến trúc bộ mã hóa-giải mã.

được biểu diễn dưới dạng toán học như sau:

Mã hoá:

$$H_{t+1} = \sigma(U_{t+1}.X_{t+1} + W_{t+1}.H_t + b_t) \quad (3)$$

$$E_{t+1} = \sigma(V_{t+1}.H_{t+1} + b_t) \quad H_{t+1} = \quad (4)$$

Bộ giải mã:

$$H_{t+1} = \sigma(U_{t+1}.E_{t+1} + W_{t+1}.H_t + b_{t+1}) \quad (5)$$

$$= \text{Softmax}(H_{t+1}.V_{t+1} + b_{t+1}) \quad (6)$$

Sau đó, việc tạo ra Chú ý [48], [49] trong năm 2014-15 đã vượt qua giới hạn của Bộ mã hóa-Giải mã RNN vốn chịu sự phụ thuộc đầu vào trước đó, khiến việc suy ra các chuỗi dài hơn trở nên khó khăn và bị biến dạng và bùng nổ độ dốc [50]. Cơ chế chú ý đã loại bỏ sự phụ thuộc RNN bằng cách vô hiệu hóa toàn bộ bối cảnh đầu vào thông qua một nút Bộ mã hóa cuối cùng. Nó cân tất cả các đầu vào riêng lẻ cung cấp cho bộ giải mã để tạo ra chuỗi mục tiêu. Điều này mang lại sự hiểu biết theo ngữ cảnh tốt hơn dẫn đến những dự đoán vượt trội trong việc tạo chuỗi mục tiêu.

Đầu tiên, căn chỉnh xác định mức độ khớp giữa đầu ra j có thể được xác định từ đầu vào và tối

$$et_j = \tanh(h_{1,j}, h_j) \quad (7)$$

Chính xác hơn, điểm căn chỉnh lấy tất cả các trạng thái đầu ra của bộ mã hóa và trạng thái ẩn được giải mã trước đó làm đầu vào, được biểu thị bằng:

$$\text{Score}_{eqnarray} = W_{comb}. \text{tính}(W_{dec}.H_{dec} + W_{enc}.H_{enc}) \quad (8)$$

Trạng thái ẩn của bộ giải mã và đầu ra bộ mã hóa được truyền qua các lớp tuyến tính tương ứng cùng với các trọng số có thể huấn luyện của chúng. Trọng số at_j cho mỗi biểu diễn ẩn được mã hóa h_j được tính như sau:

$$at_j = \frac{\exp(et_j)}{\sum_{k=1}^{Tx} \exp(et_k)} \quad (9)$$

Vectơ bối cảnh kết quả trong cơ chế chú ý này được xác định bởi:

T_x

$$ct = \sum_{j=1}^{at_j h_j} \text{ trong đó } T_x = \text{độ dài chuỗi đầu vào (10) } j=1$$

Cơ chế Chú ý về cơ bản là tạo ra vectơ ngữ cảnh được tính toán từ các điểm căn chỉnh khác nhau ở các vị trí khác nhau như trong Hình 3.

Cơ chế Chú ý của Lượng khác với cơ chế của Bahdanau nêu trên về cách tính điểm căn chỉnh. Nó sử dụng cả sự chú ý toàn cục và cục bộ, trong đó sự chú ý toàn cục sử dụng tất cả các trạng thái đầu ra của bộ mã hóa trong khi sự chú ý cục bộ tập trung vào một tập hợp con từ nhỏ. Điều này giúp đạt được bản dịch tốt hơn cho các chuỗi dài hơn. Những thiết kế chú ý này đã dẫn đến sự phát triển của kiến trúc Transformer hiện đại sử dụng cơ chế chú ý năng cao như được mô tả trong phần tiếp theo.

IV. NLU ARCHITECTURES Cách tiếp

cận của NLU trong việc chuyển các biểu diễn ngôn ngữ thần kinh được đào tạo trước đã chứng minh rằng các phần nhúng được đào tạo trước sẽ cải thiện kết quả nhiệm vụ xuôi dòng khi so sánh với các phần nhúng được học từ đầu [51], [52]. Nghiên cứu tiếp theo đã nâng cao khả năng học tập để nắm bắt những phản hồi từ được ngữ cảnh hóa và chuyển chúng sang các mô hình thần kinh [53], [54].

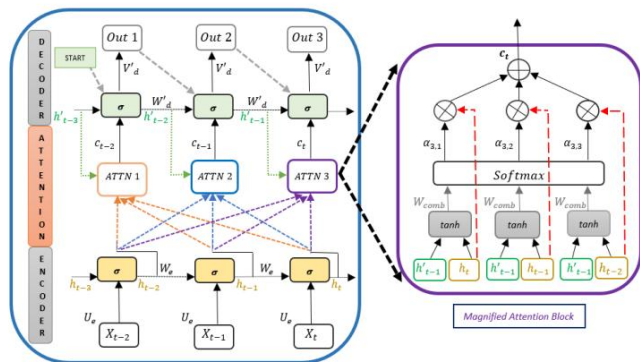
Những nỗ lực gần đây không chỉ giới hạn ở [55]–[57] đã tiếp tục xây dựng dựa trên những ý tưởng này bằng cách bổ sung việc tinh chỉnh từ đầu đến cuối các mô hình ngôn ngữ cho các tác vụ tiếp theo bên cạnh việc trích xuất các biểu diễn từ theo ngữ cảnh. Tiến bộ kỹ thuật này, cùng với khả năng tính toán lớn đã phát triển phương pháp hiện đại của NLU từ chuyển các từ nhúng sang chuyển toàn bộ mô hình ngôn ngữ nhiều tỷ tham số, đạt được kết quả chưa từng có trong các nhiệm vụ NLP. Các mô hình NLU hiện đại tận dụng Transformers cho các nhiệm vụ lập mô hình và chỉ sử dụng Bộ mã hóa hoặc phương pháp tiếp cận dựa trên Bộ giải mã theo yêu cầu. Những mô hình như vậy sẽ được giải thích một cách sinh động trong phần tiếp theo.

A. MÁY BIẾN ÁP

1) KIẾN TRÚC Máy biến áp

ban đầu là mô hình Mã hóa-Giải mã 6 lớp, tạo ra chuỗi mục tiêu thông qua Bộ giải mã từ chuỗi nguồn thông qua Bộ mã hóa. Bộ mã hóa và giải mã ở mức cao bao gồm lớp tự chú ý và lớp chuyển tiếp nguồn cấp dữ liệu. Trong Bộ giải mã, một lớp chú ý bổ sung ở giữa cho phép nó ánh xạ các mã thông báo có liên quan tới Bộ mã hóa cho mục đích dịch thuật. Tính năng Tự chú ý cho phép tra cứu các từ đầu vào còn lại ở nhiều vị trí khác nhau để xác định mức độ liên quan của từ hiện đang được xử lý. Điều này được thực hiện cho tất cả các từ đầu vào giúp đạt được khả năng mã hóa vượt trội và hiểu biết theo ngữ cảnh của tất cả các từ.

Kiến trúc máy biến áp được xây dựng để tạo ra tính song song trong dữ liệu tuần tự của RNN và LSTM, trong đó mã thông báo đầu vào được cung cấp ngay lập tức và các phần nhúng tương ứng được tạo đồng thời thông qua Bộ mã hóa. Việc nhúng này



HÌNH 3. Cơ chế chú ý trên mô hình mã hóa-giải mã.

ánh xạ một từ (mã thông báo) vào một vectơ có thể được huấn luyện trước một cách nhanh chóng hoặc để tiết kiệm thời gian, một không gian nhúng được huấn luyện trước như GloVe sẽ được triển khai. Tuy nhiên, các mã thông báo tương tự ở các chuỗi khác nhau có thể có cách hiểu khác nhau được giải quyết thông qua bộ mã hóa vị trí tạo ra thông tin từ dựa trên ngữ cảnh liên quan đến vị trí của nó. Sau đó, biểu diễn theo ngữ cảnh nâng cao được đưa đến lớp chú ý để tăng cường ngữ cảnh hóa bằng cách tạo ra các vectơ chú ý, xác định mức độ liên quan của từ i trong một chuỗi liên quan đến các từ khác. Sau đó, các vectơ chú ý này được đưa đến Mạng thần kinh chuyển tiếp nguồn cấp dữ liệu, nơi chúng được chuyển đổi thành dạng dễ tiêu hóa hơn cho khối 'Bộ mã hóa' hoặc 'Chú ý bộ giải mã-bộ giải mã' tiếp theo của Bộ giải mã.

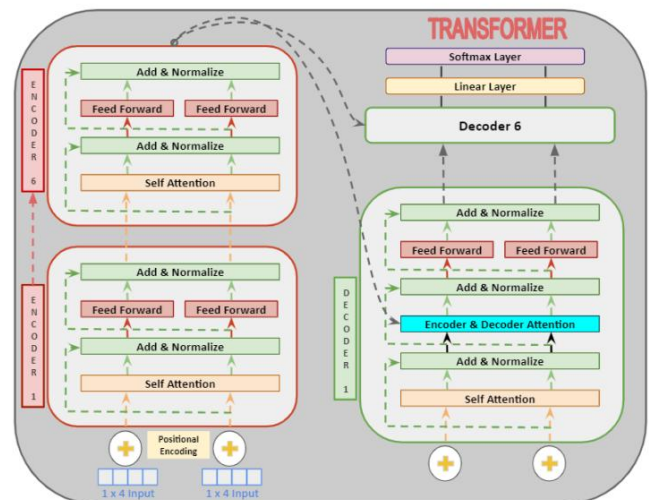
Cái sau được cung cấp với đầu ra Bộ mã hóa và nhúng đầu vào Bộ giải mã để thực hiện sự chú ý giữa hai đầu vào. Điều này xác định mức độ liên quan của mã thông báo đầu vào của Transformer liên quan đến mã thông báo mục tiêu của nó khi bộ giải mã thiết lập biểu diễn vectơ thực tế giữa ánh xạ nguồn và đích. Bộ giải mã dự đoán từ tiếp theo thông qua softmax được thực thi qua nhiều bước thời gian cho đến khi tạo ra mã thông báo cuối câu. Tại mỗi lớp Transformer, có các kết nối dư theo sau là bước chuẩn hóa lớp [58] để tăng tốc độ huấn luyện trong quá trình lan truyền ngược. Tất cả các chi tiết kiến trúc máy biến áp được thể hiện trong Hình 4.

2) TRUY VẤN, KHÓA VÀ GIÁ TRỊ ĐẦU

vào của cơ chế Chú ý của Máy biến áp là mã thông báo đích Vectơ truy vấn Q, mã thông báo nguồn tương ứng của nó Vectơ khóa K và Giá trị V là ma trận nhúng. Việc ánh xạ các mã thông báo nguồn và đích trong dịch máy có thể được định lượng theo mức độ giống nhau của từng mã thông báo trong một chuỗi thông qua sản phẩm dấu chấm bên trong. Do đó, để đạt được bản dịch chính xác, khóa phải khớp với truy vấn tương ứng của nó, thông qua giá trị tích số chấm cao giữa hai giá trị. Giả sử Q { ' LQ, D } và K { ' LK, D } trong đó LQ, LK biểu thị độ dài đích và nguồn, trong khi D biểu thị tính chất nhúng của từ.

Softmax được triển khai để đạt được phân phối xác suất trong đó tất cả các điểm tương đồng của Truy vấn, Khóa cộng lại thành một và khiến sự chú ý tập trung hơn vào các khóa phù hợp nhất.

$$WSM = \text{softmax}(QK^T) \text{ trong đó } WSM \{ ' LQ, LK \} \quad (11)$$



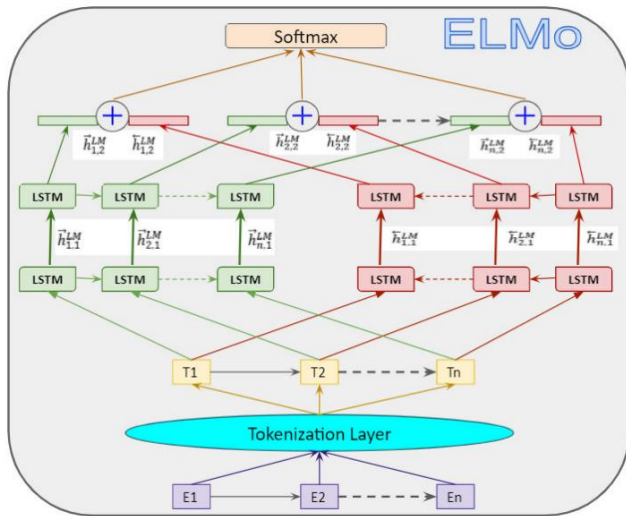
HÌNH 4. Cấu trúc máy biến áp nhiều đầu.

Truy vấn gắn xác suất cho khóa để khớp và do đó, các giá trị thường tương tự với khóa

$$Z_{Att} = \text{Chú ý } (Q, K, V) = \text{softmax } QK^T \cdot V = WSM \cdot V \quad (12)$$

3) CHÚ Ý ĐA ĐẦU (MHA) VÀ MẶT NẠ MHA nâng cao khả năng của mô hình để nhấn mạnh các vị trí mã thông báo khác nhau của chuỗi bằng cách triển khai chú ý song song nhiều lần. Các đầu ra hoặc đầu chú ý riêng lẻ thu được được nối và chuyển đổi thông qua một lớp tuyến tính thành các kích thước dự kiến. Mỗi đầu trong số nhiều đầu cho phép tham dự các phần trình tự từ một góc nhìn khác nhau, cung cấp các dạng biểu diễn tương tự cho mỗi mã thông báo. Điều này được thực hiện vì vectơ tự chú ý của mỗi mã thông báo có thể có trọng số từ mà nó đại diện cao hơn các từ khác do tích số chấm có kết quả cao. Điều này không hiệu quả vì mục tiêu là đạt được sự tương tác được đánh giá tương tự với tất cả các mã thông báo. Do đó, khả năng tự chú ý được tính toán 8 lần khác nhau, tạo ra 8 vectơ chú ý riêng biệt cho mỗi mã thông báo được sử dụng để tính toán vectơ chú ý cuối cùng thông qua tổng có trọng số của tất cả 8 vectơ cho mỗi mã thông báo. Các vectơ chú ý nhiều đầu thu được được tính toán song song và được đưa đến lớp chuyển tiếp nguồn cấp dữ liệu.

Mỗi mã thông báo mục tiêu tiếp theo T_{t+1} được tạo bằng cách sử dụng nhiều mã thông báo nguồn trong bộ mã hóa (S_0, \dots, S_{t+n}). Tuy nhiên, trong bộ giải mã tự hồi quy, chỉ các mã thông báo tar-get theo bước thời gian trước đó mới được xem xét (T_0, \dots, T_t), nhằm mục đích dự đoán trước mục tiêu trong tương lai được gọi là che dấu nhân quả. Điều này được cung cấp để cho phép tìm hiểu tối đa các mã thông báo mục tiêu được dịch sau đó. Do đó, trong quá trình song song hóa thông qua các hoạt động ma trận, đảm bảo rằng các từ mục tiêu tiếp theo được che dấu bằng 0, do đó mạng chú ý không thể nhìn thấy trong tương lai. Transformer được mô tả ở trên đã mang lại sự cải thiện đáng kể trong miền NLP. Điều này dẫn đến rất nhiều kiến trúc hiệu suất cao mà chúng tôi sẽ mô tả trong các phần tiếp theo.



HÌNH 5. Mô hình ngôn ngữ ELMo dựa trên LSTM hai chiều.

B. NHỮNG TỪ MÔ HÌNH NGÔN NGỮ: ELMo Mục tiêu của ELMo [59] là tạo ra

cách biểu diễn từ ngữ cảnh sâu sắc có thể mô hình hóa (i) các đặc điểm cú pháp và ngữ nghĩa phức tạp của từ (ii) đa nghĩa hoặc mơ hồ từ vựng, các từ có cách phát âm tương tự có thể có ý nghĩa khác nhau ở những bối cảnh hoặc địa điểm khác nhau. Những cải tiến này đã tạo ra khả năng những từ phong phú theo ngữ cảnh mà các mô hình SOTA trước đây như GloVe không có. Không giống như các phiên bản tiền nhiệm sử dụng phương pháp nhưng được xác định trước, ELMo xem xét tất cả N lần xuất hiện mã thông báo (t_1, t_2, \dots, t_N) cho mỗi mã thông báo t trong toàn bộ chuỗi trước khi tạo các phần nhúng. Các tác giả đưa ra giả thuyết rằng mô hình có thể trích xuất các thuộc tính ngôn ngữ trừu tượng ở các lớp trên cùng của kiến trúc thông qua LSTM hai chiều dành riêng cho nhiệm vụ.

Điều này có thể thực hiện được bằng cách kết hợp mô hình ngôn ngữ tiến và lùi. Tại đầu thời gian $k = 1$, mô hình ngôn ngữ chuyển tiếp dự đoán mã thông báo tiếp theo t_k dựa trên các mã thông báo được quan sát trước đó của chuỗi đầu vào thông qua phân phối xác suất chung được hiển thị trong (13). Tương tự như vậy, trong (14) với thứ tự bị đảo ngược, mô hình ngôn ngữ lạc hậu dự báo các mã thông báo trước đó cho các mã thông báo trong tương lai.

$$p(t_1, t_2, \dots, t_n) = \prod_{k=1}^N p(t_k | t_1, t_2, \dots, t_{k-1}) \quad (13)$$

$$p(t_1, t_2, \dots, t_n) = \prod_{k=1}^N p(t_k | t_{k+1}, t_{k+2}, \dots, t_N) \quad (14)$$

Điều này được thực hiện thêm thông qua softmax trên lớp LSTM cuối cùng như trong Hình 5.

ELMo cho mỗi biểu diễn mã thông báo x_k tính toán biểu diễn vector hai chiều trung gian $h_{k,j}$ tại mỗi lớp j của mô hình LSTM như sau:

$$R_k = x_k, \quad h_{k,j}^{LM} = \sum_{j=1}^L h_{k,j}^{LM} \quad (15)$$

Về mặt toán học $h_{k,j}^{LM} = x_k$ sẽ là đại diện mã thông báo cấp thấp nhất và nó có thể được khái quát là:

$$h_{k,j}^{LM} = h_{k,j}^{LM} \quad j \in \{1, \dots, L\} \quad (16)$$

ELMo học các trọng số chuẩn hóa thông qua các biểu diễn lớp j của nhiệm vụ trên L softmax. Điều này dẫn đến một siêu nhiệm vụ cụ thể số y cho phép tối ưu hóa quy mô của nhiệm vụ. tham Do đó, đối với một nhiệm vụ cụ thể, phương sai biểu diễn từ trong các lớp khác nhau được biểu thị bằng:

$$ELM_{task_k} = E R_k; \quad \theta_{task_k} = \theta_{task_k} \quad j=0 \quad (17)$$

C. MÔ HÌNH TRƯỚC ĐÀO TẠO SÁNG TẠO: GPT-I

Trong giai đoạn đầu tiên thông qua học tập không giám sát, GPT-I dựa trên bộ giải mã được đào tạo trước trên một tập dữ liệu lớn. Điều này thúc đẩy tính toán dữ liệu thô giúp loại bỏ nút thắt ghi nhãn dữ liệu của quá trình học có giám sát. Giai đoạn thứ hai thực hiện tinh chỉnh theo nhiệm vụ cụ thể trên các tập dữ liệu được giám sát nhỏ hơn đáng kể với các biến thể đầu vào cận biên. Do đó, nó dẫn đến thuyết bất khả tri về nhiệm vụ lớn hơn các mô hình SOTA như ELMo, ULMFiT [56] và đã thành công trong các nhiệm vụ phức tạp hơn như lý luận thông thường, sự tương đồng về ngữ nghĩa và khả năng đọc hiểu. Việc đào tạo trước GPT-I có thể được mô hình hóa như một hàm tối đa hóa các mã thông báo không được giám sát {ui, ..., un}.

$$L(U) = \log P(u_i | u_{1:k}, \dots, u_{i-1};) \quad (18)$$

trong đó k là kích thước của sổ ngữ cảnh và xác suất có điều kiện được tham số hóa thông qua W . Với các lớp chú ý nhiều đầu và chuyển tiếp, việc phân phối xác suất dựa trên mã thông báo mục tiêu thông qua softmax được tạo ra.

$$h_n = \text{transformer_block}(h_{n-1}) \quad i \in [1, n] \quad h_0 = \text{UWE} + Wp \quad (19)$$

$$P(u) = \text{softmax}(h_n W) \quad e \quad (20)$$

$$P(u) = \text{softmax}(h_n W) \quad e \quad (21)$$

trong đó $U = u_1, \dots, u_N$ là tập hợp vector mã thông báo ngữ cảnh, n là số lớp, W và Wp lần lượt là các ma trận nhúng mã thông báo và vị trí. Sau quá trình đào tạo trước, quá trình điều chỉnh tham số cho tác vụ cuối được giám sát diễn ra. Ở đây, chuỗi đầu vào (x) từ tập dữ liệu có nhãn C được cung cấp cho mô hình được huấn luyện trước để thu được m kích hoạt cuối cùng của khối máy biến áp h được cung cấp cho l

$$P(y | x^1, \dots, x^m) = \text{softmax}(h^l W_y) \quad (22)$$

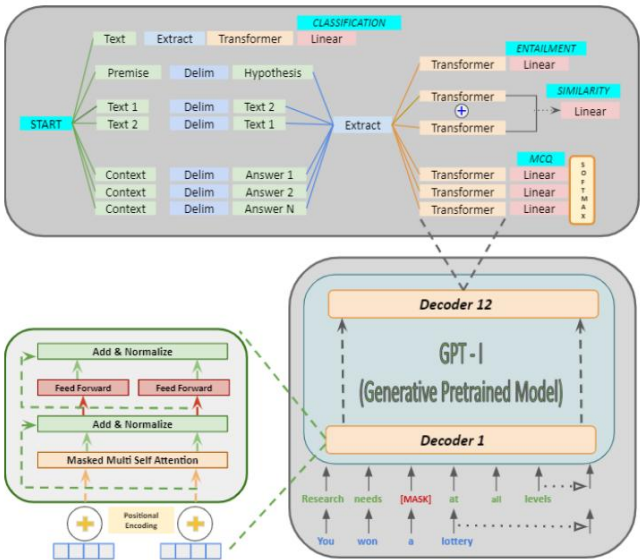
lớp đầu ra tuyến tính được tham số hóa (W_y) để dự đoán (y). Ngoài ra, mục tiêu L2 (C) được tối đa hóa như sau

$$P(y | x^1, \dots, x^m) = \text{softmax}(h^l W_y) \quad (22)$$

$$L_2(C) = \sum_{(x,y)} \log P(y | x^1, \dots, x^m) \quad (23)$$

Việc kết hợp mục tiêu mô hình hóa ngôn ngữ thứ cấp trong quá trình tinh chỉnh sẽ nâng cao việc học bằng cách khái quát hóa tốt hơn mô hình được giám sát và tăng tốc độ hội tụ như:

$$L_3(C) = L_2(C) + \lambda L_1(C) \quad (24)$$



HÌNH 6. Kiến trúc dựa trên tác vụ GPT-1 (trên cùng) và chế độ xem phóng to của bộ giải mã dựa trên máy biến áp (phía dưới).

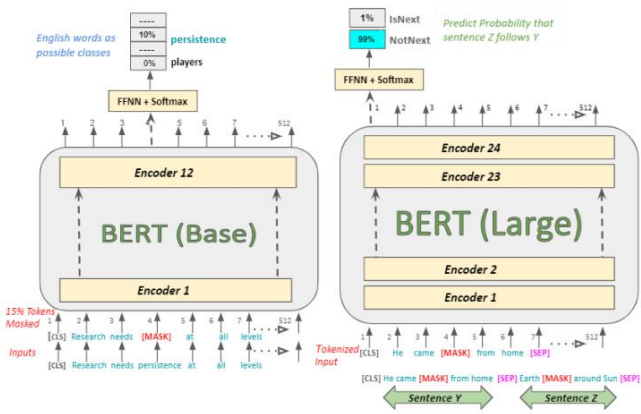
GPT thực hiện các nhiệm vụ khác nhau như phân loại, kế thừa, chỉ mục tương tự, Câu hỏi trắc nghiệm (MCQ) như trong hình 6. Giai đoạn trích xuất chất lọc các đặc điểm từ nội dung văn bản trước đó văn bản được phân tách thông qua mã thông báo 'Dấu phân cách' trong quá trình chuẩn bị văn bản. -xử lý. Mã thông báo này không cần thiết cho các nhiệm vụ phân loại vì nó không cần đánh giá mối quan hệ giữa nhiều chuỗi. Hơn nữa, các nhiệm vụ hỏi đáp hoặc yêu cầu văn bản liên quan đến các đầu vào được xác định như các cặp câu hoặc bộ ba câu có thứ tự trong một tài liệu. Đối với các nhiệm vụ MCQ, cần phải thay đổi ngữ cảnh ở đầu vào để đạt được kết quả chính xác. Điều này được thực hiện thông qua mục tiêu đào tạo Bộ giải mã dựa trên Máy biến áp trong đó các phép biến đổi đầu vào được tinh chỉnh cho các câu trả lời tương ứng.

D. ĐẠI DIỆN BỘ MÃ HÓA HAI HƯỚNG TỪ MÁY BIẾN ÁP: BERT

BERT là một tập hợp các Bộ mã hóa biến áp được đào tạo trước để khắc phục khả năng biểu đạt hạn chế của các mô hình trước đó, tức là việc thiếu ngữ cảnh hai chiều của GPT và sự ghép nối ngữ cảnh kép nông của ELMo. Mô hình sâu hơn của BERT cung cấp mã thông báo với nhiều bối cảnh với nhiều lớp và mô hình hai chiều cung cấp môi trường học tập phong phú hơn.

Tuy nhiên, tính hai chiều làm dấy lên mối lo ngại rằng các token có thể ngầm đoán trước các token trong tương lai trong quá trình đào tạo trước, dẫn đến việc học tập tối thiểu và dẫn đến những dự đoán tầm thường. Để đào tạo một mô hình như vậy một cách hiệu quả, BERT triển khai Mô hình ngôn ngữ đeo mặt nạ (MLM) che dấu ngẫu nhiên 15% tất cả các mã thông báo đầu vào trong mỗi chuỗi đầu vào. Dự đoán từ bị che này là yêu cầu mới không giống như việc tạo lại toàn bộ chuỗi đầu ra trong LM một chiều.

Mặt nạ BERT trong quá trình đào tạo trước, do đó mã thông báo [MASK] không hiển thị trong quá trình tinh chỉnh, tạo ra sự không khớp vì mã thông báo '' bị che '' không được thay thế. Để khắc phục sự khác biệt này, các sửa đổi mô hình tinh tế được thực hiện trong giai đoạn tiền đào tạo. Nếu một token Ti được chọn để che dấu thì



HÌNH 7. Kiến trúc chức năng MLM và NSP của BERT.

80% thời gian nó được thay thế bằng mã thông báo [MASK], 10% thời gian mã thông báo ngẫu nhiên được chọn và 10% còn lại, nó không thay đổi. Sau đó, việc mất entropy chéo Ti sẽ dự đoán mã thông báo ban đầu, bước mã thông báo không thay đổi được sử dụng để duy trì xu hướng dự đoán chính xác.

Phương pháp này tạo ra trạng thái ngẫu nhiên và học hỏi liên tục cho bộ mã hóa Transformer, buộc phải duy trì cách trình bày theo ngữ cảnh phân tán của từng mã thông báo. Hơn nữa, vì sự thay thế ngẫu nhiên chỉ xảy ra đối với 1,5% tổng số mã thông báo (10% của 15%), điều này dường như không làm giảm khả năng hiểu của mô hình ngôn ngữ.

Mô hình ngôn ngữ không thể hiểu rõ ràng mối liên hệ giữa nhiều chuỗi; do đó nó được coi là chưa tối ưu cho các nhiệm vụ suy luận và hỏi đáp. Để khắc phục điều này, BERT đã được đào tạo trước với ngữ liệu đơn ngữ cho nhiệm vụ Dự đoán câu tiếp theo (NSP) nhị phân.

Như được hiển thị trong Hình 7, các câu Y (Anh ấy đến [MASK] từ nhà) và Z (Trái đất [MẶT NẠ] quanh Mặt trời) không hình thành bất kỳ sự liên tục hoặc mối quan hệ nào. Vì Z không phải là câu tiếp theo thực sự sau Y nên nhãn phân loại đầu ra [NotNext] sẽ được kích hoạt và [IsNext] kích hoạt khi các chuỗi mạch lạc.

E. ĐÀO TẠO TRƯỚC TỰ ĐỘNG TỔNG HỢP CHO HIỂU NGÔN NGỮ: XLNet

XLNet nắm bắt những gì tốt nhất của cả hai thế giới, nơi nó duy trì các lợi ích của mô hình Tự động hồi quy (AR) và thu thập theo ngữ cảnh hai chiều. Để hiểu rõ hơn lý do tại sao XLNet hoạt động tốt hơn BERT, hãy xem xét chuỗi 5 mã thông báo [San, Fran-cisco, is, a, city]. Hai token được chọn để dự đoán là [San, Francisco], do đó BERT và XLNet tối đa hóa log p(San Francisco|is a city) như sau:

$$\begin{aligned} \text{LBERT} &= \log p(\text{San}|\text{là một thành phố}) \\ &+ \log p(\text{Francisco}|\text{là một thành phố}) \\ \text{LXLNet} &= \log p(\text{San}|\text{là một thành phố}) \\ &+ \log p(\text{Francisco}|\text{San là một thành phố}) \end{aligned}$$

Những điều trên có thể được khái quát hơn nữa cho bộ mã thông báo mục tiêu (T) và không phải mục tiêu (N), BERT và XLNet sẽ tối đa hóa

log p(T|N) với khả năng diễn giải khác nhau như sau:

LBERT = log p(x|N) (25)

x T

LBERT = log p(x|NT<x) (26)

x T

XLNet xem xét mục tiêu cũng như các mã thông báo còn lại để dự đoán, trong khi BERT chỉ xem xét các mã thông báo không phải mục tiêu. Do đó, XLNet nắm bắt được sự phụ thuộc giữa các cặp [San, Francisco] không giống như BERT trong đó [San] hoặc [Fran-cisco] dẫn đến dự đoán chính xác. Hơn nữa, thông qua AR XLNet, thứ tự được tính theo hệ số trên tất cả các hoán vị mã thông báo có thể có (L!= 5!) có độ dài chuỗi L trong tập hợp, tức là {[1, 2, 3, 4, 5], [1, 2, 5 , 4, 3],..., [5, 4, 3, 2, 1]} = [is, San, Francisco, a, city] v.v.

tối đa T
θ Ez ZT log pθ (xzt | xz<t) (27)
t=1

trong đó tập ZT chứa tất cả các chuỗi hoán vị có độ dài T [1, 2, ..., T] và xzt là mã thông báo tham chiếu. Do đó, mục tiêu học từ nhiều sự kết hợp để đạt được việc học theo ngữ cảnh phong phú hơn. Hơn nữa, đối với tất cả các thứ tự phân tích nhân tử có thể hoán vị, các tham số mô hình được chia sẻ để xây dựng kiến thức và bối cảnh hai chiều từ tất cả các phân tích nhân tử như được thể hiện qua phương trình 27.

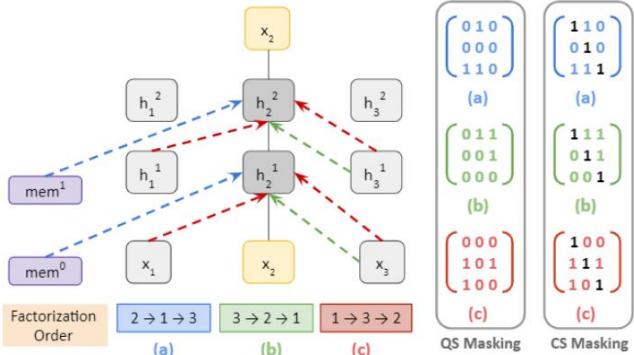
1) MASKING

Có một thách thức trong việc xác định thứ tự từ trong chuỗi vì mã thông báo (xzt) xác định quá trình tự hồi quy không được xem xét. Tuy nhiên, trật tự từ này đạt được một phần thông qua mã hóa vị trí, để hiểu theo ngữ cảnh, XLNet sử dụng mặt nạ. Hãy xem xét hoán vị được tạo của [2, 1, 3] trong chuỗi 3 mã thông báo trong đó mã thông báo đầu tiên tức là 2 không có ngữ cảnh do đó tất cả việc che dấu đều dẫn đến [0,0,0] ở hàng thứ 2 của mặt nạ 3 × 3 ma trận. Tương tự, mặt nạ thứ 2 và thứ 3 sẽ dẫn đến [0,1,0] và [1,1,0] ở hàng thứ 3 của ma

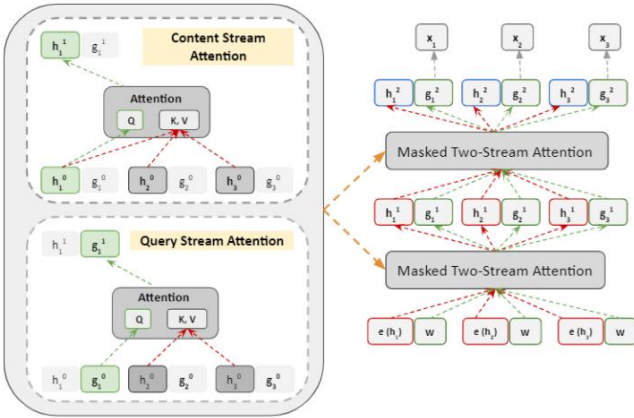
trận 1 che giấu Luồng truy vấn (QS) trong đó mã thông báo không thể nhìn thấy chính nó. Ma trận QS với sự bao gồm tất cả một đường chéo tạo thành ma trận che giấu Luồng nội dung (CS) trong đó mỗi mã thông báo có thể nhìn thấy chính nó. Việc che dấu chuỗi 3 mã thông báo này được thể hiện trong hình 8 bên dưới.

F. MÔ HÌNH KIẾN TRÚC

Hình 9 minh họa khung chú ý hai luồng của mô hình bao gồm quy trình chú ý luồng nội dung và truy vấn để đạt được sự hiểu biết tốt hơn thông qua bối cảnh hóa. Quá trình này được bắt đầu thông qua việc biểu diễn nhận biết mục tiêu, trong đó vị trí mục tiêu được đưa vào đầu vào cho các mục đích tạo mã thông báo tiếp theo.



HÌNH 8. Minh họa dự đoán x2 trong chuỗi 3 mã thông báo với các thứ tự nhân tử khác nhau và các ma trận che tương ứng của nó.



HÌNH 9. (Trái): Chú ý tiêu chuẩn thông qua luồng nội dung và chú ý luồng truy vấn mà không cần truy cập vào nội dung. (Phải): Đào tạo LM.

(i) Biểu diễn nhận biết mục tiêu: Việc triển khai tham số hóa dựa trên Transformer không đủ cho mô hình hóa ngôn ngữ dựa trên hoán vị phức tạp. Điều này là do phân phối mã thông báo tiếp theo pθ (Xzt | xz<t) độc lập với vị trí mục tiêu, tức là Zt. Sau đó, phân phối dự phòng được tạo ra, không thể khám phá các biểu diễn hiệu quả, do đó việc tái tham số hóa nhận biết vị trí mục tiêu cho phân phối mã thông báo tiếp theo được đề xuất như sau:

pθXzt = x | xz<t = (kính nghiệm (e (x)) hθ xz<t) / x kính nghiệm (e (x)) hθ xz<t) (28)

pθXzt = x | xz<t = (kính nghiệm e (x)) gθ xz<t , Zt / x kính nghiệm (e (x)) gθ xz<t , Zt) (29)

trong đó gθ (xz<t, Zt) là một biểu diễn được sửa đổi mà ngoài ra còn coi vị trí đích Zt là đầu vào.

(ii) Hai luồng tự chú ý: Việc xây dựng gθ vẫn là một thách thức mặc dù đã có giải pháp trên vì mục tiêu là dựa vào vị trí mục tiêu Zt để thu thập thông tin theo ngữ cảnh xz<t thông qua sự chú ý, do đó: (1) Để gθ đến dự đoán xzt, nó chỉ nên sử dụng vị trí của Zt để kết hợp việc học tập tốt hơn chứ không phải nội dung xzt (2) Để dự đoán các mã thông báo khác xzj trong đó j > t, gθ nên mã hóa bối cảnh xzt để cung cấp sự hiểu biết đầy đủ về ngữ cảnh.

Để giải quyết sâu hơn mâu thuẫn trên, tác giả đề xuất thay vào đó là hai bộ biểu diễn ẩn như sau:

Biểu diễn nội dung ẩn h_{θ} ($x < t$) = h_Z mã hóa cả ngữ cảnh và nội dung x_Z Biểu diễn truy vấn g_{θ} ($x < t$, Z_t)

= g_Z chỉ truy cập thông tin theo ngữ cảnh $x < t$ và vị trí Z_t không có nội dung x_Z

Hai khóa học chú ý ở trên được chia sẻ và cập nhật về mặt tham số cho mọi lớp tự chú ý m như sau: ($m-1$) ($m-1$)

Chú ý ($Q = h_{\theta} Z_t$, $KV = h_{\theta} Z_{<t}; \theta$) (m) (Z_t) (Nội dung

Luồng: sử dụng cả Z_t và x_Z) ($m-1$) ($m-1$) (m)

Chú ý ($Q = g_{\theta} Z_{<t}$, $KV = h_{\theta} Z_t$) (m) (Z_t) (Truy vấn

Sự chú ý kép này được thể hiện bằng hình ảnh trong hình 9. Để đơn giản, hãy xem xét dự đoán về mã thông báo ti không được phép truy cập vào phần nhúng tương ứng của nó từ lớp trước. Tuy nhiên, để dự đoán t_i+1 , mã thông báo t_i cần truy cập vào phần nhúng của nó và cả hai thao tác phải diễn ra trong một lần chuyển.

Do đó, hai biểu diễn ẩn được triển khai (m) (m) trong đó h được khởi tạo thông qua việc nhúng mã thông báo và g Z_t thông qua các phép biến đổi có vị trí hiện tại Z_t (m) trong khi các vị trí g dự đoán mã thông báo Z_t (m) xảy ra ở lớp cuối cùng thông qua g Z_t . Xử lý độ dài chuỗi lớn hơn, các khối bộ nhớ được lấy từ Transformer-XL, có thể xử lý dài hơn độ dài chuỗi đầu vào Transformer (m)

Các biểu diễn ẩn được áp dụng được Z_t trữ trong các khối tiêu chuẩn. Các biểu diễn ẩn được áp dụng được Z_t trữ trong các khối nhớ.

Vi

G. ĐÀO TẠO TRƯỚC BERT TỐI ƯU MẠNH MỀ

TIẾP CẬN: RoBERTa

Bài báo này tuyên bố rằng BERT đã được đào tạo thiếu đáng kể và kết quả là RoBERTa đã kết hợp một chế độ đào tạo chuyên sâu hơn. Điều này dành cho các mô hình dựa trên BERT có thể phù hợp hoặc vượt quá các phương pháp trước đó. Các sửa đổi của họ bao gồm: (i) thời gian đào tạo dài hơn với dữ liệu và kích thước lô lớn hơn (ii) loại bỏ mục tiêu NSP của BERT (iii) đào tạo theo trình tự dài hơn (iv) mẫu mặt nạ của dữ liệu đào tạo được sửa đổi linh hoạt.

Các tác giả khẳng định hiệu suất vượt trội so với BERT trong các tác vụ tiếp theo để có bộ dữ liệu CC-News đa dạng và phong phú hơn.

Hơn nữa, BERT thực hiện triển khai mặt nạ tính không hiệu quả để tránh mặt nạ dư thừa. Ví dụ: dữ liệu huấn luyện được sao chép 10 lần để một chuỗi được che theo 10 cách khác nhau trong 40 ký nguyên huấn luyện, trong đó mỗi chuỗi huấn luyện được nhìn thấy với cùng một mặt nạ 4 lần.

RoBERTa cung cấp các kết quả được nâng cao một chút thông qua việc kết hợp mặt nạ động trong đó mẫu mặt nạ được tạo ra mỗi khi mô hình được cung cấp một chuỗi trong khi huấn luyện trước các tập dữ liệu lớn hơn. Công việc gần đây đã đặt câu hỏi về vai trò NSP của BERT [60] được phỏng đoán là đóng một vai trò quan trọng trong hoạt động của nó trong các nhiệm vụ suy luận ngôn ngữ và Hỏi đáp. RoBERTa hợp nhất cả hai giả thuyết và cung cấp nhiều bổ sung

các định dạng đào tạo hoạt động giống như BERT và hoạt động tốt hơn nó đối với đào tạo câu đầy đủ, ngoại trừ mất NSP. RoBERTa cung cấp kết quả tương tự và tốt hơn một chút so với BERT trên điểm chuẩn GLUE cũng như trên bộ dữ liệu RACE và SQUAD mà không cần tinh chỉnh cho nhiều tác vụ.

H. MÔ HÌNH NGÔN NGỮ MEGATRON (LM)

Megatron là model lớn nhất khi ra mắt với kích thước $24 \times$ BERT và $5,6 \times$ GPT-2 và không thể lắp vừa một GPU. Do đó, việc triển khai kỹ thuật quan trọng là tạo ra mô hình 8 và 64 chiều và phiên bản phân tích dữ liệu trong đó các tham số được phân chia thành (512)

GPU. Nó duy trì hiệu suất cao (15,1 Petaflops) và hiệu suất mở rộng (76%), trong khi BERT dẫn đến suy giảm hiệu suất khi tăng trưởng kích thước. Thành tích này chủ yếu là do việc chuẩn hóa lớp và sắp xếp lại kết nối dư trong các lớp máy biến áp. Điều này dẫn đến hiệu suất vượt trội về mặt đơn sắc đối với các tác vụ tiếp theo với kích thước mô hình tăng lên.

Megatron khắc phục hạn chế về bộ nhớ của mô hình trước đó bằng cách chia mô hình thành nhiều máy gia tốc. Điều này không chỉ giải quyết việc sử dụng bộ nhớ mà còn nâng cao tính chất song song của mô hình bất kể kích thước lô. Nó kết hợp các phép tính tensor phân tán để tăng kích thước hoặc khả năng tăng tốc của mô hình và song song hóa tính toán đầu chú ý. Điều này không yêu cầu trình biên dịch mới hoặc viết lại mã và có thể triển khai được với một vài tham số.

Đầu tiên, khối Perceptron nhiều lớp (MLP) phân chia GEMM song song thành hai cột, cho phép tính phi tuyến tính GeLU được áp dụng độc lập cho từng GEMM được phân vùng.

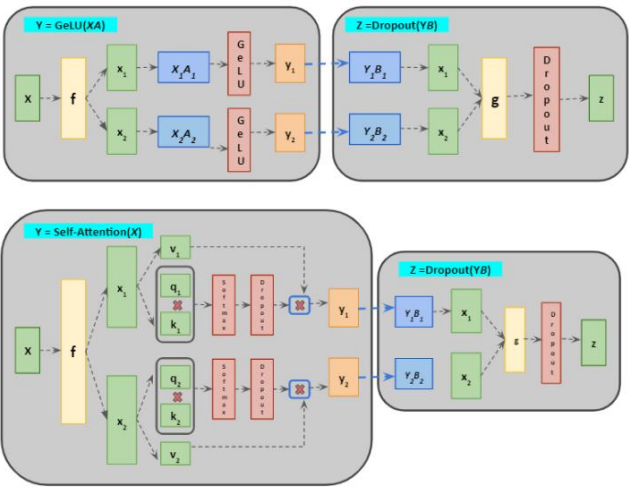
Đầu ra GeLU này được đưa trực tiếp đến GEMM song song theo hàng có đầu ra được giảm thông qua một toán tử giảm toàn bộ duy nhất (g và f) trong quá trình chuyển tiếp và lùi trước khi chuyển nó đến lớp bỏ học.

Tính song song trong khối tự chú ý đạt được bằng cách phân vùng các GEMM theo cột cho từng bộ khóa, truy vấn và giá trị. Do đó, khối lượng công việc được chia cho tất cả các GPU khi phép nhân ma trận cho mỗi đầu chú ý được thực hiện trên một GPU duy nhất. Đầu ra GEMM thu được, giống như MLP, trải qua hoạt động thu gọn hoàn toàn và được song song giữa các hàng như minh họa ở trên trong hình 10. Kỹ thuật này loại bỏ nhu cầu đồng bộ hóa giữa các GEMM cho MLP và các khối chú ý.

V. KIẾN TRÚC NLG Trong các mô hình

NLU, lượng dữ liệu tính toán khổng lồ cần thiết để tìm hiểu nhiều nhiệm vụ 'tinh chỉnh' được đào tạo trước là không hiệu quả về mặt tham số, vì mọi nhiệm vụ đều cần một mô hình hoàn toàn mới. Những mô hình này có thể được lấy làm ví dụ như những chuyên gia hẹp hơn là những nhà tổng quát thành thạo. Do đó, các mô hình NLG cung cấp sự chuyển đổi theo hướng xây dựng các hệ thống chung, hoàn thành một số nhiệm vụ mà không cần phải tạo và gắn nhãn tập dữ liệu huấn luyện theo cách thủ công cho từng nhiệm vụ.

Hơn nữa, MLM trong các mô hình NLU không thể nắm bắt được mối quan hệ phong phú giữa nhiều chuỗi. Hơn nữa, hầu hết các mô hình NLU hiệu quả đều lấy phương pháp luận của chúng từ MLM.



HÌNH 10. MLP song song của megatron và các khối tự chú ý.

các biến thể mô hình là bộ mã hóa tự động khử nhiễu được huấn luyện về tái tạo văn bản trong đó một tập hợp con các từ ngẫu nhiên bị che đi. Do đó, các mô hình NLG trong vài năm qua đã đạt được tiến bộ vượt bậc trong các nhiệm vụ như dịch và tóm tắt văn bản, Hỏi đáp, NLI, tương tác hội thoại, mô tả hình ảnh với độ chính xác chưa từng có.

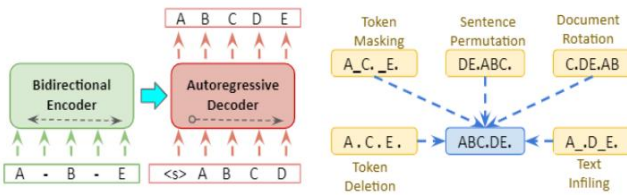
A. MÔ HÌNH NGÔN NGỮ LÀ NHIỆM VỤ ĐA NĂNG KHÔNG GIÁM SÁT
NGƯỜI HỌC: GPT-II

GPT-II [61] có thể là mẫu đầu tiên đánh dấu sự nổi lên của mẫu NLG. Nó được đào tạo theo cách không giám sát, có khả năng học các tác vụ phức tạp bao gồm Dịch máy, đọc hiểu và tóm tắt mà không cần tinh chỉnh rõ ràng. Đào tạo theo nhiệm vụ cụ thể tương ứng với tập dữ liệu của nó là lý do cốt lõi đằng sau sự thiếu hụt tính khái quát hóa được thấy trong các mô hình hiện tại. Do đó, các mô hình mạnh mẽ có thể sẽ yêu cầu đào tạo và đo lường hiệu suất trên nhiều lĩnh vực nhiệm vụ khác nhau.

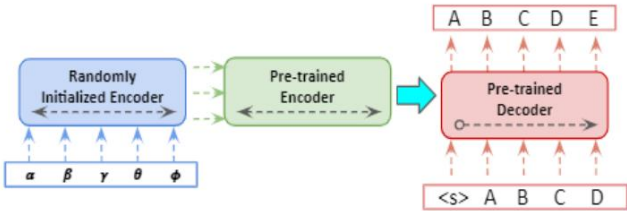
GPT-II kết hợp mô hình xác suất chung trong đó nhiều tác vụ có thể được thực hiện cho cùng một đầu vào dưới dạng p(đầu ra|đầu vào, tác vụ). Hiệu suất của tập huấn luyện và kiểm tra được cải thiện khi kích thước mô hình được tăng lên và kết quả là nó phù hợp với tập dữ liệu WebText khổng lồ. 1,5 tỷ tham số GPT-2 vượt trội so với các phiên bản tiền nhiệm của nó trên hầu hết các tập dữ liệu trong các tác vụ đã đề cập trước đó trong môi trường không có cảnh báo nào. Nó là phần mở rộng của kiến trúc chỉ dành cho bộ giải mã GPT-I được đào tạo trên dữ liệu lớn hơn đáng kể.

B. Máy biến áp hai chiều và tự hồi quy: BART

Bộ mã hóa tự động khử nhiễu BART là mô hình theo trình tự [62] kết hợp đào tạo trước hai giai đoạn: (1) Làm hỏng văn bản gốc thông qua chức năng nhiễu ngẫu nhiên và (2) Tái tạo văn bản thông qua đào tạo người mẫu. Tính linh hoạt về nhiễu là lợi ích chính của mô hình trong đó các phép biến đổi ngẫu nhiên không giới hạn ở việc thay đổi độ dài được áp dụng cho văn bản gốc. Hai biến thể ồn ào nổi bật



HÌNH 11. Mô hình BART đã khử nhiễu và các sơ đồ nhiễu của nó.



HÌNH 12. Mô hình BART đã được khử nhiễu cho các tác vụ MT được tinh chỉnh.

là sự xáo trộn thứ tự ngẫu nhiên của câu gốc và sơ đồ điền vào trong đó các văn bản có độ dài kéo dài bất kỳ được thay thế ngẫu nhiên bằng một mã thông báo bị che giấu duy nhất. BART triển khai tất cả các kế hoạch làm hỏng tài liệu có thể xảy ra như minh họa trong hình 11 bên dưới, trong đó trường hợp nghiêm trọng nhất là tất cả thông tin nguồn sẽ bị mất và BART hoạt động giống như một mô hình ngôn ngữ.

Điều này buộc mô hình phải phát triển lý luận tốt hơn trên toàn bộ chiều dài chuỗi tổng thể cho phép chuyển đổi đầu vào lớn hơn, mang lại khả năng khái quát hóa vượt trội hơn BERT.

BART được đào tạo trước thông qua việc tối ưu hóa tổn thất tái tạo được thực hiện trên các tài liệu đầu vào bị hỏng, tức là entropy chéo giữa đầu ra của bộ giải mã và tài liệu gốc. Đối với các tác vụ dịch máy, lớp nhúng bộ mã hóa của BART được thay thế bằng bộ mã hóa được khởi tạo tùy ý, được đào tạo từ đầu đến cuối với mô hình được đào tạo trước như trong Hình 12. Bộ mã hóa này ảnh xạ từ vựng nước ngoài của nó tới đầu vào của BART được ký hiệu bằng ngôn ngữ mục tiêu là tiếng Anh. Bộ mã hóa nguồn được đào tạo theo hai giai đoạn, chia sẻ sự lan truyền ngược của tổn thất entropy chéo từ đầu ra của BART. Thứ nhất, hầu hết các tham số BART đều bị đóng băng và chỉ bộ mã hóa được khởi tạo tùy ý, phần nhúng vị trí của BART và ma trận chiếu đầu vào tự chú ý của bộ mã hóa của nó mới được cập nhật. Thứ hai, tất cả các tham số của mô hình đều được huấn luyện chung trong vài lần lặp. BART đạt được hiệu suất tiên tiến trong một số nhiệm vụ tạo văn bản, thúc đẩy việc khám phá thêm các mô hình NLG. Nó đạt được kết quả so sánh đối với các nhiệm vụ có tính phân biệt khi so sánh với RoBERTa.

C. ĐÀO TẠO TRƯỚC GIẢI NHÔI ĐA NGÔN NGỮ DỊCH MÁY THẦN KINH: mBART

1) DỊCH MÁY ĐƯỢC GIÁM SÁT mBART chứng minh rằng đạt được mức tăng hiệu suất đáng kể so với các kỹ thuật trước đó [63], [64] bằng cách tự động đào tạo trước BART, thông qua mục tiêu khử nhiễu được tái tạo theo trình tự trên 25 ngôn ngữ từ quá trình thu thập thông tin chung (CC-25) từ thi [65]. khả năng tinh chỉnh tham số của mBART

được giám sát hoặc không giám sát đối với bất kỳ cặp ngôn ngữ nào mà không cần sửa đổi theo nhiệm vụ cụ thể. Ví dụ: việc tinh chỉnh một cặp ngôn ngữ (tiếng Đức-tiếng Anh) cho phép mô hình dịch từ bất kỳ ngôn ngữ nào trong tập huấn luyện trước đơn ngữ, tức là (tiếng Anh Pháp), không cần đào tạo thêm. Vì mỗi ngôn ngữ chứa các mã thông báo có các biến thể số đáng kể nên kho ngữ liệu được cân bằng thông qua việc lấy mẫu lên/xuống văn bản từ mỗi ngôn ngữ i với tỷ lệ λ_i

$$\lambda_i = \frac{1}{\pi_i} \cdot \frac{\alpha^{s_{p_i}}}{\alpha^{\sum_{i=1}^K s_{p_i}}} \tag{30}$$

trong đó π_i là tỷ lệ phần trăm của mỗi ngôn ngữ trong tập dữ liệu với tham số nhẹ nhàng $\alpha = 0.7$. Dữ liệu huấn luyện bao gồm K ngôn ngữ: $C = \{C_1, \dots, C_K\}$ trong đó mỗi C_i là tập hợp tài liệu đơn ngữ của ngôn ngữ i . Hãy xem xét hàm nhiễu làm hỏng văn bản $g(X)$ trong đó mô hình được huấn luyện để dự đoán văn bản gốc X , do đó tổn thất L0 được tối đa hóa là:

$$L_0 = - \sum_{C_i \in C} \log P(X | g(X); \theta) \tag{31}$$

trong đó ngôn ngữ i có phiên bản X trở lên phân phối P được xác định thông qua mô hình tuần tự.

2) DỊCH MÃY KHÔNG GIÁM SÁT mBART được đánh giá

trên các tác vụ trong đó các cặp văn bản hoặc văn bản đích không có sẵn ở 3 định dạng khác nhau này.

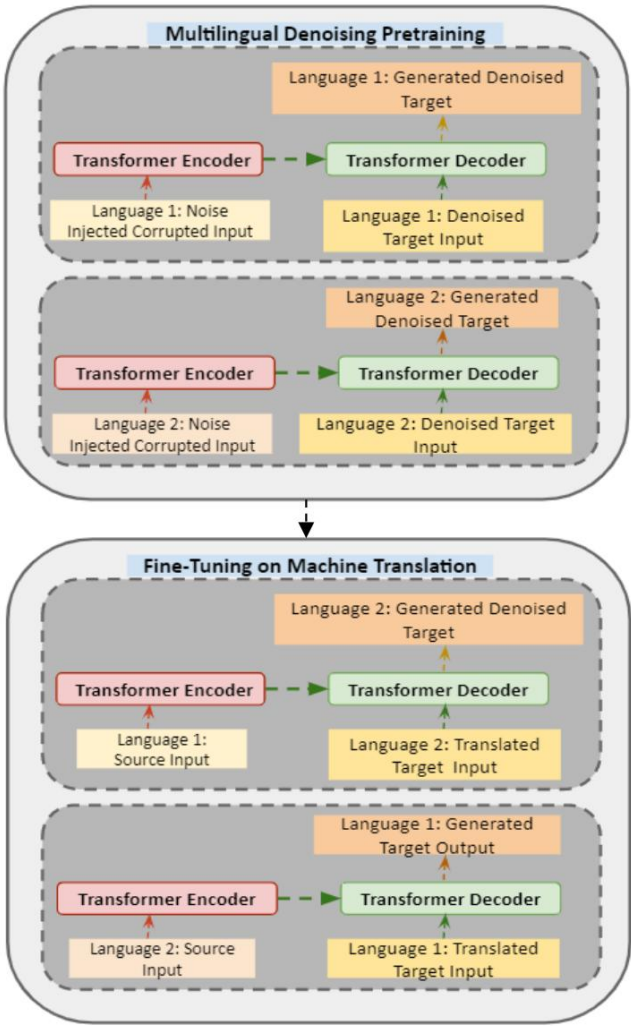
Không có loại bi-text nào được cung cấp, ở đây dịch ngược (BT) [66], [67] là một giải pháp quen thuộc. mBART cung cấp sơ đồ khởi tạo rõ ràng và hiệu quả cho các kỹ thuật như vậy.

Văn bản kép cho cặp của ngôn ngữ đích không khả dụng, tuy nhiên, cặp văn bản này có sẵn trong kho văn bản kép của ngôn ngữ đích cho các cặp ngôn ngữ khác. Văn bản bi không có sẵn cho cặp mục tiêu, tuy nhiên, có sẵn để dịch từ ngôn ngữ khác sang ngôn ngữ đích. Sơ đồ đánh giá mới này thể hiện khả năng học chuyển tiếp của mBART mặc dù không có văn bản song ngữ của ngôn ngữ nguồn mBART được đào tạo trước cho tất cả 25 ngôn ngữ và được tinh chỉnh cho ngôn ngữ đích như trong hình 13.

D. KHÁM PHÁ GIỚI HẠN CỦA CHUYỂN HỌC VỚI MÁY CHUYỂN ĐỔI TEXT-TO-TEXT: T5

Mô hình này được xây dựng bằng việc khảo sát và áp dụng các phương pháp học tập chuyển giao hiệu quả nhất. Ở đây, tất cả các tác vụ NLP được sắp xếp trong cùng một mô hình và các siêu tham số được sắp xếp lại thành một thiết lập chuyển văn bản thành văn bản thống nhất trong đó các chuỗi văn bản là đầu vào và đầu ra. Cần có một tập dữ liệu chất lượng cao, đa dạng và rộng lớn để đo lường hiệu quả mở rộng của quá trình đào tạo trước trong 11 tỷ tham số T5. Vì vậy, Colos-sal Clean Crawled Corpus (C4) được phát triển, lớn gấp đôi Wikipedia.

Các tác giả kết luận rằng việc che giấu nhân quả sẽ hạn chế khả năng của mô hình chỉ tham dự cho đến mục nhập đầu vào thứ i của chuỗi, điều này trở nên bất lợi. Do đó T5 kết hợp



HÌNH 13. Đào tạo trước và tinh chỉnh mô hình tổng quát mBART.

mặt nạ hiển thị đầy đủ trong phần tiền tố của chuỗi (tiền tố LM) trong khi mặt nạ nhân quả được kết hợp để huấn luyện dự đoán của mục tiêu. Các kết luận sau đây được đưa ra sau khi khảo sát bối cảnh học tập chuyển giao hiện tại. Cấu hình mô hình: Thông thường các mô hình có kiến trúc Bộ mã hóa-Giải mã hoạt động tốt hơn các mô hình ngôn ngữ dựa trên bộ giải mã. Mục tiêu đào tạo trước: Khử nhiễu hoạt động tốt nhất cho vai trò điền vào chỗ trống trong đó mô hình được đào tạo trước để truy xuất các từ còn thiếu đầu vào với chi phí tính toán chấp nhận được. Bộ dữ liệu trong miền: Đào tạo dữ liệu trong miền hóa ra là tuy nhiên, việc đào tạo trước các tập dữ liệu nhỏ thường dẫn đến việc trang bị quá mức. Phương pháp đào tạo trước, tinh chỉnh để học đa nhiệm có thể có hiệu quả, tuy nhiên, cần phải theo dõi tần suất đào tạo của từng nhiệm vụ. Chia tỷ lệ về mặt kinh tế: Để truy cập hiệu quả các tài nguyên điện toán hữu hạn, việc đánh giá giữa việc mở rộng quy mô mô hình, thời gian đào tạo và số lượng mô hình tổng hợp được thực hiện.

E. TURING TẠO NGÔN NGỮ TỰ NHIÊN: T-NLG

T-NLG là mô hình ngôn ngữ tổng hợp dựa trên Transformer gồm 78 lớp, vượt trội hơn T5 với 17 tỷ tham số có thể huấn luyện. Nó sở hữu tốc độ nhanh hơn Mega-tron của Nvidia, dựa trên việc kết nối nhiều máy thông qua các bus có độ trễ thấp. T-NLG là một mô hình ngày càng lớn hơn, được đào tạo trước với lượng dữ liệu đa dạng và đa dạng hơn. Nó mang lại kết quả vượt trội trong các nhiệm vụ tổng quát tiếp theo với các mẫu tinh chỉnh ít hơn. Do đó, các tác giả của nó đã khái niệm hóa việc đào tạo một mô hình đa nhiệm tập trung không lồ với các tài nguyên được chia sẻ cho nhiều nhiệm vụ khác nhau, thay vì phân bổ từng mô hình cho một nhiệm vụ. Do đó, mô hình thực hiện việc trả lời câu hỏi một cách hiệu quả mà không cần bối cảnh trước, dẫn đến việc nâng cao khả năng học tập không cần trả lời. Trình tối ưu hóa dự phòng bằng không (ZeRO) đạt được đồng thời cả mô hình và dữ liệu, đây có lẽ là lý do chính để đào tạo T-NLG với thông lượng cao.

F. MÔ HÌNH NGÔN NGỮ LÀ NGƯỜI HỌC ÍT HỌC: GPT-III

Họ GPT (I, II và III) là các mô hình ngôn ngữ tự hồi phục, dựa trên các khối bộ giải mã biến áp, không giống như BERT dựa trên bộ mã hóa tự động denoising. GPT-3 được đào tạo trên 175 tỷ tham số từ bộ dữ liệu gồm 300 tỷ mã thông báo văn bản được sử dụng để tạo các ví dụ đào tạo cho mô hình. Vì GPT-3 có kích thước gấp 10 lần bất kỳ mô hình ngôn ngữ nào trước đó và đối với tất cả các nhiệm vụ cũng như mục đích, nó sử dụng phương pháp học tập nhanh chóng thông qua giao diện văn bản mà không cần cập nhật độ dốc hoặc tinh chỉnh nên nó đạt được tính chủ nghĩa của nhiệm vụ. Nó sử dụng đào tạo trước không có giám sát, trong đó mô hình ngôn ngữ có được nhiều kỹ năng và khả năng nhận dạng mẫu. Chúng được triển khai nhanh chóng để nhanh chóng thích ứng hoặc xác định nhiệm vụ mong muốn. GPT-3 đạt được SOTA trong một số nhiệm vụ NLP mặc dù quá trình học vài lần của nó không thể tạo ra kết quả tương tự cho các nhiệm vụ khác.

G. CHIA SẺ MÔ HÌNH KHÔNG LỖ VỚI TÍNH TOÁN CÓ ĐIỀU KIỆN VÀ CHIA SẺ TỰ ĐỘNG:

GShard GShard cho phép mở rộng quy mô vượt quá 600 tỷ tham số cho dịch máy đa ngôn ngữ thông qua một nhóm chuyên gia (MoE) được kiểm soát thừa thớt bằng cách phân chia tự động với chi phí tính toán và thời gian biên dịch thấp. Máy biến áp được thu nhỏ quy mô bằng cách tạo ra một hỗn hợp các lớp chuyên gia (MoE) theo vị trí bao gồm các mạng chuyên tiếp nguồn cấp dữ liệu E FFN1, . . .

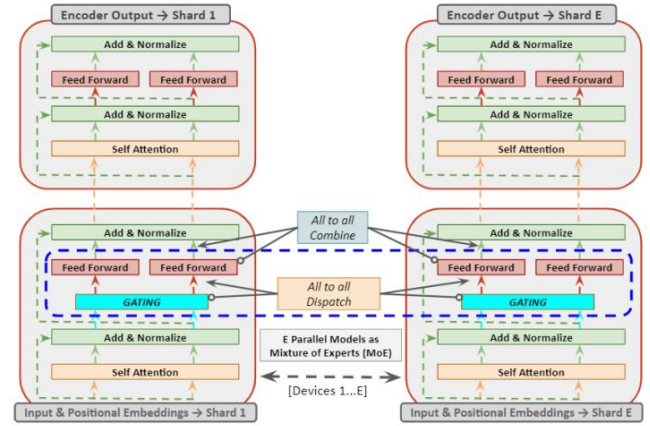
, FFNE trên Máy biến áp của nó.

Gs,E = CÔNG (xs) (32)

FFNe (xs) = woe.ReLU (wie.xs) (33)

văng =
$$e=1 \sum_{i=1}^E Gs,E.FFNe (xs) \tag{34}$$

trong đó xs và ys là đầu vào được mã hóa và đầu ra có trọng số trung bình cho lớp MoE, wie và woe là ma trận chiếu đầu vào và đầu ra của chuyên gia (lớp chuyển tiếp). Mạng cổng cho biết sự đóng góp của chuyên gia vào kết quả đầu ra cuối cùng thông qua vectơ Gs,E. Điều này nhận một giá trị khác 0 cho các mã thông báo được gửi tối đa là hai



HÌNH 14. Bộ mã hóa biến áp lớp MoE được phân chia khi được chia tỷ lệ thành nhiều thiết bị, tất cả các lớp khác sẽ được sao chép.

các chuyên gia đóng góp vào giá trị khác 0 trong một ma trận thừa thớt.

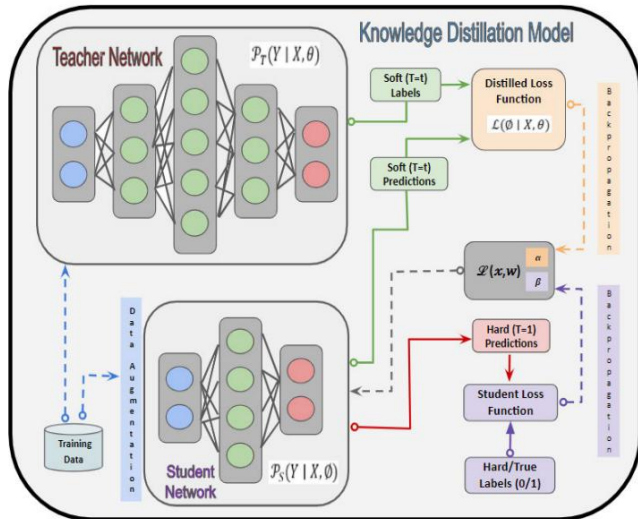
Để đạt được sự song song hiệu quả trên các cụm TPU: (i) Lớp chú ý song song được phân chia theo kích thước lô và trọng số được sao chép trên tất cả các thiết bị. (ii) Do hạn chế về kích thước, việc sao chép các chuyên gia lớp MoE trên tất cả các thiết bị là không khả thi, do đó các chuyên gia bị phân chia trên một số thiết bị, như minh họa bên dưới.

Hai yếu tố quyết định chất lượng mô hình là (i) Ngôn ngữ có nguồn lực cao, nơi có sẵn lượng dữ liệu đào tạo khổng lồ (ii) Cải tiến cho các ngôn ngữ có nguồn lực thấp với dữ liệu hạn chế. Các nhiệm vụ hoặc cặp ngôn ngữ tăng lên trong mô hình dịch thuật mang lại khả năng chuyển giao ngôn ngữ tích cực [68] cho các ngôn ngữ có nguồn tài nguyên thấp.

Chiến lược ba mũi nhọn để có thời gian đào tạo hợp lý và hiệu quả cho một số lượng lớn ngôn ngữ là: (i) Tăng độ sâu mạng bằng cách xếp chồng nhiều lớp hơn (ii) Tăng độ rộng mạng bằng cách sao chép các chuyên gia (iii) Phân bổ ít mã thông báo cho các chuyên gia thông qua định tuyến đã học mô-đun. Khi số lượng chuyên gia trên mỗi lớp tăng gấp bốn lần từ 128 lên 512 trong mô hình sâu 12 lớp, hiệu suất tăng đáng kể là 3,3 đã được quan sát thấy trong điểm BLEU trên 100 ngôn ngữ. Hơn nữa, việc tăng gấp bốn lần chiều rộng từ 512 lên 2048 dẫn đến mức tăng BLEU giảm dần đi 1,3. Việc tăng thêm gấp ba lần độ sâu lớp từ 12 lên 36 đối với độ rộng chuyên gia đã đề cập trước đó mang lại lợi ích đáng kể cho các ngôn ngữ có nguồn tài nguyên thấp cũng như cao. Tuy nhiên, độ sâu mô hình tăng lên sẽ không có kết quả trừ khi hạn chế về công suất của mô hình (chiều rộng MoE) không được nới lỏng.

VI. GIÁM KÍCH THƯỚC MÔ HÌNH A. Chứng cật

Mục tiêu của Chất lọc Kiến thức (KD) là đào tạo mô hình học sinh nhỏ hơn dưới sự giám sát của mô hình giáo viên lớn hơn, chính xác hơn thông qua hàm mất mát được sửa đổi để đạt được độ chính xác tương tự trên các mẫu không được dán nhãn. Các mẫu mô hình giáo viên dự đoán được cung cấp để cho phép học sinh học tập thông qua việc phân bổ lớp theo xác suất nhẹ nhàng hơn trong khi dự đoán thông qua phân loại mục tiêu cứng thông qua một hàm mất mát riêng biệt.



HÌNH 15. Kiến trúc tổng quát của mô hình ngôn ngữ.

Quá trình chuyển đổi nhãn cứng sang nhãn mềm này tạo ra sự khác biệt lớn hơn về thông tin cho việc học tập của học sinh, ví dụ: mục tiêu cứng phân loại chó là {bò, chó, mèo, ô tô 0, 1, 0, 0} và mục tiêu mềm là {10 6, 0, 9, 0, 1, 10 9}. Đối với tính toán phân loại cứng, lớp được kết nối đầy đủ cuối cùng của mạng nơ ron sâu là một vectơ logit z , trong đó z_i là logit của lớp i .

Do đó, xác suất p_i mà đầu vào phù hợp với i được đánh giá bằng hàm softmax trong (35) và phần tử nhiệt độ T được quy vào để tác động đến tầm quan trọng của từng mục tiêu mềm để chuyển sang mô hình học tập của học sinh trong (36).

$$p_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}; \quad (35)$$

$$p_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}; \quad (36)$$

Để phân bố xác suất mềm hơn trên các lớp, cần có nhiệt độ cao hơn ($T = t$). Về mặt thực nghiệm, người ta phát hiện ra rằng sẽ rất hiệu quả khi đào tạo mô hình học sinh về các nhãn chính xác (sự thật cứng/thực tế) ngoài các nhãn mềm của giáo viên.

Mặc dù mô hình sinh viên không thể khớp chính xác với các mục tiêu mềm, nhưng việc đào tạo nhãn cứng sẽ hỗ trợ thêm để nó không mắc phải dự đoán sai. Tổn thất chưng cất mục tiêu mềm ($T = t$) được tính bằng cách khớp các log giữa mô hình giáo viên và học sinh như sau:

$$LD(p(z_t, T), p(z_s, T)) = - \sum_i p_i(z_t, T) \log(p_i(z_s, T)) \quad (37)$$

trong đó z_t và z_s lần lượt biểu thị logit của mô hình giáo viên và học sinh. Cơ chế chưng cất được giải thích rõ ràng trong hình 15. Entropy chéo giữa nhãn chân lý cơ bản y và logit mềm của mô hình học sinh cấu thành sự mất mát của học sinh như sau:

$$L_s(y, p(z_s, T)) = - \sum_i y_i \log(p_i(z_s, T)) \quad (38)$$

Mô hình chất lọc kiến thức chuẩn mực tích hợp phần chất lọc và phần mất học sinh như hình dưới đây,

$$L(x, W) = \alpha \times LD(p(z_t, T), p(z_s, T)) + \beta \times L_s(y, p(z_s, T)) \quad (39)$$

trong đó W tham số sinh viên và tham số quy định α, β . Trong bài báo ban đầu, giá trị trung bình có trọng số được sử dụng liên quan đến α và β , tức là $\beta = 1 - \alpha$ và để có kết quả tốt nhất, người ta quan sát thấy rằng $\alpha = 0.9$.

1) DistilBERT

DistilBERT, phiên bản dành cho sinh viên của BERT giáo viên giữ lại 97% hiệu suất hiểu ngôn ngữ của BERT và tại thời điểm suy luận nhẹ hơn, nhanh hơn và yêu cầu chi phí đào tạo thấp hơn. Thông qua KD, DistilBERT giảm kích thước BERT xuống 40%, nhanh hơn 60% và mô hình nén đủ nhỏ để vận hành trên các thiết bị biên. Độ sâu lớp của Distil-BERT bị giảm một nửa khi so sánh với BERT vì cả hai đều có cùng kích thước và nhìn chung có kiến trúc tương đương. Việc giảm lớp được thực hiện vì quá trình chuẩn hóa và tối ưu hóa tuyến tính của nó không hiệu quả về mặt tính toán ở các lớp cuối cùng. Để tối đa hóa độ lệch quy nạp của các mô hình được đào tạo trước lớn, DistilBERT đã giới thiệu hàm mất ba kết hợp tuyến tính quá trình chưng cất (LD) với đào tạo có giám sát (Lm1) hoặc mất mô hình ngôn ngữ đeo mặt nạ. Người ta quan sát thấy rằng việc bổ sung tổn thất trước đó bằng tổn thất cosine nhúng (Lcos) là có lợi vì nó căn chỉnh theo hướng các ẩn của giáo viên và học sinh.

vectơ trạng thái.

2) TinyBERT Để

khắc phục sự phức tạp trong quá trình chất lọc của mô hình đào tạo trước rồi tinh chỉnh, TinyBERT đã giới thiệu một quy trình chuyển giao kiến thức rõ ràng bằng cách tạo ra 3 hàm mất: (i) Đầu ra lớp nhúng (ii) Ma trận chú ý, Hid-den Các trạng thái từ Nhật ký đầu ra của máy biến áp (iii). Điều này không chỉ giúp TinyBERT duy trì hơn 96% hiệu suất của BERT ở kích thước giảm đáng kể mà còn triển khai 28% tham số và 31% thời gian suy luận trên tất cả các mô hình chưng cất dựa trên BERT. Hơn nữa, nó tận dụng tiềm năng có thể trích xuất chưa được khai thác từ trọng số chú ý đã học của BERT [69], đối với lớp thứ $(M + 1)$, kiến thức thu được được nâng cao bằng cách giảm thiểu:

$$L_{model} = \sum_{m=0}^{M+1} \lambda_m L_{layer}(S_m, T_g(m)) \quad (40)$$

trong đó L_{layer} là hàm mất mát của lớp Transformer hoặc lớp Nhúng và siêu tham số λ_m biểu thị tầm quan trọng của quá trình chưng cất của lớp m . Tính năng nâng cao khả năng hiểu ngôn ngữ dựa trên chú ý của BERT có thể được tích hợp trong TinyBERT dưới dạng:

$$L_{attn} = h \sum_{t=1}^h \text{MSE}(A_{t \times t}^{S, T}, I_{1 \times 1}), \text{ trong đó } A_i \in \mathbb{R}^{h \times h} \quad (41)$$

trong đó h là số đầu, A_i là ma trận chú ý tương ứng với đầu i của học sinh hoặc giáo viên, l là độ dài văn bản đầu vào cùng với hàm mất mát sai số bình phương trung bình (MSE). Hơn nữa, TinyBERT chất lọc kiến thức từ lớp đầu ra của Máy biến áp và có thể được thể hiện

BẢNG:

$$L_{hidn} = \text{MSE}(H \cdot S_{Wh}, H^T)$$

(42)

trong đó W_h là ma trận trọng số $l \times d$, H là ma trận kích thước $l \times d$, H^T là ma trận kích thước $l \times d$, $d < d$ đó là ma trận trọng số $l \times d$ của học sinh và giáo viên, kích thước ẩn của mô hình giáo viên và học sinh được biểu thị thông qua các giá trị vô hướng của d và d , W_h là ma trận có thể học được, biến đổi các trạng thái ẩn của mạng học sinh thành trạng thái không gian của mạng giáo viên. Tương tự, TinyBERT cũng thực hiện chung cất trên lớp những:

$$L_{embd} = \text{MSE}(E \cdot S_{We}, E^T)$$

(43)

trong đó E là ma trận trọng số $l \times d$ và H là các ma trận nhúng của mạng học sinh và giáo viên. Ngoài việc bắt chước hành vi của lớp trung gian, TinyBERT còn triển khai KD để phù hợp với các dự đoán của mô hình giáo viên thông qua việc mất entropy chéo giữa log của học sinh và giáo viên.

$$L_{pred} = -\sum_{t=1}^T \log(\text{softmax}(z_t^S))$$

(44)

sinh Here z^S và z là các log tương ứng được dự đoán bởi mô hình giáo viên và học sinh.

3) MobileBERT Không

giống như các mô hình chất lọc trước đó, MobileBERT đạt được khả năng nén bất khả tri về nhiệm vụ từ BERT để đạt được sự hội tụ đào tạo thông qua dự đoán và mất mát chưng cất. Để đào tạo một mô hình mỏng sâu như vậy, một mô hình giáo viên thất cổ chai ngược độc đáo được thiết kế kết hợp BERT (IB-BERT) từ nơi chuyển giao kiến thức sang MobileBERT. Nó nhỏ hơn 4,3×, nhanh hơn 5,5× so với BERT và đạt được điểm cạnh tranh thấp hơn 0,6 đơn vị so với BERT trong các nhiệm vụ suy luận dựa trên GLUE. Hơn nữa, độ trễ thấp 62 ms trên điện thoại Pixel 4 có thể là do việc thay thế Chuẩn hóa lớp và kích hoạt gelu bằng phép biến đổi tuyến tính dựa trên sản phẩm Hadamard () đơn giản hơn.

$$\text{NoNorm}(h) = Y \cdot h + \beta, \text{ trong đó } Y, \beta \in \mathbb{R}^N$$

(45)

Để chuyển giao kiến thức, sai số bình phương trung bình giữa các bản đồ tính năng của MobileBERT và IB-BERT được triển khai như một mục tiêu chuyển giao.

$$L_{FMT} = \frac{1}{TN} \sum_{t=1}^T \sum_{n=1}^N (H_{t,1,n}^{tr} - H_{t,1,n}^{st})^2$$

(46)

Trong đó l là chỉ mục lớp, T là độ dài chuỗi, N là kích thước bản đồ đặc trưng. Để TinyBERT khai thác khả năng chú ý từ BERT, độ phân kỳ KL được giảm thiểu giữa mỗi đầu

phân phối của hai mô hình, trong đó A biểu thị số lượng đầu chú ý.

$$L_{TAI} = \frac{1}{TA} \sum_{t=1}^T \sum_{a=1}^M \text{DKL}(a_{|a}^{tr} || a_{t,1,a}^{st})$$

(47)

Ngoài ra, tổn thất KD mới có thể được thực hiện trong quá trình đào tạo trước của MobileBERT bằng sự kết hợp tuyến tính giữa tổn thất MLM và NSP của BERT, trong đó α là siêu tham số nằm giữa $(0,1)$.

$$LPD = \alpha \text{MLM} + (1 - \alpha) \text{LKD} + \text{LNSP}$$

(48)

Đối với các mục tiêu nêu trên, 3 chiến lược đào tạo được đề xuất: (i) Chuyển giao kiến

thức phụ trợ: Chuyển giao trung gian thông qua sự kết hợp tuyến tính giữa mất mát chuyển giao tất cả các lớp và mất mát trước khi đào tạo. (ii) Chuyển giao kiến thức chung:

Để có kết quả vượt trội, 2 tổn thất riêng biệt được đề xuất trong đó MobileBERT được đào tạo với tất cả các lớp cùng chuyển tổn thất và thực hiện quá trình chưng cất được đào tạo trước. (iii) Chuyển giao kiến thức lũy tiến: Để giảm

thiểu việc chuyển lỗi từ lớp thấp hơn lên lớp cao hơn, đề xuất chia việc chuyển giao kiến thức thành các giai đoạn L lớp L trong đó mỗi lớp được đào tạo dần dần.

B. CẮT TỈA

Cắt tỉa [70] là một phương pháp trong đó các trọng số, độ lệch, lớp và kích hoạt nhất định được loại bỏ và không còn là một phần của lan truyền ngược của mô hình. Điều này tạo ra sự thưa thớt trong các phần tử như lớp ReLU hiển thị sau, chuyển đổi các giá trị âm thành 0 ((ReLU(x) : max(0, x))). Việc cắt tỉa lặp lại tìm hiểu các trọng số chính, loại bỏ những trọng số ít quan trọng nhất dựa trên các giá trị ngưỡng và đào tạo lại mô hình cho phép mô hình phục hồi sau khi cắt tỉa bằng cách thích ứng với các trọng số còn lại. Các mô hình NLP như BERT, RoBERTa, XLNet đã được cắt giảm 40% và giữ được hiệu suất 98%, tương đương với DistilBERT.

1) TỈA LỚP a: BỎ CẤU TRÚC

Kiến trúc này [71] loại bỏ ngẫu nhiên các lớp trong thời gian huấn luyện và kiểm tra, cho phép lựa chọn mạng con ở bất kỳ độ sâu mong muốn nào, vì mạng đã được đào tạo để cắt tỉa mạnh mẽ. Đây là bản nâng cấp từ các kỹ thuật hiện tại yêu cầu đào tạo lại một mô hình mới từ đầu thay vì đào tạo một mạng từ đó trích xuất nhiều mô hình nông. Việc lấy mẫu mạng con như Dropout [72] và DropConnect [73] này xây dựng một mạng lưới cắt tỉa mạnh mẽ hiệu quả nếu nhóm trọng số đồng thời được chọn thông minh bị loại bỏ. Về mặt chính thức, việc cắt bớt độ mạnh trong các mạng chính quy hóa có thể đạt được bằng cách giảm độc lập từng trọng số thông qua phân phối Bernoulli trong đó tham số $p > 0$ quy định tốc độ loại bỏ. Điều này có thể so sánh với tích từng điểm của ma trận trọng số W với ma trận mặt nạ $\{0, 1\}$ M được lấy mẫu tùy ý, $W_d = MW$.

Chiến lược loại bỏ lớp hiệu quả nhất là loại bỏ mọi lớp khác, trong đó tốc độ cắt tia p và loại bỏ các lớp ở độ sâu d sao cho $d \equiv 0 \pmod{1/p}$. Đối với N nhóm có tỷ lệ loại bỏ cố định p, số lượng nhóm trung bình được sử dụng trong quá trình huấn luyện mạng là $N(1 - p)$, do đó việc cắt bớt kích thước cho r nhóm, tốc độ loại bỏ lý tưởng sẽ là $p = 1 - r/N$. Cách tiếp cận này đã mang lại hiệu quả cao trong nhiều nhiệm vụ NLP và đã dẫn đến các mô hình có kích thước tương đương với các phiên bản BERT đã được chất lọc và thể hiện hiệu suất tốt hơn.

b: BERT CỦA NGƯỜI NGHÈO

Do việc tham số hóa quá mức của mạng lưới thần kinh sâu, nên không cần có sẵn tất cả các tham số tại thời điểm suy luận, do đó một số lớp bị loại bỏ một cách chiến lược dẫn đến kết quả cạnh tranh cho các nhiệm vụ tiếp theo [74]. Chiến lược thả xen kẽ lẻ mang lại kết quả vượt trội so với chiến lược thả hàng đầu và thậm chí xen kẽ trong khoảng $K = 2$ trên tất cả các nhiệm vụ. Chẳng hạn, trong mạng 12 lớp, thả: top - {11, 12}; chẵn-xen kẽ - {10, 12}; lẻ-xen kẽ - {9, 11}, đi đến kết luận là (i) bỏ liên tiếp hai lớp cuối cùng sẽ có hại hơn so với việc loại bỏ các lớp xen kẽ, và (ii) việc bảo tồn lớp cuối cùng có ý nghĩa lớn hơn các lớp trên cùng khác.

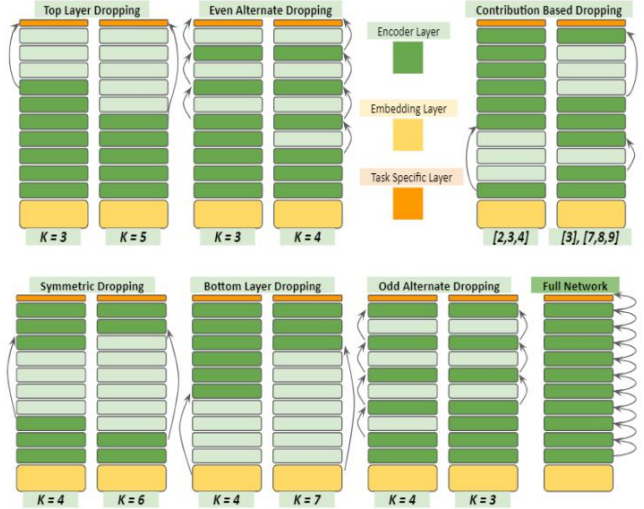
Ở giá trị K cao hơn, phương pháp loại bỏ thay thế biểu thị sự sụt giảm lớn về hiệu suất, được đưa ra giả thuyết là do loại bỏ các lớp thấp hơn. Cách tiếp cận đối xứng nhấn mạnh việc bảo toàn các lớp trên và dưới trong khi các lớp ở giữa bị loại bỏ. Điều này dẫn đến tác động tối thiểu đến BERT trong khi nó làm giảm đáng kể hiệu suất của XLNet, dẫn đến chiến lược tốt thứ hai cho BERT mang lại kết quả chắc chắn ngay cả sau khi loại bỏ 4 lớp.

Về mặt quan sát, XLNet thể hiện độ mạnh được cắt tia tốt hơn BERT khi quá trình học của nó diễn ra gần với lớp thứ 7 trong khi BERT tiếp tục học cho đến lớp thứ 11. Do đó (i) XLNet thu thập kiến thức theo định hướng nhiệm vụ ở các lớp thấp hơn trái ngược với BERT, (ii) các lớp cuối cùng của XLNet có thể trở nên khá dư thừa và có khả năng bị loại bỏ mà không làm giảm hiệu suất đáng kể. Các tác giả đã tiếp tục giảm thử nghiệm sang DistilBERT, ở đây việc giảm 30% số lớp của nó dẫn đến suy giảm hiệu suất ở mức tối thiểu.

Giống như các mô hình trước đó, việc thả lớp trên cùng tỏ ra đáng tin cậy nhất vì RoBERTa tỏ ra có khả năng cắt tia mạnh mẽ hơn BERT vì RoBERTa 6 lớp có hiệu suất tương tự như DistilRoBERTa. Tất cả các chiến lược thả ping lớp có thể được hình dung từ hình 16 ở trên.

2) TẮT TRỌNG LƯỢNG Công việc

trước đây tập trung chủ yếu vào việc cắt tia trọng lượng riêng lẻ không có cấu trúc [75], [76], mặc dù hiệu quả của nó là các ma trận thừa thớt phi cấu trúc đang gặp khó khăn khi xử lý trên phần cứng thông thường. Điều này gây khó khăn cho việc đảm bảo tốc độ suy luận mặc dù đã giảm kích thước mô hình. Việc cắt tia có cấu trúc ngược lại thực thi các ma trận trọng số có cấu trúc cao mà khi được tối ưu hóa thông qua đại số tuyến tính dày đặc



HÌNH 16. Chiến lược cắt tia lớp được triển khai bởi các mô hình ngôn ngữ.

thực hiện, dẫn đến tăng tốc đáng kể nhưng hiệu suất thấp hơn so với việc cắt tia không có cấu trúc do những hạn chế lớn hơn.

a: CẤU TRÚC CẤU TRÚC

Để khắc phục những thiếu sót trên, một mô hình cắt tia có cấu trúc mới đã được giới thiệu [77] với hệ số hóa thứ hạng thấp giữ lại cấu trúc ma trận dày đặc và chỉ tiêu l0 giúp nới lỏng các ràng buộc được thực thi thông qua việc cắt tia có cấu trúc. Các ma trận trọng số được phân tích thành tích của hai ma trận nhỏ hơn với mặt nạ đường chéo được cắt bớt trong khi huấn luyện thông qua bộ điều chỉnh l0 để kiểm soát độ thưa cuối của mô hình. Phương pháp chung FLOP (Cắt tia L0 nhân tố) này có thể được sử dụng cho bất kỳ phép nhân ma trận nào. Đối với mạng thần kinh f (; θ) được tham số hóa bởi $\theta = \{\theta_j\}_{j=1}^N$ trong đó mỗi θ_j đại diện cho một trọng số riêng lẻ hoặc một khối trọng số (ví dụ: ma trận cột) và n biểu thị số khối. Hãy xem xét một biến nhị phân cắt tia $z = \{z_j\}$ trong đó $z_j \in \{0, 1\}$, $\tilde{\theta} = \{\tilde{\theta}_j\}_{j=1}^N$ $\tilde{\theta}_j$ biểu thị tập tham số mô hình, cắt tia sau thông qua chuẩn hóa l0.

$$\tilde{\theta} = \theta z \quad \tilde{\theta}_j = \theta_j z_j \tag{49}$$

Xét ma trận W được phân tích thành nhân tử của hai ma trận nhỏ hơn P và Q trong đó $W = PQ$ và r là số P cột hoặc Q hàng. Việc cắt tia có cấu trúc cho từng thành phần đạt được thông qua biến cắt tia zk

$$W = PGQ = \sum_{k=1}^r z_k \times (p_k \times q_k) \quad \times \text{ trong đó } G = \text{diag}(z_1, \dots, z_r) \tag{50}$$

3) CẮT TỎA ĐẦU MẶC

dù một số mô hình nhất định có sự phụ thuộc lớn hơn vào các đầu nhiều đầu trong môi trường chú ý nhiều đầu, công việc gần đây cho thấy rằng một phần đáng kể các đầu chú ý có thể được loại bỏ, dẫn đến một mô hình được cắt tia với hiệu quả, tốc độ bộ nhớ được nâng cao và sự chính xác. Công việc trước đây [78], [79] đã đánh giá tầm quan trọng của phần đầu thông qua việc tính trung bình các trọng số chú ý

trên tất cả các đầu ở một vị trí cụ thể hoặc dựa trên kết quả của chúng dựa trên giá trị trọng số chú ý tối đa. Tuy nhiên, cả hai cách tiếp cận đều không xem xét một cách dứt khoát tầm quan trọng dao động của các phần đầu khác nhau.

a: PHÂN TÍCH SỰ TỰ CHÚ Ý CỦA NHIỀU ĐẦU: CÁC ĐẦU CHUYÊN DỤNG LÀM VIỆC NẶNG NẶNG, Phần còn lại CÓ THỂ ĐƯỢC CẮT LẠI Mô hình này [80] đã khai quật ba vai trò phân lớp của đầu: (i) Các đầu vị trí: Tham dự vào một mã thông báo liên kết (ii) Chiến thuật tổng hợp đầu: Chú ý đến những từ có sự phụ thuộc về cú pháp (iii) Đầu từ hiếm: Biểu thị các mã thông báo ít thường xuyên nhất trong một chuỗi. Dựa trên các vai trò trên [80], những phát hiện được tóm tắt là (a) Tập hợp con nhỏ các đầu là chìa khóa để dịch (b) Các đầu chính sở hữu một chức năng mô hình duy nhất, thường chuyên biệt hơn và dễ hiểu hơn (c) Các vai trò đầu tương ứng với các mã thông báo liên kết chú ý trong một mối quan hệ phụ thuộc cú pháp rõ ràng. Độ tin cậy cao thông qua Tuyên truyền mức độ liên quan theo lớp (LRP) [81] liên quan đến tỷ lệ chú ý của mã thông báo được xác định là trung bình của trọng số chú ý tối đa được tính trên tất cả các mã thông báo, được cho là rất quan trọng đối với một nhiệm vụ. Kiến trúc Trans-former được sửa đổi thông qua sản phẩm của headi phản hồi được tính toán của mỗi đầu và cổng vô hướng gi , MultiHead (Q,K, V) = Concat(gi .headi)W0, trong đó gi là các tham số cụ thể của đầu độc lập đầu vào, chính quy hóa L0 là áp dụng cho gi cho các đầu ít quan trọng hơn cần được vô hiệu hóa, trong đó h (số đầu).

$$L_0(g_1 \dots g_h) = \prod_{t=1}^h (1 - \prod_{i=1}^t [g_i = 0]) \quad (51)$$

Tuy nhiên, chuẩn L0 là không khả vi; do đó nó không thể được quy nạp như một thuật ngữ chính quy hóa trong hàm mục tiêu. Do đó, một sự phục hồi ngẫu nhiên được áp dụng trong đó mỗi cổng gi được chọn ngẫu nhiên từ phân bố đầu thu được thông qua việc kéo dài (0, 1) đến (- , 1+) và thu gọn phân bố xác suất (- , 1] thành [1, 1+) đến các điểm kỳ dị 0 và 1. Việc kéo dài hiệu chỉnh này dẫn đến một phân bố trên [0,1] được hỗn hợp rời rạc-liên tục. Tổng xác suất của các đầu khác 0 có thể được thực hiện dưới dạng chuẩn L0 thoải mái.

$$LC(\theta) = \prod_{t=1}^h (1 - P(g_i = 0 | i)) \quad (52)$$

Chế độ huấn luyện sửa đổi có thể được biểu thị bằng $L(\theta, \lambda) = L_{xent}(\theta, \lambda) + \lambda LC(\theta)$, trong đó θ là tham số ban đầu của Trans-former, $L_{xent}(\theta, \lambda)$ là mất entropy chéo của mô hình dịch thuật và $LC(\theta, \lambda)$ là bộ điều chỉnh.

b: 16 cái đầu có thực sự tốt hơn một cái không?

Trong sự chú ý nhiều đầu (MHA), hãy xem xét một chuỗi các vectơ thứ nguyên $x = x_1, \dots, x_n \in \mathbb{R}^d$ và vectơ truy vấn d_v , d_k và d_v và d_k và d_v ,
· Tham số lớp MHA $W_h \in \mathbb{R}^{d \times d_h}$, $q, v \in \mathbb{R}^{d_h}$
 d_v , khi $d_h = d$. Để che giấu sự chú ý

phương trình máy biến áp ban đầu được sửa đổi như sau:

$$MHAttn(x, q) = \sum_{h=1}^{Nh} \xi_h AttnWh_{q, \dots, Cat[gi^{cái} gi^{cái} gi^{cái}]}(x, q) \quad (53)$$

trong đó ξ_h là các biến che dấu có giá trị từ $\{0, 1\}$, $Attn(x)$ là đầu ra của đầu h cho đầu vào x . Các thí nghiệm sau đây mang lại kết quả tốt nhất [82] về việc tia số lượng đầu khác nhau tại các thời điểm thử nghiệm:

- (i) Chỉ cắt bớt một đầu: Nếu hiệu suất của mô hình giảm đáng kể trong khi che phần đầu h thì h là đầu chính, nếu không nó sẽ dư thừa đối với phần còn lại của mô hình. Chỉ có 8 đầu (trong số 96) đầu gây ra thay đổi đáng kể về hiệu suất khi bị loại khỏi mô hình, trong đó một nửa dẫn đến điểm BLEU cao hơn. (ii) Cắt bớt tất cả các đầu trừ một đầu: Một đầu duy nhất cho hầu hết các lớp được coi là đủ tại thời điểm thử nghiệm, ngay cả đối với các mạng có 12 hoặc 16 đầu chú ý, dẫn đến giảm tham số đáng kể. Tuy nhiên, nhiều đầu chú ý là yêu cầu bắt buộc đối với các lớp cụ thể, tức là lớp cuối cùng của chú ý bộ mã hóa-giải mã, trong đó hiệu suất giảm đi rất nhiều 13,5 điểm BLEU trên một đầu.

Độ nhạy dự kiến của mô hình đối với mặt nạ ξ được đánh giá theo điểm proxy về tầm quan trọng của phần đầu.

$$I_h = E_x \left[X \frac{L(x)}{\xi_h} \right] \quad (54)$$

$$I_h = E_x \left[X Attn(x) \frac{T \frac{L(x)}{Attn(x)}}{Attn(x)} \right] \quad (55)$$

trong đó X là phân bố dữ liệu, $L(x)$ là tổn thất trên mẫu x . Nếu I_h cao thì việc sửa đổi ξ_h có thể sẽ có tác động đáng kể đến mô hình, do đó các đầu có giá trị I_h thấp sẽ bị loại bỏ lặp đi lặp lại.

C. QUANTIZATION Đầu

phẩy động 32 bit (FP32) là định dạng số chiếm ưu thế cho học sâu, tuy nhiên, sự gia tăng hiện nay về bằng thông và tài nguyên điện toán giảm đã thúc đẩy việc triển khai các định dạng có độ chính xác thấp hơn. Người ta đã chứng minh rằng trọng số và biểu diễn kích hoạt thông qua số nguyên 8 bit (INT8) không dẫn đến mất độ chính xác rõ ràng. Ví dụ: việc lượng tử hóa BERT sang định dạng trọng lượng 16/8-bit dẫn đến nén mô hình 4× với mức độ mất độ chính xác tối thiểu, do đó, BERT được nâng cấp sẽ phục vụ hàng tỷ yêu cầu CPU mỗi ngày.

1) LQ-NETS Mô hình này [83] tạo ra cơ chế huấn luyện trọng số và kích hoạt mạng đơn giản thông qua việc huấn luyện chung mạng lưới thần kinh sâu. Nó lượng tử hóa với khả năng có độ chính xác bit thay đổi không giống như các sơ đồ cố định hoặc thủ công [84], [85]. Nói chung, một hàm lượng tử hóa có thể biểu diễn các trọng số dấu phẩy động w , kích hoạt a , trong một vài bit như sau:

$$Q(x) = q_l, \text{ if } x \in (t_l, t_{l+1}] \text{ trong đó } q_l, l = (1, \dots, L) \quad (56)$$

Ở đây q_l và (t_l, t_{l+1}) lần lượt là các mức lượng tử hóa và các khoảng. Để duy trì thời gian suy luận nhanh, các hàm lượng tử hóa cần phải tương thích với các phép toán theo bit, điều này đạt được thông qua phân phối đồng đều

ánh xạ các số dấu phẩy động tới các số nguyên dấu phẩy cố định gần nhất của chúng với hệ số chuẩn hóa. Hàm lượng tử hóa có thể học được LQ có thể được biểu diễn dưới dạng:

$$QLQ(x, v) = v \cdot \lfloor \frac{x}{e_l} \rfloor, \text{ nếu } x \in (t_l, t_{l+1}] \tag{57}$$

trong đó v_K là cơ sở dấu phẩy động có thể học được và $e_l \in \{1, 2\}$ với $l = (1, \dots, L)$ liệt kê các mã hóa nhị phân K-bit từ $[1, \dots, 1]$ đến $[1, \dots, 1]$. Việc tính toán tích số bên trong của các trọng số lượng tử hóa và các kích hoạt được tính toán bằng các phép toán theo bit sau đây với độ rộng bit trọng số Kw.

$$QLQ(w, v) \cdot QLQ(a, v) = \sum_{i=1}^{Kw} \sum_{j=1}^{Ka} w_{ij} \cdot a_{jv} \tag{58}$$

trong đó $w, a \in \mathbb{R}^N$ được mã hóa bởi vector bit $\{b_i^w, b_j^a\} \in \mathbb{R}^{Kw, Ka}$, đó $i = 1, \dots, Kw$ và $j = 1, \dots, Ka$ và v biểu thị v trong \mathbb{R}^{Ka} , phép toán xnor tích số bên trong theo bit.

2) QBER

QBERT [86] triển khai lượng tử hóa BERT hai chiều với đầu vào $x \in \mathbb{R}^D$, nhãn tương ứng của nó là $y \in \mathbb{R}^D$ thông qua hàm mất mát dựa trên entropy chéo

$$L(\theta) = -\sum_{(x_i, y_i)} \text{CE}(\text{softmax}(W_c(W_n(\dots W_1(W_e(x_i))))), y_i) \tag{59}$$

trong đó W_e là bảng nhúng, với các lớp mã hóa W_1, W_2, \dots, W_n và bộ phân loại W_c . Việc chỉ định biểu diễn kích thước bit giống nhau cho các lớp mã hóa khác nhau với độ nhảy khác nhau đối với các cấu trúc khác nhau [5] là chưa tối ưu và trở nên phức tạp đối với kích thước mục tiêu nhỏ (2/4 bit) yêu cầu độ chính xác cực thấp. Do đó, thông qua Lượng tử hóa nhận thức Hessian (HAWQ), nhiều bit hơn được gán cho các lớp nhảy cảm hơn để duy trì hiệu suất. Ma trận Hessian được tính toán thông qua kỹ thuật lặp không ma trận tiết kiệm về mặt tính toán trong đó gradient bộ mã hóa lớp đầu tiên g_1 cho một vector v từ trước.

$$\frac{g_1^T v}{W_1} = \frac{g_1^T}{W_1} v + g_1^T \frac{v}{W_1} = \frac{g_1^T}{W_1} v = H_1 v \tag{60}$$

Trong đó H_1 là ma trận Hessian của bộ mã hóa đầu tiên và v được lập với W_1 , phương pháp này xác định các giá trị riêng hàng đầu cho các lớp khác nhau và lượng tử hóa tích cực hơn được triển khai cho các lớp có giá trị riêng nhỏ hơn. Để tối ưu hóa hơn nữa thông qua lượng tử hóa theo nhóm, mỗi ma trận dày đặc được coi là một nhóm với phạm vi lượng tử hóa của nó và được phân chia theo từng nơ-ron đầu ra liên tục.

3) Q8BERT

Để lượng tử hóa trọng số và kích hoạt thành 8 bit, lượng tử hóa tuyến tính đối xứng được thực hiện [87], trong đó S là

hệ số tỷ lệ lượng tử hóa cho đầu vào x và $(M = 2^b - 1)$ là giá trị lượng tử hóa cao nhất khi lượng tử hóa thành b bit.

$$Lượng\ tử\ hóa\ x \rightarrow S, M : = \lceil \frac{x}{S} \rceil, M, M$$
$$Kep(x, a, b) = \min(\max(x, a), b) \tag{61}$$

Triển khai kết hợp lượng tử hóa giá [88] và Công cụ ước tính xuyên suốt (STE) [89], lượng tử hóa thời gian suy luận đạt được trong quá trình đào tạo với khả năng lan truyền ngược có độ chính xác hoàn toàn cho phép các trọng số FP32 khắc phục lỗi. Đây $\frac{1}{q} \cdot x = 1$, trong đó x, q là kết quả của việc lượng tử hóa giá x .

VII. TÌM KIẾM THÔNG TIN

Đối với các nhiệm vụ đòi hỏi nhiều kiến thức như cập nhật và truy xuất dữ liệu hiệu quả, cần phải có kho lưu trữ kiến thức tiềm ẩn khổng lồ. Các mô hình ngôn ngữ tiêu chuẩn không thành thạo trong các nhiệm vụ này và không phù hợp với các kiến trúc dành riêng cho nhiệm vụ có thể rất quan trọng đối với Hỏi & Đáp trong miền mở. Ví dụ: BERT có thể dự đoán từ còn thiếu trong câu "The is the money of the US" (câu trả lời: "dollar"). Tuy nhiên, vì kiến thức này được lưu trữ ngàm trong các tham số của nó nên kích thước sẽ tăng lên đáng kể để lưu trữ thêm dữ liệu. Ràng buộc này làm tăng độ trễ của mạng và khiến việc lưu trữ thông tin trở nên cực kỳ tốn kém vì không gian lưu trữ bị hạn chế do hạn chế về kích thước của mạng.

A. CHỖ CHỖ VÀNG

QA miền mở dựa trên nhiều bước truyền thống bao gồm câu hỏi q và từ một kho văn bản lớn chứa các tài liệu S (vàng) theo ngữ cảnh có liên quan d_1, \dots, d_s tạo thành một chuỗi lý luận thông qua các điểm tương đồng về văn bản dẫn đến câu trả lời ưa thích a . Tuy nhiên, bước nhảy đầu tiên của Golden Retriever [90] tạo ra một truy vấn tìm kiếm q_1 truy xuất tài liệu d cho một câu hỏi q nhất định, sau đó đối với các bước lý luận tiếp theo ($k = 2, \dots, S$), một truy vấn q_k được tạo ra từ câu hỏi (q) và ngữ cảnh có sẵn (d_1, \dots, d_{k-1}). Golden truy xuất các tài liệu theo ngữ cảnh lớn hơn theo cách lặp đi lặp lại trong khi nổi ngữ cảnh được truy xuất để mô hình QA của nó trả lời. Nó đọc lập với tập dữ liệu và các mô hình IR dành riêng cho nhiệm vụ trong đó việc lập chỉ mục các tài liệu bổ sung hoặc các loại câu hỏi dẫn đến sự thiếu hiệu quả. Một mô hình RNN nhẹ được điều chỉnh trong đó các khoảng văn bản được trích xuất từ dữ liệu theo ngữ cảnh để có khả năng giảm không gian truy vấn lớn. Mục đích là tạo ra truy vấn tìm kiếm q_k giúp truy xuất d_k cho bước tiếp theo, dựa trên khoảng văn bản từ ngữ cảnh C_k , q được chọn từ trình đọc tài liệu được đào tạo.

$$q_k = G_k(q, C_k), \tag{62}$$

$$C_{k+1} = C_k \text{ concat } IR_n(q_k) \tag{63}$$

trong đó G_k là trình tạo truy vấn và $IR_n(q_k)$ là n tài liệu được truy xuất hàng đầu thông qua q_k .

B. ORQA

Trình đọc và trình truy xuất thành phần được đào tạo chung theo kiểu từ đầu đến cuối trong đó BERT được triển khai để chấm điểm tham số. Nó có thể truy xuất bất kỳ văn bản nào từ một kho văn bản mở và

không bị hạn chế bằng cách trả về một bộ tài liệu cố định như mô hình IR thông thường. Việc tính toán điểm truy xuất là tích bên trong dày đặc q của câu hỏi với khối bằng chứng b .

$$h_q = W_q BERT_Q(q)[CLS] h_b = \quad (64)$$

$$W_b BERT_B(b)[CLS], \quad (65)$$

$$S_{retr}(b, q) = h_q^T h_b \quad (66)$$

trong đó ma trận W_q và W_b chiếu đầu ra BERT thành các vectơ 128 chiều. Tương tự, đầu đọc là biến thể nhập BERT của mô hình đọc.

$$h_{start} = BERT_R(q, b)[START(s)], \quad h_{end} \quad (67)$$

$$= BERT_R(q, b)[END(s)], \quad (68)$$

$$\text{Độ rộng}(b, s, q) \text{MLP}(h_{start}; h_{end}) \quad (69)$$

Mô hình truy xuất được đào tạo trước bằng Nhiệm vụ kết thúc nghịch đảo (ICT), trong đó ngữ cảnh của câu có liên quan về mặt ngữ nghĩa và được sử dụng để ngoại suy dữ liệu bị thiếu trong chuỗi q .

$$\text{HÌNH ẢNH}(b|q) = \frac{\exp(S_{retr}(b, q))}{\sum_{b \in \text{BATCH}} \exp(S_{retr}(b, q))} \quad (70)$$

trong đó q được coi là câu hỏi giả, b là văn bản bao quanh q và BATCH là tập hợp các khối bằng chứng được sử dụng để lấy mẫu phủ định.

Ngoài việc học các tính năng khớp từ, nó còn học các cách biểu diễn trừu tượng như câu hỏi giả có thể có hoặc không có trong bằng chứng. Hậu CNTT, việc học được xác định là phân phối dựa trên các dẫn xuất câu trả lời. $\exp(S(b, s, q)) \propto \text{TOP}(k) \propto \exp(S(b, s, q))$

$$P_{learn}(b, s|q) = \frac{\text{trong đó TOP}(k)}{\text{là các khối được truy xuất hàng đầu}} \quad (71)$$

dựa trên S_{retr} . Trong khuôn khổ này, việc truy xuất bằng chứng từ Wikipedia hoàn chỉnh được triển khai như một biến tiềm ẩn không thể huấn luyện từ đầu, do đó người truy xuất đã được huấn luyện trước bằng CNTT.

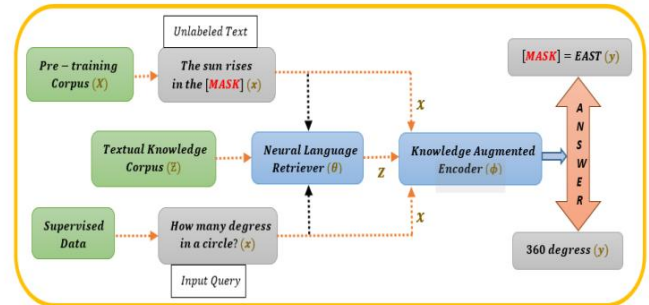
C. THỰC TẾ

Tuy nhiên, khung này liên quan rõ ràng đến một kho văn bản rộng lớn như Wikipedia, trình truy xuất của nó học thông qua lan truyền ngược và thực hiện tìm kiếm sản phẩm bên trong tối đa (MIPS) thông qua độ tương tự cosine để chọn mức độ phù hợp của tài liệu. Trình truy xuất được thiết kế để lưu vào bộ đệm và cập nhật không đồng bộ từng tài liệu nhằm vượt qua thách thức tính toán khi truy xuất hàng triệu đơn hàng các tài liệu ứng viên.

Trong quá trình đào tạo trước, mô hình cần dự đoán các mã thông báo được che dấu ngẫu nhiên thông qua điểm liên quan truy xuất kiến thức $f(x, z)$, tích bên trong của việc nhúng vectơ giữa x và z (MIPS). Để triển khai bộ mã hóa dựa trên kiến thức, sự kết hợp giữa đầu vào x và tài liệu z được truy xuất từ kho văn bản được cung cấp dưới dạng một chuỗi để tính chỉnh biến áp $p(y|z, x)$ như trong Hình 17. Điều này cho phép sự chú ý chéo hoàn toàn giữa x và z cho phép dự đoán đầu ra y trong đó:

$$f(x, z) = \text{EmbedInput}(x) \exp f^T \text{Nhúngdoc}(z) \quad (72)$$

$$p(z|x) = \frac{(x, z) \exp}{\sum_z f(x, z)}$$



HÌNH 17. Huấn luyện trước không giám sát (trên cùng) và tinh chỉnh có giám sát (dưới) trong kiến trúc REALM.

$$p(y|x) = \sum_z p(y|z, x) p(z|x) \quad (73)$$

Giống như ORQA, BERT được triển khai để nhúng:

$$\text{joinBERT}(x) = [CLS] \times [SEP] \quad (74) \quad \text{joinBERT}(x_1, x_2) =$$

$$[CLS] \times x_1 [SEP] \times x_2 [SEP] \quad (75)$$

Trong quá trình đào tạo trước nhiệm vụ mô hình hóa ngôn ngữ mật nà của BERT, mỗi mật nà trong mã thông báo x cần được dự đoán là:

$$p(y|z, x) = \prod_{j=1}^{J_x} p(y_j|z, x) \quad (76)$$

$$p(y_j|z, x) = \exp \left(\sum_j^T \text{BERT_MASK}(j) \text{ tham giaBERT } x, z_{body} \right) \quad (77)$$

trong đó $\text{BERT_MASK}(j)$ đại diện cho vectơ đầu ra của Máy biến áp tương ứng với mã thông báo đeo mặt nạ j . J_x là tổng số mã thông báo $[MASK]$ trong x và w_j là từ nhúng đã học cho mã thông báo y_j . Đối với nhiệm vụ tinh chỉnh Hỏi & Đáp mở, câu trả lời y ở dạng chuỗi mã thông báo kéo dài trong tài liệu đích z . Tập khoảng $S(z, y)$ khớp với y trong z có thể được mô hình hóa như sau:

$$p(y|z, x) = \exp \left(\text{MLP} \left(\text{hSTART}(\text{các}); \text{các} \right) \text{hEND} \right) \quad (78)$$

$$\text{hBẮT ĐẦU}(\text{các}) = \text{BERT_BẮT ĐẦU}(\text{các}) \text{ tham giaBERT} \quad (79)$$

$$x, z_{body} \text{hEND}(s) = \text{BERT_END}(s) \text{ tham giaBERT } x, z_{body} \quad (80)$$

trong đó $\text{BERT_START}(\text{các})$ và $\text{BERT_END}(s)$ biểu thị các vectơ đầu ra Trans-former tương ứng với mã thông báo bắt đầu và kết thúc của span S và MLP biểu thị mạng nơ-ron chuyển tiếp nguồn cấp dữ liệu.

D. THỂ HỆ TĂNG CƯỜNG THU HỒI: RAG

RAG là sự kết hợp linh hoạt giữa 'cuốn sách đóng' tức là mô hình tham số và hiệu suất của 'cuốn sách mở' tức là các phương pháp tiếp cận mô hình truy xuất, vượt trội so với các mô hình ngôn ngữ hiện tại. Bộ nhớ tham số là một trình tự sắp xếp theo trình tự mô hình được đào tạo trước trong khi việc gửi lại Wikipedia thông qua chỉ mục vectơ dày đặc tạo thành bộ nhớ không tham số, được truy cập thông qua bộ truy xuất thần kinh được đào tạo trước. Vì RAG được xây dựng như là đỉnh cao của

cả hai, nó không yêu cầu đào tạo trước vì kiến thức có sẵn thông qua dữ liệu được đào tạo trước được truy xuất, không giống như các kiến trúc phi tham số trước đây [91]. Để đạt được bối cảnh lớn hơn trong việc tạo chuỗi đầu ra (y), RAG có mục đích chung kết hợp các đoạn văn bản z được truy xuất cho một đầu vào x nhất định, bao gồm hai thành phần

chính: (i) Retriever $p(z|x)$, được tham số hóa thông qua nội dung phù hợp nhất từ các đoạn văn bản cho truy vấn x, đoạn được truy xuất của kiến trúc RAGSequence này hoạt động như một biến tiềm ẩn được loại trừ để đạt được xác suất tối đa $p(y|x)$ trên các xấp xỉ top-K.

$$p_{\text{RAG}}(y|x) = \prod_{z \in \text{top } k(p(\cdot|x))} p_{\theta}(y_i | x, z, y_{1:i-1}) \quad (81)$$

(ii) Trình tạo $p_{\theta}(y_i | x, z, y_{1:i-1})$, được tham số hóa thông qua θ , nó tạo ra mã thông báo y_i hiện tại dựa trên biểu diễn theo ngữ cảnh của $i-1$ mã thông báo $y_{1:i-1}$ trước đó, nhập x và truy xuất pass-sage z. Mô hình RAGToken dự đoán từng mã thông báo mục tiêu dựa trên một đoạn tiềm ẩn khác nhau, đồng thời cho phép trình tạo chọn chủ đề từ nhiều tài liệu khác nhau.

$$p_{\text{RAG}}(y|x) = \prod_{z \in \text{top } k(p(\cdot|x))} p_{\theta}(y_i | x, z, y_{1:i-1}) \quad (82)$$

Mô-đun truy xuất $p(z|x)$ dựa trên Truy xuất đường đi dày đặc (DPR) trong đó d(z) biểu diễn dày đặc của tài liệu được tạo thông qua BERT và q(x) biểu diễn truy vấn được tạo thông qua BERT khác.

$$p(z|x) = \text{expd}(z), q(x) \quad (83)$$

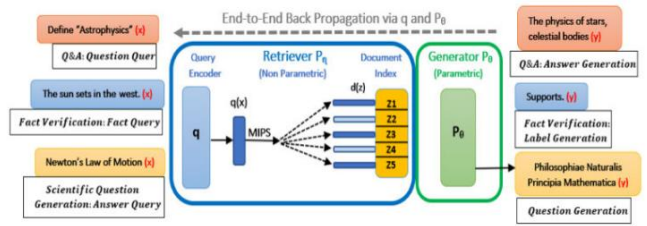
Để tính toán hiệu quả các phần tử tổng $k(p(\cdot|x))$ z có xác suất cao nhất $p(z|x)$ DPR sử dụng chỉ mục MIPS trong đó BART được sử dụng làm bộ tạo $p_{\theta}(y_i | x, z, y_{1:i-1})$. Trình truy xuất và trình tạo được đào tạo kết hợp để truy xuất tài liệu đích theo cách bán không giám sát.

E. THU HỒI HÀNH TRÌNH DENSE: DPR

DPR tăng cường khả năng truy xuất QA trong miền mở bằng cách sử dụng phương pháp mã hóa kép, không giống như CNTT chuyên sâu về tính toán.

Bộ mã hóa dày đặc $EP(\cdot)$ của nó lập chỉ mục tất cả M đoạn trong không gian (d) liên tục, có chiều thấp để có thể truy xuất hiệu quả các đoạn có liên quan hàng đầu cho một truy vấn trong thời gian chạy. Một bộ mã hóa $EQ(\cdot)$ riêng biệt được triển khai cho truy vấn và vector d chiều để ánh xạ trong thời gian chạy, truy xuất k đoạn phù hợp nhất với vector câu hỏi. Việc tính toán tích số chấm giữa truy vấn và đoạn văn sẽ xác định độ giống nhau của chúng. $\text{sim}(q, p) = \text{EQ}(q)$ một chức năng nhưng ưu việt thông qua bộ mã hóa huấn luyện liên quan đến việc tạo ra không gian vector trong đó các cặp câu hỏi, đoạn văn liên quan có khoảng cách nhỏ hơn, tức là độ tương tự lớn hơn các câu hỏi không liên quan. Giá sử dữ liệu huấn luyện với + m phiên bản $D = \{y_1, q_1, p \text{ trong } d \text{ mỗi phiên bản } i, p_{1,1}, \dots, p_{i,n} \text{ chứa một truy vấn } q_i\}$.

$$\prod_{i=1}^m \text{đoạn tích cực (có liên quan)}$$



HÌNH 18. Kiến trúc mô hình truy xuất và tham số của RAG.

$$p^+ \text{ với } n \text{ đoạn phủ định (không liên quan) } p \text{ Hàm mất mát } i, j .$$

$$L_{q_i, p^+} = \log \frac{e^{\text{sim } q_i, p^+}}{e^{\text{sim } q_i, p^+} + \sum_{j=1}^N e^{\text{sim } q_i, p_{i,j}}}$$
 (84)

VIII. MÔ HÌNH TRÌNH TỰ DÀI Vanilla

Transformers chia chuỗi đầu vào thành các khối nếu độ dài của chúng vượt quá 512 mã thông báo, dẫn đến mất ngữ cảnh khi các từ liên quan tồn tại trong các khối khác nhau. Hạn chế này dẫn đến việc thiếu thông tin theo ngữ cảnh, dẫn đến dự đoán kém hiệu quả và ảnh hưởng đến hiệu suất, đồng thời dẫn đến sự phát triển của các mô hình như vậy.

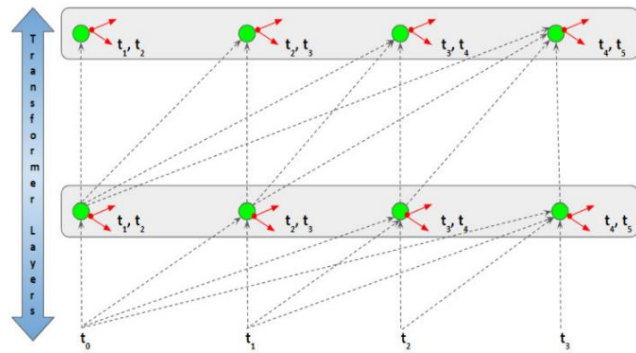
A. SỰ CHẮC CHẮN SÂU HƠN

Máy biến áp 64 lớp này [92] được chế tạo dựa trên khám phá rằng nó sở hữu mô hình cấp độ ký tự cao hơn của các chuỗi tầm xa hơn. Thông tin được truyền nhanh chóng qua các khoảng cách ngẫu nhiên so với tiến trình từng bước đơn nhất của RNN. Tuy nhiên, ba tham số tổn thất hỗ trợ sau đây đã được thêm vào Transformer thông thường giúp tăng tốc độ hội tụ và cung cấp khả năng đào tạo các mạng sâu hơn. (i) Dự đoán trên nhiều vị trí : Nói chung dự đoán nhân quả xảy ra ở

một vị trí duy nhất trong lớp cuối cùng, tuy nhiên trong trường hợp này tất cả các vị trí đều được sử dụng để dự đoán. Những tổn thất phụ này buộc mô hình phải dự đoán trong các bối cảnh nhỏ hơn và tăng tốc quá trình luyện tập mà không giảm trọng lượng. (ii) Dự đoán trên Lớp trung gian : Ngoài lớp cuối cùng, các dự đoán từ tất cả các lớp trung gian được thêm vào cho một chuỗi nhất định, khi quá trình đào tạo diễn ra, trọng số của các lớp thấp hơn sẽ giảm dần. Đối với n

các lớp, sự đóng góp của l^{th} lớp trung gian chấm dứt tồn tại sau khi hoàn thành $l/2n$ của quá trình đào tạo. (iii) Dự đoán nhiều mục tiêu : Mô hình được sửa đổi để tạo ra hai dự đoán trở lên về các ký tự trong tương lai trong đó một bộ phân loại riêng được đưa ra cho mỗi mục tiêu mới. Tổn thất mục tiêu bổ sung được cân nhắc một nửa trước khi được cộng vào tổn thất lớp tương ứng.

3 cách triển khai trên được thể hiện trong hình 19. Đối với độ dài chuỗi L, mô hình ngôn ngữ sẽ tính toán



HÌNH 19. Tăng tốc hội tụ thông qua dự đoán nhiều mã thông báo mục tiêu trên nhiều vị trí thông qua lớp trung gian.

phân phối tự hồi quy xác suất chung trên các chuỗi mã thông báo.

$$P(t_{0:L}) = P(t_0) \prod_{t=1}^L P(t_i | t_{0:i-1}) \quad (85)$$

B. MÁY BIẾN ÁP-XL

Để giảm thiểu tình trạng phân mảnh ngữ cảnh trong Transformers thông thường, XL kết hợp các phần phụ thuộc dài hơn trong đó nó tái sử dụng và lưu vào bộ nhớ đệm các trạng thái ẩn trước đó từ nơi dữ liệu được truyền đi thông qua tính năng lặp lại. Cho một tập hợp các mã thông báo $x = (x_1, x_2, \dots, x_T)$, một mô hình ngôn ngữ sẽ tính toán xác suất chung $P(x)$ một cách tự động, trong đó bối cảnh $x < t$ được mã hóa thành trạng thái ẩn có kích thước cố định.

$$P(x) = P(x_t | x < t) \quad (86)$$

Giá trị hai câu liên tiếp có độ dài L , $st = th$

trạng thái ẩn lớp thứ được tạo bởi t thứ trong đó $st = [x_{t+1}, 1]$ và $st+1 = [x_{t+1}, 1]$, chuỗi $[x_{t+1}, 1]$ trong đó n đoạn

trạng thái ẩn lớp st $L \times d$, là chiều ẩn. Sau đó

trạng thái lớp ẩn cho đoạn $st+1$ được tính như sau:

$$h_{t+1}^{n-1} = SG(h_{t+1}^{n-1}, h_{t+1}^{n-1}) \quad (87)$$

$$h_{t+1}^{n-1} = h_{t+1}^{n-1} + W_{t+1}^{n-1} Q_{t+1}^{n-1} W_{t+1}^{n-1} K_{t+1}^{n-1} W_{t+1}^{n-1} V_{t+1}^{n-1} \quad (88)$$

$$h_{t+1}^{n-1} = \text{Máy biến áp} \quad \text{Lớp qkv} \quad t+1, t+1, t+1 \quad (89)$$

trong đó $SG(\cdot)$ đại diện cho điểm dừng, $[hu \cdot hv]$ là phép nối hai chuỗi ẩn và W là tham số mô hình.

Điểm khác biệt chính so với Transformer ban đầu nằm ở việc mô hình hóa khóa k liên quan đến $t+1$

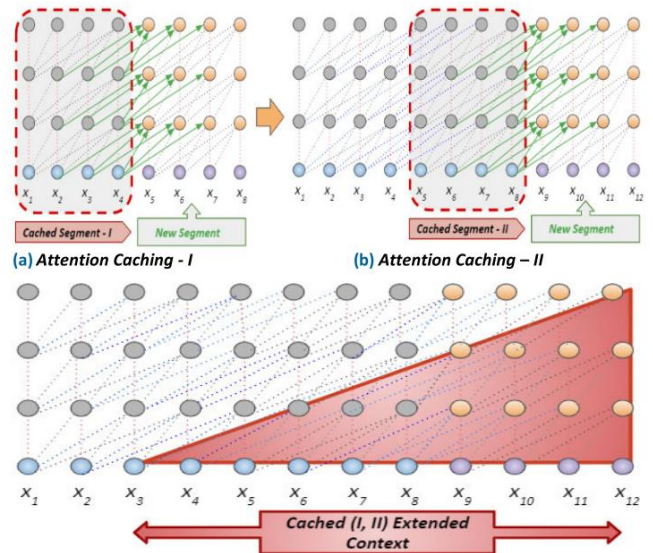
$n-1$ bối cảnh mở rộng h_{t+1}

$$v_{t+1}^{n-1} \quad \text{và giá trị} \quad v_{t+1}^{n-1}$$

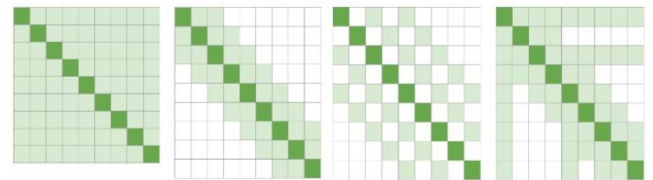
$$\text{và do đó trước} \quad h_{t+1}^{n-1} \quad \text{được lưu} \quad h_{t+1}^{n-1}$$

trữ. Điều này có thể được chứng minh từ hình 20 ở trên, trong đó khoảng chú ý trước được lưu vào bộ nhớ đệm bằng khoảng chú ý sau tạo thành cơ chế bộ nhớ đệm có cổng kéo dài.

Sự lặp lại như vậy được áp dụng cho mỗi hai phân đoạn liên tiếp để tạo ra sự lặp lại ở cấp độ phân đoạn thông qua các trạng thái ẩn. Trong máy biến áp ban đầu, điểm chú ý trong cùng phân đoạn giữa vectơ truy vấn (q_i) và vectơ khóa (k_i)



HÌNH 20. Kết hợp chụp bối cảnh kéo dài (a) và (b).



(a) Full Attention (b) Sliding Window (c) Dilated Sliding (d) Global Attention

HÌNH 21. Các cấu hình chú ý thưa thớt khác nhau của Longformer.

là:

$$M_{j,xi}^{t+1} = E_{xi}^T W_{t+1}^T Q_{t+1} W_{t+1}^T K_{t+1} E_{xi} + E_{xi}^T W_{t+1}^T Q_{t+1} W_{t+1}^T K_{t+1} E_{xi} + U_{t+1}^T W_{t+1}^T Q_{t+1} W_{t+1}^T K_{t+1} E_{xi} + U_{t+1}^T W_{t+1}^T Q_{t+1} W_{t+1}^T K_{t+1} E_{xi} \quad (90)$$

Từ góc độ mã hóa vị trí tương đối, phương trình trên được sửa lại theo cách sau

$$M_{j,xi}^{t+1} = E_{xi}^T W_{t+1}^T Q_{t+1} W_{t+1}^T K_{t+1} E_{xi} + E_{xi}^T W_{t+1}^T Q_{t+1} W_{t+1}^T K_{t+1} R_{t+1} j + u_{t+1}^T W_{t+1}^T Q_{t+1} W_{t+1}^T K_{t+1} E_{xi} + v_{t+1}^T W_{t+1}^T Q_{t+1} W_{t+1}^T K_{t+1} R_{t+1} j \quad (91)$$

C. LÂU DÀI

Kiến trúc này cung cấp tính thưa thớt cho ma trận chú ý đầy đủ trong khi xác định các cặp vị trí đầu vào tham dự lẫn nhau và triển khai ba cấu hình chú ý: (i) Cửa sổ trượt: Đối với kích thước cửa

sổ cố định w , mỗi mã thông báo sẽ tuân theo độ dài chuỗi (n) là $w/2$ ở hai bên.

Điều này dẫn đến độ phức tạp tính toán của $O(n \times w)$ tỷ lệ tuyến tính với độ dài chuỗi đầu vào và vì mục đích hiệu quả $w < n$. Một máy biến áp lớp 1 xếp chồng lên nhau cho phép khả năng tiếp nhận có kích thước $1 \times w$ trên toàn bộ đầu vào w trên tất cả các lớp. Các giá trị w khác nhau có thể được chọn cho hiệu quả hoặc hiệu suất. (ii) Cửa sổ trượt mở rộng: Để bảo toàn tính toán và nơi

mở rộng kích thước trường tiếp nhận lên $1 \times d \times wd$ các khoảng trống có kích thước thay đổi được quy cho sự giãn nở ở kích thước cửa sổ w . Hiệu suất năng cao đạt được thông qua

cho phép một số đầu không giãn nở (kích thước cửa sổ nhỏ hơn) để chú ý đến ngữ cảnh cục bộ (các lớp thấp hơn) và các đầu giãn nở còn lại (kích thước cửa sổ tăng) tham gia vào ngữ cảnh dài hơn (các lớp cao hơn).

(iii) Sự chú ý toàn cầu : Hai cách triển khai trước không có đủ tính linh hoạt để học tập chính xác theo nhiệm vụ. Do đó ''sự chú ý toàn cầu được triển khai trên một số mã thông báo đầu vào được chỉ định trước (n) trong đó mã thông báo tham dự tất cả các mã thông báo chuỗi và tất cả các mã thông báo đó tham dự vào nó. Điều này duy trì sự phức tạp chú ý cục bộ và toàn cầu đối với $O(n)$.

Độ phức tạp chú ý của nó là tổng của sự chú ý cục bộ và toàn cầu so với độ phức tạp bậc hai của RoBERTa được giải thích bằng các biểu thức toán học sau.

sự chú ý cục bộ = $(n \times w)$

trong đó n (kích thước chuỗi đầu vào), w (kích thước cửa sổ) chú ý toàn cục = $(2 \times n \times$

s) trong đó s (số lượng mã thông báo được chú ý hoàn toàn)

Kích thước chú ý của cửa sổ = $n\theta$, do đó $(n\theta = w)$

Tổng độ phức tạp chú ý = $n (n\theta + 2s) \in O (n)$

nếu $n\theta = n$

Tổng yêu cầu bộ nhớ = $n (n\theta +$

2s) \times Số lớp chuyển đổi

Sự chú ý toàn cầu cho phép xử lý tài liệu ít đoạn hơn, tuy nhiên, độ phức tạp về không-thời gian của nó sẽ lớn hơn RoBERTa, nếu độ dài chuỗi vượt quá kích thước cửa sổ.

$$O(\text{RoBERTa}) = O(n\theta) \qquad \qquad \qquad 2 \qquad < \qquad \qquad \qquad O(\text{Người cũ})$$

$$= O(n(n\theta + 2s))$$

nếu $> n\theta$

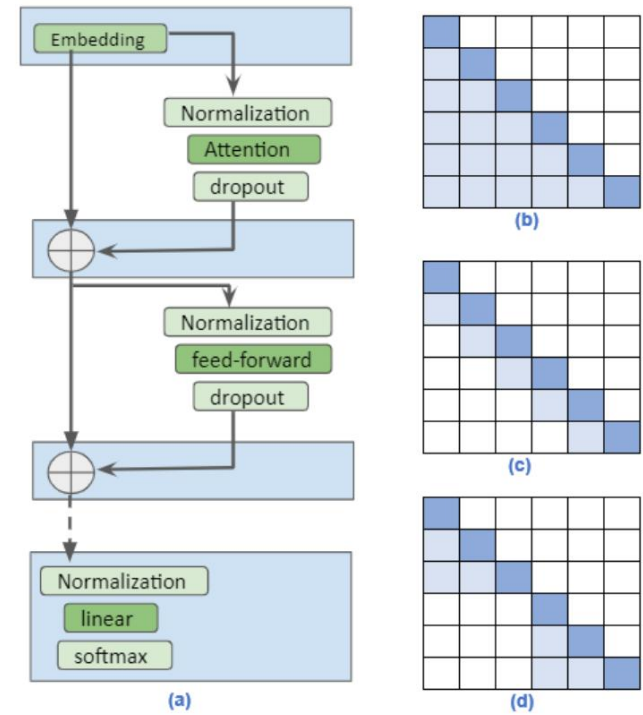
D. THI CÔNG MÁY BIẾN ÁP MỞ RỘNG: ETC

ETC là một phiên bản chuyển thể của thiết kế Longformer nhận đầu vào toàn cầu (ng) và dài (nl) trong đó ng Nó tính toán bốn biến n_1 . thể chú ý toàn cầu-cục bộ: toàn cầu đến toàn cầu (g2g), toàn cầu đến dài (g2l), dài -to-global (l2g) và long-to-long (l2l) để đạt được quá trình xử lý chuỗi dài.

Đầu vào toàn cầu và ba biến thể còn lại có sự chú ý vô hạn để bù đắp cho khoảng bán kính cố định của l2l nhằm đạt được sự cân bằng giữa hiệu suất và chi phí tính toán. Hơn nữa, nó thay thế mã hóa tuyệt đối bằng mã hóa vị trí tương đối nhằm cung cấp thông tin về các mã thông báo đầu vào liên quan đến nhau.

E. CHIM LỚN

Về mặt toán học, Big Bird chứng minh sự chú ý thưa thớt ngẫu nhiên có thể là Turing hoàn chỉnh và hoạt động giống như một Longformer được hỗ trợ với sự chú ý ngẫu nhiên. Nó được thiết kế như (i) một nhóm mã thông báo toàn cầu g tham gia vào tất cả các phần của chuỗi (ii) tồn tại tại một nhóm r khóa ngẫu nhiên mà mỗi truy vấn q tham gia (iii) một cửa sổ lân cận cục bộ w khối mà mỗi nút cục bộ tham gia . Mã thông báo toàn cầu của Big Bird được xây dựng bằng cách sử dụng phương pháp tiếp cận hai phần (i) Big Bird-ITC: Triển khai nội bộ



HÌNH 22. (a) Kiến trúc máy biến áp thưa thớt (b) Sự chú ý hoàn toàn dựa trên bộ giải mã với mật độ nhân quả (c) Độ thưa thớt cố định (d) Độ thưa thớt cố định.

Transformer Construction (ITC) trong đó có rất ít mã thông báo hiện tại được tạo ra trên toàn cầu tham gia vào chuỗi hoàn chỉnh. (ii) Big Bird-ETC: Triển khai Cấu trúc máy biến áp mở rộng (ETC), các mã thông báo toàn cầu bổ sung cần thiết g được bao gồm [CLS] liên quan đến tất cả các mã thông báo hiện có.

Quá trình chú ý dứt khoát của nó bao gồm các thuộc tính sau: truy vấn tham dự r khóa ngẫu nhiên trong đó mỗi truy vấn tham dự w/2 mã thông báo ở bên trái và bên phải vị trí của nó và có g mã thông báo toàn cầu có nguồn gốc từ mã thông báo hiện tại hoặc có thể được bổ sung khi cần .

IX. KIẾN TRÚC TÍNH TOÁN HIỆU QUẢ A. BIẾN ÁP THƯỜNG THỨC

Hiệu suất kinh tế của mô hình này là do sự tách biệt khỏi quy trình tự chú ý đầy đủ được sửa đổi qua một số bước chú ý. Kết quả đầu ra của mô hình được lấy từ hệ số của mảng đầu vào đầy đủ, tức là (\sqrt{N}) trong đó N ' Độ dài chuỗi như được biểu thị trong Hình 22. Điều này dẫn đến độ phức tạp chú ý thấp hơn $O(N \sqrt{N})$ trái ngược với Transformer $O(N^2)$ (Dài hơn N lần so với các phiên bản trước. Khả năng tự chú ý được nhân tố hóa của nó bao gồm p đầu riêng biệt trong đó đầu m xác định một (m) tập hợp con các chỉ số chú ý $A_{ij} : j \leq i$ và để tạo ra $1^{2 \times n}$). Hơn nữa, nó giải mã chuỗi 30

th

(m) —

chú ý được chú ý _ v p n dẫn đến các lựa chọn hiệu quả cho tập A. độ thưa thớt A Sự

được thực hiện theo hai chiều trong đó một đầu chú ý đến l vị trí trước đó và cái còn lại

chú ý đến từng vị trí l

vị trí, trong đó giá trị sai phân l gần với \sqrt{n} .

Điều này được thể hiện dưới dạng A $_{ij}^{(1)} = \{t, t+1, \dots, i\}$ với $t = \max(0, i - l)$

và A $_{ij}^{(1)} = \{j : (i - j) \bmod l = 0\}$. Phép biến đổi tuyến tính này

dẫn đến sự chú ý dày đặc này:

Chú ý (X) = Wp.attend(X, S) (92)

trong đó Wp là ma trận bài viết được chú ý. Tương tự, để triển khai các đầu chú ý được phân tích thành hệ số, một loại chú ý được sử dụng luân phiên trên mỗi khối dư hoặc xen kẽ hoặc siêu tham số xác định tỷ lệ.

Chú ý (X) = Wp.attend(X, A (rmod p)) (93)

trong đó r là chỉ số khối dư hiện tại và p là số lượng nhân viên được hệ số hóa. Một cách tiếp cận bằng đầu được hợp nhất thay thế kết hợp một đầu có mặt với các vị trí mục tiêu mà cả hai đầu có hệ số sẽ tham gia. Cách tiếp cận này tốn kém hơn về mặt tính toán bởi hệ số không đổi.

Chú ý (X) = Wp.attend X, P Là) (94)

m=1

Giải pháp thay thế thứ ba sử dụng sự chú ý nhiều đầu, trong đó các tích chú ý (nh) được tính toán song song và ghép nối dọc theo kích thước đối tượng.

Chú ý (X) = Wp(tham dự(X, A)) i { ' 1, ,nh) (95)

Nhiều đầu cho kết quả vượt trội trong khi đó, đối với các chuỗi dài hơn trong đó sự chú ý quyết định tính toán thì sự chú ý theo trình tự được ưu tiên hơn.

B. NHÀ CẢI CÁCH

Reformer giảm độ phức tạp của sự chú ý của Transformer xuống O(L log L) thông qua hàm băm nhảy cảm cục bộ (LSH). Điều này gán mỗi vectơ x cho một hàm băm h(x), trong đó các vectơ lân cận thu được cùng một hàm băm trong các nhóm băm có kích thước tương tự với xác suất cao còn các vectơ ở xa thì không. Phương trình chú ý LSH đã sửa đổi:

ôi = exp qi .kj z(i, Pi) vj (96)

j 'Pi

trong đó Pi = {j : i ≥ j} th

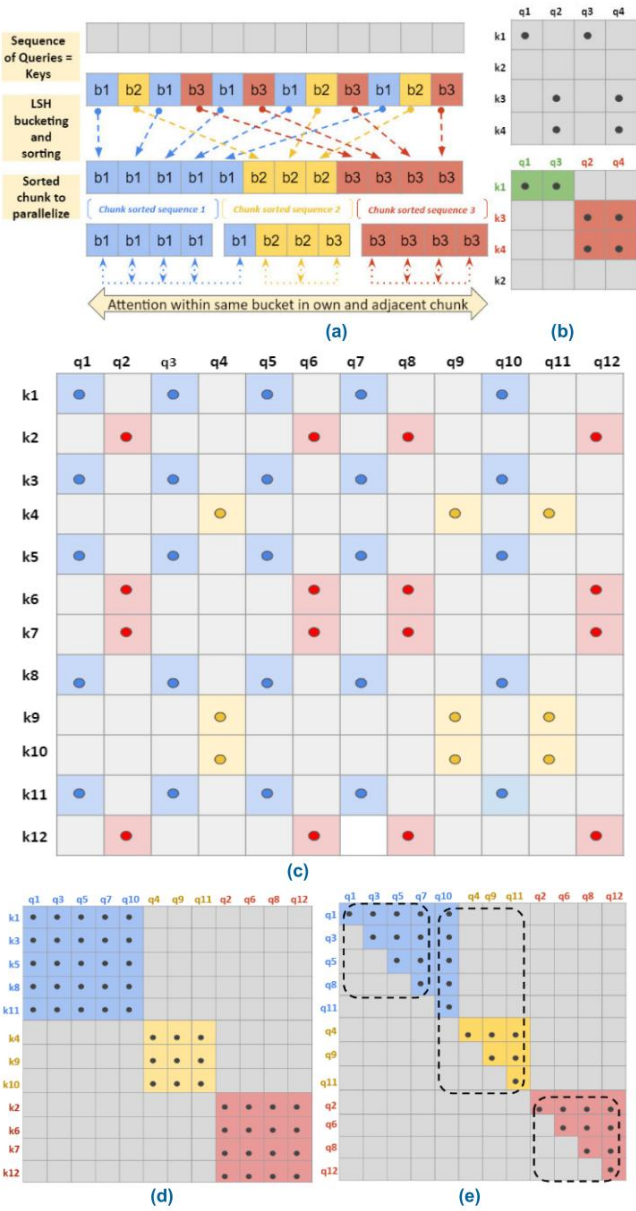
Pi thuộc tập hợp mà tôi truy vấn vị trí tham dự, z là hàm phân vùng chứa một phạm vi các khóa lân cận mà truy vấn tham gia. Với mục đích phân khối, sự chú ý được thực hiện trên P~ = {0, 1 } Pi trong đó Pi là tập con của P~ i và các phần tử không có trong Pi bị che.

ôi = exp(qi .kj m(j, Pi) z(i, Pi))vj (97)

j 'P~ i

trong đó m(j, Pi) = ∞, nếu j / Pi 0 ngược lại

Bộ giải mã thực hiện việc che dấu để ngăn truy cập vào các vị trí truy vấn trong tương lai. Các mục mục tiêu Pi đã đặt chỉ có thể được tham dự bằng một truy vấn tại nhóm băm thứ i . Để tiếp tục giảm xác suất của các mặt vị trí, bằng cách cho phép sự chú ý trong một hàng tương tự



HÌNH 23. (a) Sự hình thành nhóm của các vectơ chú ý tương tự (b) Việc phân nhóm đơn giản của một cặp Khóa truy vấn (c) Phân phối chuỗi khóa truy vấn dựa trên (a) trước Nhóm (d), (e) Phân nhóm và phân chia (c).

rơi vào các nhóm khác nhau, một số phép băm song song (nround) được thực hiện với các hàm băm riêng biệt {h (1) , h (2) , ...}

Pi = exp(qi .kj m(j, Pi) z(i, Pi))vj (98)

r=1

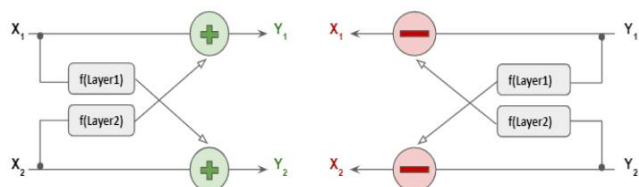
Sự chú ý được thực hiện trên các khối truy vấn khóa được sắp xếp và các khóa theo lô:

P~ (r) (x) = j : 1 ≤ j ≤ S(r) (99)

Từ (96) và (97) ta viết được:

ôi = exp(qi .kj m(j, Pi) z(i, Pi))vj (100)

j 'P~ i



HÌNH 24. Rev-Nets bỏ qua bộ nhớ trung gian thông qua quá trình tính toán trước.

$$= \frac{1}{\sum_{j=1}^n \exp(z(i, P_{j-1}(r)) - z(i, P_j(r)))} \exp(z(i, P_{j-1}(r)) - z(i, P_j(r))) \quad (101)$$

$$= \frac{1}{\sum_{j=1}^n \exp(z(i, P_{j-1}(r)) - z(i, P_j(r)))} \exp(z(i, P_{j-1}(r)) - z(i, P_j(r))) \quad (102)$$

Ví dụ sau đây trong hình 23 thể hiện một cách toàn diện các cơ chế hoạt động khác nhau của Reformer.

Mạng dư thừa có thể đảo ngược [93] là một động lực khác đằng sau việc tiêu thụ bộ nhớ tiết kiệm của Reformer trong đó các giá trị kích hoạt được xây dựng lại nhanh chóng trong quá trình truyền ngược, ngoại trừ các yêu cầu lưu kích hoạt trong bộ nhớ. Từ hình 24 bên dưới, khối đảo ngược của mỗi lớp được tính toán lại từ các lần kích hoạt của lớp tiếp theo như sau:

$$Y1 = X1 + f(X2, Layer2), Y2 = X2 + f(X1, Layer1) \quad (104)$$

$$X1 = Y1 - f(Y2, Layer2), X2 = Y2 - f(Y1, Layer1) \quad (105)$$

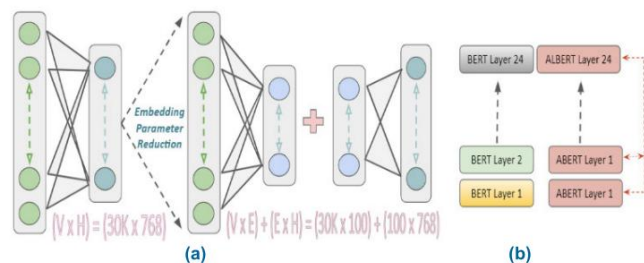
C. MỘT LITE BERT: ALBERT

ALBERT, trong một mô hình duy nhất, tích hợp các kỹ thuật giảm hai tham số sau đây dẫn đến chỉ có 12 triệu tham số như trong hình 25. Điều này giúp giảm gần 90% tham số so với BERT-base trong khi vẫn duy trì hiệu suất chuẩn cạnh tranh.

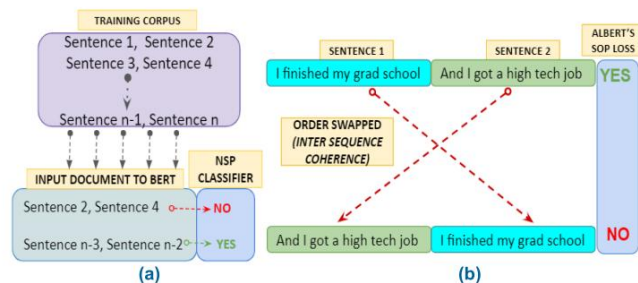
- (i) Tham số hóa nhúng nhân tố: Để có kết quả tối ưu, các tác vụ NLP yêu cầu vốn từ vựng V lớn, trong đó (kích thước nhúng) $E \equiv H$ (lớp ẩn) và kích thước ma trận nhúng $V \times E$ có thể mở rộng lên tới hàng tỷ tham số. ALBERT phân tích không gian nhúng E thành hai ma trận nhỏ hơn trong đó các tham số nhúng được giảm từ $O(V \times H)$ xuống $O(V \times E + E \times H)$.

(ii) Chia sẻ tham số lớp chéo: ALBERT được xây dựng để chia sẻ tham số chú ý giữa các lớp thông qua mạng chuyển tiếp nguồn cấp dữ liệu (FFN). Do đó, quá trình chuyển đổi giữa các lớp của nó mượt mà hơn đáng kể vì kết quả cho thấy tác động ổn định của việc chia sẻ trọng lượng đối với các tham số mạng.

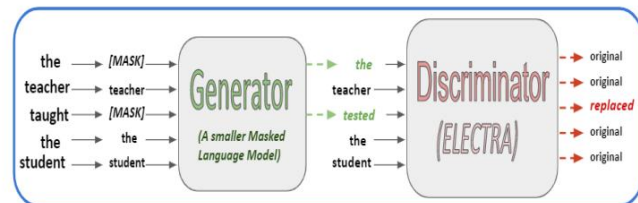
Giống như NSP của BERT, việc mất khả năng dự đoán thứ tự câu (SOP) của ALBERT kết hợp việc học theo hai hướng từ hai phân đoạn văn bản tích cực liên tiếp cũng bao gồm các mẫu phủ định tương ứng với các thứ tự đảo ngược như được minh họa trong hình 26. Điều này ảnh hưởng đến mô hình để tìm hiểu theo ngữ cảnh những khác biệt chi tiết hơn trong bất kỳ diễn ngôn nào mang lại hiệu quả mạch lạc vượt trội. Mục tiêu MLM của nó triển khai mặt nạ n -gram bao gồm các chuỗi tối đa 3 ký tự,



HÌNH 25. (a) Mô hình nhỏ hơn thông qua giảm kích thước nhúng (b) học tập hiệu quả thông qua chia sẻ các tham số chú ý.



HÌNH 26. (a) Học NSP của BERT thông qua thứ tự cặp không đảo ngược đơn giản (b) Học tính cảm kép SOP của ALBERT thông qua đảo ngược thứ tự câu.



HÌNH 27. Phát hiện mã thông báo được thay thế thông qua đào tạo kết hợp của mô hình.

như "Bóng đá World Cup" hay "Xử lý ngôn ngữ tự nhiên".

$$p(n) = \frac{1/n}{\sum_{k=1}^n 1/k} \quad (106)$$

D. ĐIỂN

Ưu điểm nằm ở việc học theo ngữ cảnh thông qua khả năng phân biệt hiệu quả, nơi nó học hỏi từ tất cả mọi người. mã thông báo đầu vào không giống như BERT học từ tập hợp con bị che giấu chỉ 15%. ELECTRA triển khai "phát hiện mã thông báo được thay thế", như minh họa trong hình 27, trong đó sự lây nhiễm xảy ra bằng cách thay thế một số mã thông báo ngẫu nhiên bằng các thay thế có ý nghĩa xác suất thông qua Trình tạo (G), một 'mô hình ngôn ngữ đeo mặt nạ' nhỏ.

Đồng thời, thông qua phân loại nhị phân, Bộ phân biệt đối xử (D) mô hình lớn hơn sẽ được đào tạo trước để dự đoán xem mỗi mã thông báo có được khôi phục chính xác thông qua trình tạo hay không.

$$\text{LMLM}(x, \theta_G) = E_{x \sim p(x)} \log p_G(x | x_{\text{che mặt}}) \quad (107)$$

$$\text{LDisc}(x, \theta_D) = E_{x \sim p(x)} \log p_D(x | x_{\text{đúng}}, x_{\text{đúng}}) \quad (108)$$

Hai mạng dựa trên bộ mã hóa (G, D) chuyển đổi chuỗi mã thông báo đầu vào $x = [x_1, \dots, x_n]$ thành biểu diễn vectơ theo ngữ cảnh $h_x = [h_1, \dots, h_n]$. Thông qua Softmax, G mang lại khả năng tạo mã thông báo vị trí t x_t , trong đó $x_t = [\text{MASK}]$.

$$p_G(x_t | x) = \frac{\sum_{x'} \text{điểm kinh nghiệm}(x_t) h_G(x) \cdot \sum_{x'} \text{kinh nghiệm}(x) h_G(x) \cdot t}{\sum_{x'} \text{kinh nghiệm}(x) h_G(x) \cdot t} \quad (109)$$

Tổn hao tổng hợp trên một tập lớn χ được giảm thiểu

BẰNG:

$$\min_{\theta_G, \theta_D} \text{LMLM}(x, \theta_G) + \lambda \text{Disc}(x, \theta_D) \quad (110)$$

E. LINFORMER

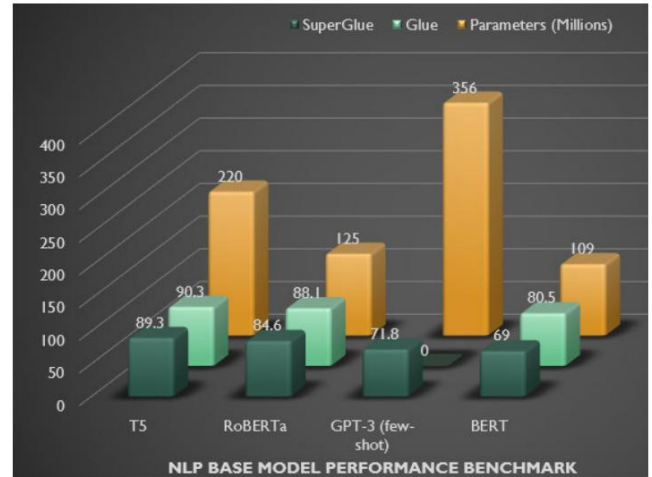
Nó chứng minh [94] rằng trọng số chú ý bị chi phối bởi một số mục nhập chính, do đó độ dài chuỗi được chiếu xuống ma trận đầu ra mục tiêu thông qua khả năng tự chú ý cấp thấp để đạt được độ phức tạp về không gian và thời gian tuyến tính $O(L)$. Trong quá trình tính toán khóa và giá trị, hai ma trận chiếu tuyến tính được thêm $n \times k$ E_i , F_i R trong đó $(n \times d)$ khóa chiếu, các lớp giá trị KWK và VWV được chiếu vào $(k \times d)$ khóa chiếu, các lớp giá trị, sau đó là kết quả $(n \times k)$ ánh xạ ngữ cảnh các chiếu được tính toán bằng cách sử dụng chú ý tích số chằm được chia tỷ lệ.

$$\text{đầu} = \text{softmax} \left(\frac{QWQ - E_i KWK \sqrt{d_k}}{d_k} \right) \cdot (F_i VWV) \quad (111)$$

Nếu $k \ll n$ thì sẽ giảm đáng kể mức tiêu thụ bộ nhớ và dung lượng. Để tối ưu hóa hiệu quả hơn nữa, việc chia sẻ tham số giữa các phép chiếu được thực hiện ở ba cấp độ: (i) Chia sẻ theo chiều dọc: đối với mỗi lớp, hai ma trận chiếu E và F được chia sẻ trong đó $E_i = E$, $F_i = F$ qua tất cả các đầu i . (ii) Chia sẻ khóa-giá trị: bao gồm (i) khóa, các phép chiếu giá trị được chia sẻ trong đó ma trận chiếu đơn của mỗi lớp $E = E_i = F_i$ được tạo cho mỗi ma trận chiếu khóa-giá trị cho tất cả các đầu i (iii) Chia sẻ theo lớp: một ma trận chiếu đơn E được triển khai cho tất cả các lớp, đầu, khóa và giá trị. Đối với Máy biến áp 12 lớp, 12 đầu, (i), (ii), (iii) sẽ kết hợp lần lượt 24, 12, 1 ma trận chiếu tuyến tính riêng biệt.

F. NGƯỜI BIỂU TƯỢNG

Chú ý tiêu chuẩn $(Q \times L \times d) \cdot K$ độ phức tạp $\frac{1}{d} \times L \cdot V \times d$ dẫn đến thời gian bậc hai của $O(L^2 d)$, thích hợp hơn là imple- $V \times L \times d$) đề cập đến $Q \times L \times d \cdot (K \times L \times \frac{1}{d} \times L)$ dẫn đến $O(d^2 L)$ trong đó d . Tuy nhiên, việc phân tách sự chú ý của khóa truy vấn sản phẩm ở dạng nguyên sơ là không thể sau khi thực hiện hàm phi tuyến tính softmax. Tuy nhiên, có thể phân tách sự chú ý trước softmax thông qua việc tính gần đúng các truy vấn và khóa được xếp hạng thấp hơn cho phép $QKT = \text{softmax}(\frac{1}{d} \times L \times d)$ lớn hơn hiệu quả, cụ thể là $QK = \exp(QKT)$. Điều này đạt được thông qua phép tính gần đúng kernel hàm số $K(x, y) = \exp(-\frac{1}{2} \|x - y\|^2)$, tích số chằm của a



HÌNH 28. Biểu diễn đồ họa của hiệu suất mô hình ngôn ngữ.

bản đồ đặc trưng chiếu cao. Ngược lại với thủ thuật kernel-nel trong đó số chiều được tăng lên, Per-former [95] phân rã ma trận chú ý $A(i, j) = K(q_i, k_j) = \exp(q_i \cdot k_j \text{ map})$ đến một tính năng có chiều thấp hơn

X. PHÂN LOẠI MÔ HÌNH CỦA LM Các mô hình ngôn ngữ dựa

trên máy biến áp (LM) có thể được phân loại thành 3 loại [96] từ góc độ mô hình hóa:

- Tự hồi quy:** Đây là các mô hình chuyển tiếp được đào tạo trước để dự đoán các mã thông báo trong tương lai từ mã thông báo của anh ấy. Ở đây đầu ra y_t phụ thuộc vào đầu vào tại thời điểm x_t và đầu vào bước thời gian trước đó $x_{<t}$. Đây chủ yếu là các Transformers dựa trên bộ giải mã, kết hợp tính năng che dấu nhân quả, trong đó người đứng đầu sự chú ý bị ngăn không cho tham dự vào các mã thông báo trong tương lai. Các mô hình như vậy thường được tinh chỉnh cho mục đích tạo văn bản và triển khai phương pháp học không cần chụp trong chuỗi GPT.
- Tự động mã hóa:** Các mô hình dựa trên Bộ mã hóa này có toàn quyền truy cập vào mảng đầu vào mà không có bất kỳ mặt nạ nào. Để tìm hiểu, họ được đào tạo trước bằng cách kết hợp các sơ đồ che dấu mã thông báo đầu vào và sau đó tinh chỉnh để tái tạo các mã thông báo được che dấu làm đầu ra. Các mô hình này (BERT) thường thích hợp cho các nhiệm vụ phân loại chuỗi hoặc mã thông báo.
- Từ trình tự đến trình tự:** Các mô hình tổng quát dựa trên Bộ mã hóa-Bộ giải mã này tạo ra dữ liệu sau khi học từ một tập dữ liệu lớn. Không giống như phân phối phân biệt $P(Y|X)$, chúng lập mô hình phân phối chung $P(X, Y)$ của đầu vào X và Y đích trong đó đầu vào có thể bị hỏng trên một số sơ đồ. Mặt nạ nhân quả dựa trên bộ giải mã được triển khai để tối đa hóa việc học cho việc tạo mục tiêu tiếp theo. Các mô hình như BART và T5 hoạt động tốt nhất trong các nhiệm vụ NMT, tóm tắt hoặc QA.

Tổng quan toàn diện về phân loại mô hình nêu trên được trình bày trong hình 29.

MODEL	DESCRIPTION	TASKS	LANGUAGAE MODELING TYPE
GPT-I, II, III	<ul style="list-style-type: none">Unsupervised pre-training on large datasetsAutoregressive Language Modeling and Causal Masking	Q&A, NMT, Reading Comprehension, Text Summarization, Common Sense Reasoning, Zero-Shot	Autoregressive DECODER based Transformer
XLNET	<ul style="list-style-type: none">Greater Contextual Learning via Factorized Ordering on Input's Sequence LengthBidirectional Contextual Language Modeling	Reading Comprehension, Natural Language Inference, Sentiment Analysis, Q&A	Autoregressive DECODER based Transformer
REFORMER	<ul style="list-style-type: none">Attention via Local Sensitive Hashing reducing memory footprintIncorporates re-computation of weights and activations bypassing their respective storage via reversible residual networks	Reduced Attention Complexity enabling lengthy sequence processing on pragmatic memory requirements	Autoregressive DECODER based Transformer
LONGFORMER	<ul style="list-style-type: none">Sparsity in Attention Matrices for lengthy sequence speedup and efficient computationLocalized Attention for nearby tokens and all access Globalized Attention for few preselected tokens to enhance receptivity	Co-reference Resolution, Q&A, Document Classification.	Autoregressive DECODER based Transformer
BERT	<ul style="list-style-type: none">Deep Bidirectional ContextualizationMasked Language Modeling (MLM) for continual learning	Sentence Classification, Q&A, Natural Language Inference,	Auto-encoded ENCODER based Transformer
RoBERTa	<ul style="list-style-type: none">Diverse learning via Dynamic Masking, where tokens are masked differently for each epochLarger pre-training Batch-Size	Sentiment Analysis, Q&A, Natural Language Inference	Auto-encoded ENCODER based Transformer
DistilBERT	<ul style="list-style-type: none">Produces similar target probability distribution as its larger teacher model, BERTGenerates cosine similarity between student and teacher model's hidden states	Semantic Textual Similarity, Semantic Relevance, Q&A, Textual Entailment	Auto-encoded ENCODER based Transformer
ALBERT	<ul style="list-style-type: none">Smaller and efficient model via Embedding Parameter Reduction, i.e., Factorized ParametrizationLayers split into groups via Cross-Layer Parameter Sharing reducing memory footprint	Reading Comprehension, Semantic Textual Similarity, Q&A, Language Inference	Auto-encoded ENCODER based Transformer
ELECTRA	<ul style="list-style-type: none">Predict if the re-generated corrupted token is original or replaced via pre-training GeneratorEffective and low-cost discriminative learning via replaced token detection	Provides competitive performances on Sentiment Analysis, Natural Language Inference tasks at 25% compute	Auto-encoded ENCODER based Transformer
BART/mBART	<ul style="list-style-type: none">Superior sequence generation quality via greater noising variationsFlexible denoising autoencoder acts as a language model in severest noising scheme	Supervised and Unsupervised multi-lingual Machine Translation, Q&A, Semantic Equivalence	Generative Sequence to Sequence based Transformer
T5/mT5	<ul style="list-style-type: none">Positional Encodings learned at each layer for greater semantical performanceTransforms all tasks in a text-to-text format to incorporate most NLP task varieties.	More Diverse and Challenging coreference, entailment, Q&A tasks via SuperGLUE benchmark	Generative Sequence to Sequence based Transformer

HÌNH 29. Biểu diễn dạng bảng của phân loại mô hình ngôn ngữ.

XI. SO SÁNH MÔ HÌNH NGÔN NGỮ HIỆU SUẤT

Hiệu suất định lượng của một số mô hình NLP chính được thể hiện trong hình 28 dựa trên điểm chuẩn của Keo và SuperGlue. Các điểm chuẩn này chứa nhiều bộ dữ liệu khác nhau để đánh giá mô hình trên một số nhiệm vụ NLP.

Với số lượng tham số có thể huấn luyện cao nhất, GPT-3 là mô hình lớn nhất trong cuộc so sánh này. Vì GPT-3 là

mô hình mới nhất ở đây, nó không tham gia vào tiêu chuẩn Keo cũ hơn.

Từ góc độ định tính, T5 trong cùng một mô hình sử dụng cùng một hàm mất mát và các siêu tham số trải rộng trên nhiều nhiệm vụ khác nhau, dẫn đến môi trường học tập đa nhiệm. Nó hoạt động tốt nhất vì mô hình tạo văn bản thành văn bản (NLG) có thể mở rộng này kết hợp với việc khử nhiễu

mục tiêu trong quá trình đào tạo với lượng lớn dữ liệu chưa được dán nhãn. Điều này dẫn đến khả năng học tập vượt trội và hiệu suất tổng quát cao hơn so với các mô hình NLU như RoBERTa, được tinh chỉnh cho các nhiệm vụ tiếp theo riêng lẻ sau khi đào tạo trước.

Động cơ chính của một số vòng tinh chỉnh trong các mô hình NLU là đạt được hiệu suất mạnh mẽ trên nhiều nhiệm vụ. Nhược điểm chính là yêu cầu một tập dữ liệu mới và thường lớn cho mỗi tác vụ. Điều này làm tăng khả năng khái quát hóa kém ngoài phân phối dẫn đến so sánh không công bằng với khả năng ở cấp độ con người. GPT-3 không hoạt động dựa trên tinh chỉnh vì trọng tâm của nó là cung cấp khả năng thực thi bất khả tri về nhiệm vụ. Tuy nhiên, có phạm vi tinh chỉnh tối thiểu trong GPT-3 dẫn đến việc học một hoặc vài lần.

Ý tưởng là thực hiện cập nhật độ dốc bằng 0 hoặc tối thiểu sau khi đào tạo trước một mô hình khổng lồ trên một tập dữ liệu lớn.

Mặc dù GPT-3 không được xếp hạng cao với điểm chuẩn SuperGlue, nhưng điều quan trọng là mô hình tổng quát này có tốc độ học nhanh nhất bất kỳ tác vụ nào tại thời điểm suy luận. Nó phù hợp với hiệu suất với các mô hình được tinh chỉnh SOTA trên một số tác vụ NLP ở cài đặt 0, một và vài lần chạy. Nó cũng tạo ra các mẫu chất lượng cao và mang lại hiệu suất ổn định về chất lượng cho các nhiệm vụ được xác định nhanh chóng.

XII. KẾT LUẬN VÀ ĐỊNH HƯỚNG TƯƠNG LAI

Chúng tôi cung cấp bản tóm

tắt toàn diện và chi tiết về các mô hình ngôn ngữ chính đã dẫn đến SOTA hiện tại trong hiệu suất NLP. Kể từ khi ra mắt cơ chế Chú ý và kiến trúc Transformer, NLP đã phát triển theo cấp số nhân. Chúng tôi đã trình bày bản đồ tư duy cấp cao về phân loại mô hình thông qua phân loại. Các phân loại này chủ yếu dựa trên kiến trúc phái sinh Transformer, được xây dựng cho các nhiệm vụ chuyên biệt như Tìm hiểu và tạo ngôn ngữ, Giám kích thước mô hình thông qua chum cắt, Lượng tử hóa và cắt tia, Truy xuất thông tin, Lập mô hình chuỗi dài và các công nghệ Giám mô hình tổng quát khác. niques. Các mô hình ngôn ngữ gần đây chủ yếu được thúc đẩy bằng cách đạt được hiệu suất NLP cao hơn đòi hỏi tài nguyên máy tính khổng lồ. Vì vậy, việc mở rộng quy mô mô hình đã trở thành con đường tự nhiên trong ngành. Tỷ lệ mở rộng theo cấp số nhân này cùng với độ phức tạp chú ý cao hơn khiến các mô hình này không thể truy cập được ở quy mô toàn cầu. Sau đó, những nỗ lực đáng kể đã được thực hiện để thiết kế các mô hình có kích thước hợp lý và tính toán chú ý hiệu quả nhằm tăng tốc độ hội tụ mô hình, dẫn đến độ trễ trong mô hình thấp hơn.

Việc kết hợp phương pháp Hỗn hợp Chuyên gia (MoE) [97] là một cách hiệu quả để các mô hình lớn đạt được hiệu quả tính toán, vì chỉ một tập hợp con của mạng lưới thần kinh được kích hoạt cho mỗi đầu vào. Do đó, điều này dẫn đến tình trạng thừa thớt và mặc dù đào tạo về thừa thớt là một lĩnh vực nghiên cứu tích cực, các GPU hiện tại phù hợp hơn cho các phép tính ma trận dày đặc. Trong khi các mô hình của MoE đã chứng tỏ được triển vọng trong việc đào tạo các ma trận thừa thớt, thì chi phí liên lạc và độ phức tạp của chúng lại cản trở việc triển khai trên quy mô rộng. Hơn nữa, các mô hình lớn hơn có xu hướng ghi nhớ dữ liệu huấn luyện dẫn đến việc trang bị quá mức và giảm khả năng học tập [98]. Để khắc phục điều này, các mô hình chỉ

được đào tạo cho một kỷ nguyên duy nhất trên các phiên bản loại bỏ trùng lặp trên các tập dữ liệu khổng lồ, do đó thể hiện mức độ trang bị quá mức ở mức tối thiểu. Do đó, thiết kế của MoE kết hợp với mô hình đào tạo mạnh mẽ trong tương lai có thể dẫn đến các mô hình hiệu quả và có khả năng mở rộng cao. Những mô hình này sẽ có khả năng hiểu ngôn ngữ vượt trội vì việc ghi nhớ dữ liệu sẽ được giảm thiểu. Cách tiếp cận hiện tại trong các mô hình SOTA dựa vào việc học có giám sát trên các bộ dữ liệu khổng lồ. Một lĩnh vực đầy hứa hẹn về những cải tiến trong tương lai của NLP sẽ là kết hợp học tập tăng cường trong Dịch máy, tóm tắt văn bản và các nhiệm vụ Hỏi & Đáp.

NGƯỜI GIỚI THIỆU

[1] Y. LeCun, Y. Bengio và G. Hinton, "Học sâu," Nature, tập. 521, trang 436-444, tháng 5 năm 2015.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, AN Gomez, Ł. Kaiser và I. Polosukhin, "Tất cả những gì bạn cần là sự chú ý" trong Proc. NeurIPS, 2017, trang 1-11.

[3] A. Radford, K. Narasimhan, T. Salimans và I. Sutskever, " Cải thiện khả năng hiểu ngôn ngữ bằng cách học không giám sát, " OpenAI Blog, San Francisco, CA, USA, Tech. Dân biểu, 2018.

[4] J. Devlin, M.-W. Chang, K. Lee và K. Toutanova, " BERT: Đào tạo trước các máy biến áp hai chiều sâu để hiểu ngôn ngữ," trong Proc. NAACL-HLT, 2019, trang 4171-4186.

[5] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy và S. Bowman, "GLUE: Nền tảng phân tích và điểm chuẩn đa tác vụ để hiểu ngôn ngữ tự nhiên," trong Proc. BlackboxNLP@EMNLP, 2018, trang 353-355.

[6] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy và SR Bowman, " SuperGLUE: Một tiêu chuẩn chính xác hơn cho các hệ thống hiểu ngôn ngữ có mục đích chung, " trong Proc. NeurIPS, 2019, trang 1-29.

[7] S. Bianco, R. Cadene, L. Celona và P. Napoletano, " Phân tích điểm chuẩn của các kiến trúc mạng thần kinh sâu đại diện, " IEEE Access, tập. 6, trang 64270-64277, 2018.

[8] P. Rajpurkar, J. Zhang, K. Lopyrev và P. Liang, "SQuAD: Hơn 100.000 câu hỏi để máy hiểu văn bản," 2016, arXiv:1606.05250. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/1606.05250>

[9] P. Rajpurkar, R. Jia và P. Liang, "Biết những gì bạn không biết: Những câu hỏi không thể trả lời cho SQuAD," 2018, arXiv:1806.03822. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1806.03822>

[10] A. Warstadt, A. Singh và SR Bowman, "Các phán đoán về khả năng chấp nhận của mạng lưới thần kinh," Trans. PGS.TS. Máy tính. Ngôn ngữ học, tập. 7, trang 625-641, tháng 11 năm 2019.

[11] RT McCoy, J. Min và T. Linzen, "BERT của một chiếc lông vũ không khái quát hóa cùng nhau: Sự biến đổi lớn về mức độ khái quát hóa giữa các mô hình có hiệu suất của tập thử nghiệm tương tự," 2020, arXiv:1911.02969. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1911.02969>

[12] H. Elsahar và M. Gallé, "Chú thích hay không? Dự đoán hiệu suất giảm khi chuyển đổi tên miền" trong Proc. EMNLP/IJCNLP, 2019, trang 2163-2173.

[13] S. Ruder và B. Plank, "Học cách chọn dữ liệu để học chuyển giao bằng tối ưu hóa Bayesian" trong Proc. EMNLP, 2017, trang 372-382.

[14] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov và QV Le, " XLNet: Đào tạo trước về quá trình hiểu ngôn ngữ theo phương pháp tự hồi quy tổng quát" trong Proc. NeurIPS, 2019, trang 1-18.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer và V. Stoyanov, "RoBERTa: A mạnh mẽ phương pháp đào tạo trước BERT được tối ưu hóa" 2019, arXiv:1907.11692. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1907.11692>

[16] B. McCann, NS Keskar, C. Xiong và R. Socher, " Mười món phối hợp ngôn ngữ tự nhiên: Học đa nhiệm như trả lời câu hỏi," 2018, arXiv:1806.08730. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1806.08730> [17] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li và PJ Liu, "Khám phá các giới hạn của việc học chuyển tiếp bằng một công cụ chuyển đổi văn bản thành văn bản thống nhất" J. Mach. Học hội. Đồ phân giải, tập. 21, trang 140:1-140:67, 2020.

[18] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov và L. Zettlemoyer, " BART: Khử nhiễu đào tạo trước theo trình tự cho tạo, dịch và hiểu ngôn ngữ tự nhiên" 2020, arXiv:1910.13461. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1910.13461>

- [19] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis và L. Zettlemoyer, "Đào tạo trước khử nhiễu đa ngôn ngữ cho dịch máy thần kinh," Dịch. PGS.TS. Máy tính. Ngôn ngữ học, tập. 8, trang 726-742, tháng 12 năm 2020.
- [20] C. Rosset, "Turing-NLG: Mô hình ngôn ngữ 17 tỷ tham số của Microsoft," Microsoft Blog, Redmond, WA, USA, Tech. Dân biểu, 2019.
- [21] Z. Xie, G. Genthial, S. Xie, A. Ng và D. Jurafsky, "Tiếng ồn và khử nhiễu ngôn ngữ tự nhiên: Dịch ngược đa dạng để sửa ngữ pháp," trong Proc. NAACL-HLT, 2018, trang 619-628.
- [22] T. Brown và cộng sự, "Các mô hình ngôn ngữ là những người học ít lần" 2020, arXiv:2005.14165. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/2005.14165> [23]
- D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer và Z. Chen, "GShard: Mở rộng quy mô các mô hình khổng lồ với tính toán có điều kiện và phân đoạn tự động," 2020, arXiv:2006.16668. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/2006.16668>
- [24] E. Strubell, A. Ganesh và A. McCallum, "Những cân nhắc về năng lượng và chính sách cho việc học sâu trong NLP," 2019, arXiv:1906.02243. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1906.02243>
- [25] GE Hinton, O. Vinyals và J. Dean, "Chất lọc kiến thức trong mạng trung tính," 2015, arXiv:1503.02531. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/1503.02531> [26] V. Sanh, L. Debut, J. Chaumond và T. Wolf, "DistilBERT, phiên bản chưng cất của BERT: Nhỏ hơn, nhanh hơn, rẻ hơn và nhẹ hơn," 2019, arXiv:1910.01108. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/1910.01108> [27] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang và Q. Liu, "TinyBERT: Chất lọc BERT để hiểu ngôn ngữ tự nhiên" 2020, arXiv:1909.10351. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1909.10351> [28] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang và D. Zhou, "MobileBERT: Một nhiệm vụ bất khả tri nhỏ gọn BERT dành cho các thiết bị có tài nguyên hạn chế" trong Proc. ACL, 2020, trang 1-13.
- [29] S. Han, H. Mao và W. Dally, "Nén sâu: Nén mạng lưới thần kinh sâu bằng cách cắt tia, lượng tử hóa được đào tạo và mã hóa Huffman," trong Proc. Máy tính. Vis. Nhận dạng mẫu, 2016, trang 1-14.
- [30] K. Lee, M.-W. Chang và K. Toutanova, "Truy xuất tiềm ẩn để trả lời câu hỏi miễn mở được giám sát yếu," 2019, arXiv:1906.00300. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1906.00300> [31] K. Guu, K. Lee, Z. Tung, P. Pasupat và M.-W. Chang, "REALM: Đào tạo trước về mô hình ngôn ngữ tăng cường truy xuất," 2020, arXiv:2002.08909. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/2002.08909> [32] P. Lewis, E. Perez, A. Piktus, V. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, M.-T. Yih, T. Rocktäschel, S. Riedel và D. Kiela, "Hệ thể tăng cường truy xuất cho các nhiệm vụ NLP chuyên sâu về kiến thức," 2020, arXiv:2005.11401. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/2005.11401> [33] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen và W.-T. Yih, "Truy xuất đoạn văn dày đặc để trả lời câu hỏi trong miền mở," 2020, arXiv:2004.04906. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/2004.04906> [34] Z. Dai, Z. Yang, Y. Yang, J. Carboneil, QV Le và R. Salakhutdinov, "Transformer-XL: Các mô hình ngôn ngữ chú ý vượt xa văn bản có độ dài cố định" 2019, arXiv:1901.02860. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1901.02860>
- [35] I. Beltagy, ME Peters và A. Cohan, "Longformer: Máy biến áp tài liệu dài" 2020, arXiv:2004.05150. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/2004.05150> [36] J. Ainslie, S. Ontañón, C. Alberti, V. Cvicek, Z. Fisher, P. Pham, A. Ravula, S. Sanghai, Q. Wang và L. Yang, "ETC: Mã hóa các đầu vào dài và có cấu trúc trong máy biến áp" trong Proc. EMNLP, 2020, trang 268-284.
- [37] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontañón, P. Pham, A. Ravula, Q. Wang, L. Yang và A. Ahmed, "Con chim lớn: Máy biến áp cho các chuỗi dài hơn," 2020, arXiv:2007.14062. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/2007.14062>
- [38] N. Kitaev, I. Kaiser và A. Levskaya, "Nhà cải cách: Máy biến áp hiệu quả," 2020, arXiv:2001.04451. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/2001.04451> [39] R. Child, S. Gray, A. Radford và I. Sutskever, "Tạo chuỗi dài với máy biến áp thừa thớt," 2019, arXiv:1904.10509. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1904.10509>
- [40] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma và R. Soricut, "ALBERT: Một BERT nhẹ để tự học về việc tái hiện ngôn ngữ," 2020, arXiv:1909.11942. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1909.11942> [41] K. Clark, M.-T. Luong, QV Le và CD Manning, "ELECTRA: Đào tạo trước bộ mã hóa văn bản như bộ phân biệt đối xử chữ không phải bộ tạo," 2020, arXiv:2003.10555. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/2003.10555>
- [42] BA Plummer, N. Dryden, J. Frost, T. Hoefler và K. Saenko, "Mạng Shapeshifter: Chia sẻ tham số nhiều lớp để học sâu hiệu quả và có thể mở rộng," 2020, arXiv:2006.10598. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/2006.10598> [43] M. Joshi, D. Chen, Y. Liu, DS Weld, L. Zettlemoyer và O. Levy, "Span-BERT: Cải thiện đào tạo trước bằng cách biểu diễn và dự đoán các nhịp" Trans. PGS.TS. Máy tính. Ngôn ngữ học, tập. 8, trang 64-77, tháng 12 năm 2020.
- [44] Z. Yang, P. Qi, S. Zhang, Y. Bengio, WW Cohen, R. Salakhutdinov và CD Manning, "HotpotQA: Một tập dữ liệu để trả lời câu hỏi nhiều bước đa dạng, có thể giải thích được" 2018, arXiv:1809.09600. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1809.09600>
- [45] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk và Y. Bengio, "Học cách biểu diễn cụm từ bằng bộ mã hóa-giải mã RNN cho dịch máy thống kê," 2014, arXiv:1406.1078. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1406.1078> [46] S. Hochreiter và J. Schmidhuber, "Trí nhớ ngắn hạn dài", Neural Comput., tập. 9, không. 8, trang 1735-1780, 1997.
- [47] J. Chung, Ç. Gülçehre, K. Cho và Y. Bengio, "Đánh giá thực nghiệm của mạng thần kinh tái phát có kiểm soát trên mô hình trình tự," 2014, arXiv:1412.3555. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1412.3555> [48] M.-T. Luong, H. Pham và CD Manning, "Các phương pháp tiếp cận hiệu quả đối với dịch máy thần kinh dựa trên sự chú ý" 2015, arXiv:1508.04025. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1508.04025>
- [49] D. Bahdanau, K. Cho và Y. Bengio, "Dịch máy thần kinh bằng cách cùng học cách căn chỉnh và dịch," CoRR, tập. abs/1409.0473, trang 1-15, 2015.
- [50] R. Pascanu, T. Mikolov và Y. Bengio, "Về khó khăn trong việc đào tạo mạng lưới thần kinh tái phát," trong Proc. ICML, 2013, trang 1310-1318.
- [51] T. Mikolov, K. Chen, GS Corrado và J. Dean, "Ước tính hiệu quả các cách biểu diễn từ trong không gian vectơ," CoRR, tập. abs/1301.3781, trang 1-12, tháng 1 năm 2013.
- [52] J. Pennington, R. Socher và C. Manning, "Găng tay: Các vectơ toàn cầu để biểu diễn từ" trong Proc. EMNLP, 2014, trang 1532-1543.
- [53] O. Melamud, J. Goldberger và I. Dagan, "Context2vec: Học cách nhúng ngữ cảnh chung với LSTM hai chiều," trong Proc. CoNLL, 2016, trang 51-61.
- [54] B. McCann, J. Bradbury, C. Xiong và R. Socher, "Đã học trong bản dịch: Vectơ từ theo ngữ cảnh," trong Proc. NIPS, 2017, trang 1-12.
- [55] P. Ramachandran, PJ Liu và QV Le, "Đào tạo trước không có giám sát để học theo trình tự," 2017, arXiv:1611.02683. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1611.02683>
- [56] J. Howard và S. Ruder, "Tinh chỉnh mô hình ngôn ngữ phổ quát cho văn bản phân loại" trong Proc. ACL, 2018, trang 328-339.
- [57] X. Liu, P. He, W. Chen và J. Gao, "Mạng lưới thần kinh sâu đa tác vụ để hiểu ngôn ngữ tự nhiên" 2019, arXiv:1901.11504. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1901.11504>
- [58] JL Ba, JR Kiros và GE Hinton, "Chuẩn hóa lớp," 2016, arXiv:1607.06450. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1607.06450> [59] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee và L. Zettlemoyer, "Từ ngữ ngữ cảnh sâu sắc đại diện" trong Proc. NAACL-HLT, 2018, trang 2227-2237.
- [60] Y. Wang, W. Che, J. Guo, Y. Liu và T. Liu, "Chuyển đổi BERT đa ngôn ngữ để phân tích cú pháp phụ thuộc zero-shot," 2019, arXiv:1909.06775. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1909.06775>
- [61] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, và I. Sutskever, "Mô hình ngôn ngữ là những người học đa nhiệm không được giám sát," Công nghệ. Dân biểu, 2019.
- [62] I. Sutskever, O. Vinyals và QV Le, "Học theo trình tự với mạng lưới thần kinh," trong Proc. NIPS, 2014, trang 1-9.
- [63] G. Lample và A. Conneau, "Đào tạo trước mô hình ngôn ngữ đa ngôn ngữ" trong Proc. NeurIPS, 2019, trang 1-11.
- [64] K. Song, X. Tan, T. Qin, J. Lu và T.-Y. Liu, "MASS: Trình tự đeo mặt nạ để đào tạo trước trình tự để tạo ngôn ngữ" trong Proc. ICML, 2019, trang 1-11.
- [65] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin và E. Grave, "CCNet: Trích xuất bộ dữ liệu đơn ngữ chất lượng cao từ dữ liệu thu thập thông tin trên Web," 2020, arXiv:1911.00359. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/1911.00359>
- [66] M. Artetxe, G. Labaka và E. Agirre, "Học cách nhúng từ song ngữ với (gần như) không có dữ liệu song ngữ" trong Proc. ACL, 2017, trang 451-462.
- [67] G. Lample, M. Ott, A. Conneau, L. Denoyer và M. Ranzato, "Dịch máy không giám sát thần kinh và dựa trên cụm từ," 2018, arXiv:1804.07755. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/1804.07755>

[68] TT Baldwin và JK Ford, "Chuyển giao đào tạo: Đánh giá và định hướng cho nghiên cứu trong tương lai," *Personnel Psychol.*, vol. 41, không. 1, trang 63-105, tháng 3 năm 1988.

[69] K. Clark, U. Khandelwal, O. Levy và CD Manning, "BERT nhìn vào cái gì? Phân tích sự chú ý của BERT" 2019, arXiv:1906.04341. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1906.04341> [70] Z. Liu,

M. Sun, T. Zhou, G. Huang và T. Darrell, "Suy nghĩ lại về giá trị của việc cắt tia mạng," 2019, arXiv:1810.05270. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1810.05270> [71] A. Fan, E. Grave và A. Joulin, "Giảm độ sâu của máy

biến áp theo yêu cầu bằng phương pháp bỏ qua có cấu trúc," 2020, arXiv:1909.11556. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1909.11556> [72] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever và R. Salakhutdinov, "Bỏ học: Một

cách đơn giản để ngăn chặn mạng lưới thần kinh bị trang bị quá mức," J. Mach. Học hỏi. Độ phân giải, tập. 15, không. 1, trang 1929-1958, 2014.

[73] L. Wan, MD Zeiler, S. Zhang, Y. LeCun và R. Fergus, "Thường xuyên hóa mạng lưới thần kinh bằng cách sử dụng dropconnect," trong Proc. ICML, 2013, trang 1058-1066.

[74] H. Sajjad, F. Dalvi, N. Durrani và P. Nakov, "BERT của người nghèo: Các mô hình máy biến áp nhỏ hơn và nhanh hơn," 2020, arXiv:2004.03844. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/2004.03844>

[75] S. Narang, E. Elsen, G. Diamos và S. Sengupta, "Khám phá tính thừa thớt trong mạng lưới thần kinh tái phát," 2017, arXiv:1704.05119. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1704.05119>

[76] M. Zhu và S. Gupta, "Cắt tia hay không cắt tia: Khám phá hiệu quả của việc cắt tia để nén mô hình," 2018, arXiv:1710.01878. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1710.01878>

[77] Z. Wang, J. Wohlwend và T. Lei, "Cắt tia có cấu trúc của các mô hình ngôn ngữ lớn" 2020, arXiv:1910.04732. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/1910.04732> [78] E.

Voita, P. Serdyukov, R. Sennrich và I. Titov, "Dịch máy thần kinh nhận biết ngữ cảnh học cách phân giải anaphora" 2018, arXiv:1805.10163. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1805.10163>

[79] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova và J. Lin, "Chất lọc kiến thức về nhiệm vụ cụ thể từ BERT thành các mạng thần kinh đơn giản" 2019, arXiv:1903.12136. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1903.12136> [80]

E. Voita, D. Talbot, F. Moiseev, R. Sennrich, và I. Titov, "Phân tích sự tự chú ý của nhiều người: Những cái đầu chuyên biệt làm việc nặng nhọc, phần còn lại có thể được cắt tia" ở Proc. ACL, 2019, trang 5797-5808.

[81] Y. Ding, Y. Liu, H. Luan và M. Sun, "Trực quan hóa và hiểu bản dịch máy thần kinh" trong Proc. ACL, 2017, trang 1150-1159.

[82] P. Michel, O. Levy và G. Neubig, "Mười sáu cái đầu có thực sự tốt hơn một?" 2019, arXiv:1905.10650. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/1905.10650> [83] D. Zhang, J.

Yang, D. Ye và G. Hua, "LQ-Nets: Đã học lượng tử hóa cho mạng lưới thần kinh sâu nhỏ gọn và có độ chính xác cao," 2018, arXiv: 1807.10029. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1807.10029> [84] A. Zhou, A. Yao, Y. Guo, L. Xu và Y.

Chen, "Lượng tử hóa mạng tăng dần: Hướng tới các CNN không mất dữ liệu với độ chính xác thấp trọng lượng" 2017, arXiv:1702.03044. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1702.03044> [85] X. Lin, C. Zhao và W. Pan, "Hướng tới mạng

trung tính tích chập nhị phân chính xác," 2017, arXiv:1711.11294. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1711.11294> [86] S. Shen, Z. Dong, J. Ye, L. Ma, Z. Yao, A. Gholami, M.

Mahoney và K. Keutzer, "Q-BERT: Siêu dựa trên Hessian lượng tử hóa BERT có độ chính xác thấp" trong Proc. AAAI, 2020, trang 8815-8821.

[87] O. Zafrir, G. Boudoukh, P. Izsak và M. Wasserblat, "Q8BERT: BERT 8bit được lượng tử hóa," 2019, arXiv:1910.06188. [Trực tuyến]. Có sẵn: <https://arxiv.org/abs/1910.06188> [88] B. Jacob, S.

Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam và D. Kalenichenko, "Lượng tử hóa và huấn luyện mạng lưới thần kinh để suy luận hiệu quả chỉ dựa trên số học số nguyên" trong Proc. Hội nghị IEEE/CVF. Máy tính. Vis. Nhận dạng mẫu, tháng 6 năm 2018, trang 2704-2713.

[89] Y. Bengio, N. Léonard và A. Courville, "Ước tính hoặc truyền bá độ dốc thông qua các nơ-ron ngẫu nhiên để tính toán có điều kiện," 2013, arXiv: 1308.3432. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1308.3432> [90] P. Qi, X. Lin, L.

Mehr, Z. Wang và CD Manning, "Trả lời các câu hỏi miễn mở phức tạp thông qua việc tạo truy vấn lặp," 2019, arXiv:1910.07000. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1910.07000> [91] A. Kumar, O. Irsay, P. Ondruska, M. Iyyer, J.

Bradbury, I. Gulrajani, V. Zhong, R. Paulus và R. Socher, "Hãy hỏi tôi bất cứ điều gì: Mạng bộ nhớ động để xử lý ngôn ngữ tự nhiên" trong Proc. ICML, 2016, trang 1378-1387.

[92] R. Al-Rfou, D. Choe, N. Constant, M. Guo và L. Jones, "Mô hình hóa ngôn ngữ ở cấp độ nhân vật với sự tự chú ý sâu sắc hơn" trong Proc. AAAI, 2019, trang 3159-3166.

[93] AN Gomez, M. Ren, R. Urtasun và RB Grosse, "Mạng dự có thể đảo ngược: Lan truyền ngược mà không lưu trữ kích hoạt," 2017, arXiv:1707.04585. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/1707.04585> [94] S. Wang, BZ Li, M. Khabisa, H. Fang

và H. Ma, "Linformer: Tự chú ý với độ phức tạp tuyến tính," 2020, arXiv:2006.04768. [Trực tuyến].

Có sẵn: <http://arxiv.org/abs/2006.04768>

[95] K. Choromanski, V. Likhoshesterov, D. Dohan, X. Song, A. Gane, T. Sarlós, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell và A. Weller, "Suy nghĩ lại về sự chú ý với người biểu diễn" 2020, arXiv:2009.14794. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/2009.14794>

[96] Phân loại mô hình khuôn mặt âm của các mô hình NLP.

[97] W. Fedus, B. Zoph và N. Shazeer, "Chuyển đổi máy biến áp: Mở rộng quy mô đến các mô hình tham số nghìn tỷ với độ thưa thớt đơn giản và hiệu quả," 2021, arXiv:2101.03961. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/2101.03961> [98]

N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, Ú. Erlingsson, A. Oprea và C. Raffel, "Trích xuất dữ liệu đào tạo từ các mô hình ngôn ngữ lớn" 2020, arXiv:2012.07805. [Trực tuyến]. Có sẵn: <http://arxiv.org/abs/2012.07805>



SUSHANT SINGH (Thành viên, IEEE) đã nhận bằng MS về kỹ thuật điện và bằng MS về khoa học máy tính của Đại học Bridgeport, Bridgeport, CT, Hoa Kỳ, lần lượt vào năm 2013 và 2017, nơi anh hiện đang theo đuổi bằng Tiến sĩ. D. bằng cấp về khoa học máy tính.

Anh là Kỹ sư thiết kế mạch của Advanced Micro Devices (AMD), Austin, TX, Hoa Kỳ. Mỗi quan tâm nghiên cứu của ông bao gồm học sâu, xử lý ngôn ngữ tự nhiên, thiết kế VLSI và kiến trúc máy tính.



AUSIF MAHMOOD (Thành viên, IEEE) hiện là Giáo sư của Khoa Khoa học và Kỹ thuật Máy tính. Ông cũng là Giám đốc Trường Kỹ thuật, Đại học Bridgeport. Mỗi quan tâm nghiên cứu của ông bao gồm thị giác máy tính, máy móc và học sâu, kiến trúc máy tính và xử lý song song.

• • •