Mã hóa vị trí đơn giản và hiệu quả cho máy biến áp

Pu-Chin Chen , Henry Tsai , Srinadh Bhojanapalli ,
Hyung Won Chung, Yin-Wen Chang, Chun-Sung Ferng

Nghiên cứu của Google

trừu tượng

Các mô hình máy biến áp là hoán vị tương đương.
Để cung cấp thông tin về thứ tự và loại của mã thông
báo đầu vào, phần nhúng vị trí và phân đoạn thường
được thêm vào đầu vào. Các công trình gần đây đã
đề xuất các biến thể của mã hóa vị trí với mã hóa
vị trí tương đối để đạt được hiệu suất tốt hơn.
Của chúng tôi
phân tích cho thấy rằng lợi ích thực sự đến
từ việc di chuyển thông tin vị trí đến lớp chú ý từ
đầu vào. Được thúc đẩy bởi điều này,
chúng tôi giới thiệu Chú ý vị trí tách biệt
cho Máy biến áp (DIET), một cơ chế đơn giản nhưng hiệu
quả để mã hóa thông tin vị trí và phân đoạn vào các mô
hình Máy biến áp. Phương pháp đề xuất có thời gian huấn
luyện nhanh hơn
và thời gian suy luận, đồng thời đạt được hiệu
suất cạnh tranh trên GLUE, XTREME và
Điểm chuẩn WMT. Chúng tôi khái quát hơn nữa
phương pháp cho máy biến áp tầm xa và hiển thị
đạt được hiệu suất.

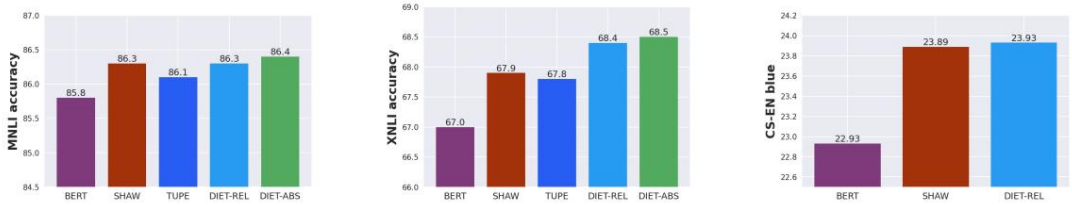
1. Giới thiệu

Máy biến áp là mô hình tuần tự
đạt được hiệu suất hiện đại ở nhiều nơi
Các tác vụ Xử lý ngôn ngữ tự nhiên (NLP), chẳng hạn như
dịch máy, mô hình hóa ngôn ngữ và trả lời câu
hỏi (Vaswani và cộng sự, 2017; Devlin và cộng sự,
2018; Yang và cộng sự, 2019; Liu và cộng sự, 2020). Máy
biến áp có hai thành phần chính: tự chú ý
và một lớp chuyển tiếp nguồn cấp dữ liệu theo vị trí. Cả hai đều là
hoán vị tương đương và không nhạy cảm với
thứ tự của các mã thông báo đầu vào. Để làm cho các mô
hình này nhận biết được vị trí, thông tin vị trí của
các từ đầu vào thường được thêm vào dưới dạng phần
nhúng bổ sung cho phần nhúng mã thông báo đầu vào (Vaswani
và cộng sự, 2017). Ví dụ: nhúng đầu vào (W)
của một câu được thêm vào phần nhúng vị trí
(P), dẫn đến đầu vào W + P cho Máy biến áp.
Những vị trí nhúng này chỉ phụ thuộc vào vị trí mà
từ đó xuất hiện. Đối với các nhiệm vụ nhiều phân đoạn,
các phần nhúng phân đoạn bổ sung có thể được thêm vào chỉ
giống như các phần nhúng vị trí (Devlin và cộng sự, 2018).

Đã có nhiều công trình khám phá những cách khác
nhau để đưa thông tin vị trí vào Transformers (Shaw
và cộng sự, 2018; Yang và cộng sự, 2019; Raffel và
cộng sự, 2020). Nhiều người trong số họ lưu ý đến
những lợi ích của việc sử dụng sơ đồ mã hóa vị trí tương đối.
qua mã hóa vị trí tuyệt đối (xem thêm Hình 1).
Tuy nhiên nguyên nhân gây ra sự khác biệt này vẫn chưa rõ ràng.
Yun và cộng sự. (2020) đã chỉ ra rằng Transformers
với mã hóa vị trí tuyệt đối là các hàm xấp xỉ
phổ quát của tất cả các hàm chuỗi với chuỗi,
chứng minh rằng mã hóa vị trí tuyệt đối có thể nắm bắt
được thông tin vị trí. Do đó nguyên nhân gì
tính ưu việt của mã hóa vị trí tương đối? MỘT
nghiên cứu có hệ thống và hiểu biết về lợi ích
và thiếu các nhược điểm của các phương pháp mã hóa vị
trí khác nhau . Kế và cộng sự. (2020) đưa ra giả thuyết rằng
mối tương quan chéo giữa từ và vị trí
nhúng trong khi tính toán sự chú ý có thể
nguyên nhân của việc thực hiện kém vị trí tuyệt đối
mã hóa. Tuy nhiên, các điều khoản chéo như vậy có mặt
trong một số phương pháp mã hóa vị trí tương đối
(Shaw và cộng sự, 2018; Yang và cộng sự, 2019), và những điều này
các phương pháp thực hiện ngang bằng hoặc tốt hơn các phương pháp khác
sơ đồ mã hóa vị trí (xem §4).

Trong bài viết này chúng tôi thực hiện một nghiên cứu có hệ thống
để hiểu các phương pháp mã hóa vị trí khác nhau.
Chúng tôi lập luận rằng việc nhúng vị trí tuyệt đối chủ yếu
chịu đựng việc được thêm vào ở đầu vào. Chúng tôi hiển thị, với
thử nghiệm của chúng tôi về các nhiệm vụ phân loại, trả
lời câu hỏi và dịch máy, mã hóa vị trí tuyệt đối được
thêm vào ma trận chú ý với
các tham số khác nhau cho mỗi đầu cải thiện đáng kể so
với việc thêm mã hóa vị trí tuyệt đối
đến đầu vào. Điều này nhấn mạnh rằng thông tin vị trí
được đưa vào Máy biến áp là
quan trọng, đưa ra lời giải thích cho khoảng cách trong
hiệu suất giữa mã hóa vị trí tuyệt đối và tương đối. Chúng tôi
cũng so sánh vị trí khác nhau
mã hóa và tác dụng của việc chia sẻ mã hóa vị trí

Các tác giả đóng góp như nhau cho bài viết này.
Email của tác giả tương ứng : puchin@google.com



(a) Học chuyển tiếp tiếng Anh trên MultiNLI

(b) Bật chuyển tiếp đ a ngôn ngữ XNLI

(c) Dịch thuật trên CS-EN

Hình 1: Hiệu quả hiệu suất của các phư ơng pháp mã hóa vị trí khác nhau cho Transformers (xem § 2) trên hai bộ dữ liệu Suy luận ngôn ngữ tự nhiên từ GLUE (Wang và cộng sự, 2019), XTREME (Hu và cộng sự, 2020) và một bộ dữ liệu Dịch máy thần kinh WMT 18 (Bojar và cộng sự, 2018). Mã hóa vị trí tuyệt đ ối (DIET-ABS) có thể đ ạt đ ư ợc hiệu suất tốt hơn so với mã hóa tư ơng đ ối (DIET-REL), cho thấy tầm quan trọng của việc thiết kế phư ơng pháp mã hóa vị trí phù hợp.

xuyên qua các đ ầu và các lớp khác nhau của Máy biến hình. Dựa trên những quan sát này, chúng tôi đ ề xuất sự chú ý vị trí tách rời và một phư ơng pháp mã hóa phân đ oạn mới (đ ối với các nhiệm vụ có nhiều phân đ oạn) và cho thấy tính ư u việt của nó bằng thực nghiệm.

Chúng tôi tóm tắt những đ óng góp của chúng tôi trong bài viết này dư ới đ ây.

- Chúng tôi phân tích về mặt lý thuyết và thực nghiệm giới hạn của việc nhúng vị trí tuyệt đ ối đ ư ợc thêm vào đ ầu vào. Đối với cả thông tin tuyệt đ ối và tư ơng đ ối, chúng tôi cho thấy rằng vị trí mã hóa thành ma trận chú ý trên mỗi đ ầu mang lại hiệu suất vư ợt trội.
- Chúng tôi đ ề xuất một cách đ ơn giản và hiệu quả đ ể mã hóa thông tin vị trí và phân đ oạn. Mã hóa đ ư ợc đ ề xuất phù hợp với các phư ơng pháp SoTA trên nhiều tác vụ NLP tiêu chuẩn trong khi có một mô hình đ ơn giản hơn với chi phí đ ào tạo/suy luận thấp hơn.
- Phư ơng pháp đ ề xuất của chúng tôi có thể dễ dàng áp dụng cho các mô hình chuỗi dài (DIET-ABSLIN) và cải thiện tất cả các chỉ số so với Linformer (Wang et al., 2020).
- Chúng tôi trình bày các nghiên cứu cắt bỏ so sánh các phư ơng pháp mã hóa vị trí khác nhau và các cách chia sẻ thông số mã hóa vị trí giữa các đ ầu và các lớp trong Transformer.

Mã hóa 2 vị trí cho máy biến áp

Trong phần này, chúng tôi xem xét ngắn gọn các mô hình Máy biến áp (Vaswani và cộng sự, 2017) và thảo luận về cải tiến trư ớc đ ây về mã hóa vị trí cũng như phân tích hạn chế của việc nhúng vị trí phụ gia đ ư ợc đ ề xuất trong mô hình Máy biến áp ban đ ầu và đ ư ợc áp dụng rộng rãi.

2.1 Máy biến áp

Khối Transformer bao gồm hai loại lớp: 1) Lớp tự chú ý và 2) Lớp chuyển tiếp nguồn cấp dữ liệu.

Mô-đ un tự chú ý Với đ ộ dài chuỗi đ ầu vào n, kích thư ớc ẩn d, kích thư ớc chiếu xuống khóa truy vấn nhiều đ ầu dh, chúng tôi xác đ ịnh lớp ẩn truy vấn đ ầu vào cho đ ầu chú ý này dư ới dạng $n \times d$, ma trận chiếu X $R^{d \times dh}$ dư ới dạng Wi đ ự án, chính- $R^{d \times dh}$ dư ới dạng Wi $R^{d \times dh}$ và ma trận chiếu giá trị ma trận tion như W_i^V. $R^{d \times dh}$ i [h],,cho đ ầu h .

Thông thư ờng, dh < d khi chúng ta thực hiện chú ý nhiều đ ầu với biểu diễn nhỏ hơn trên mỗi đ ầu (dh = d/h). Với đ iều đ ó, chúng ta có thể viết đ iểm chú ý của sản phẩm chấm:

$$A_i = (XW_i^Q)(XW_i^K)$$

Điểm chú ý này đ ư ợc sử dụng đ ể tính toán đ ầu ra cho mỗi đ ầu, sau khi chia tỷ lệ và chuẩn hóa mỗi hàng bằng softmax:

$$\text{head}_i = \text{Softmax}(A_i / \sqrt{d}) \cdot (XW_i^V)$$

Đầu ra của tất cả các đ ầu chú ý trong một lớp đ ư ợc nối và chuyển sang lớp chuyển tiếp tiếp theo đ ư ợc áp dụng mã thông báo.

2.2 Nhận thức đ ư ợc vị trí của bản thân

Nhiều tác vụ NLP, chẳng hạn như dịch máy, mô hình hóa ngôn ngữ, rất nhạy cảm với thứ tự các từ đ ầu vào. Vì Transformers là hoán vị tư ơng đ ư ơng nên chúng tôi thư ờng đ ư a thêm thông tin vị trí vào đ ầu vào. Dư ới đ ây chúng tôi thảo luận về một số phư ơng pháp mã hóa vị trí phổ biến.

2.2.1 Mã hóa vị trí tuyệt đ ối Mã hóa vị trí

tuyệt đ ối đ ư ợc tính toán trong lớp đ ầu vào và đ ư ợc tổng hợp bằng các mã nhúng mã thông báo đ ầu vào. Vaswani và cộng sự. (2017) đ ề xuất đ iều này

cho Transformers và nó đ ã là một lựa chọn phổ biến trong các công trình tiếp theo (Radford và cộng sự, 2018; Devlin và cộng sự, 2018). Có hai biến thể phổ biến của mã hóa vị trí tuyệt đ ối - cố đ ịnh và học đ ư ợc.

2.2.2 Mã hóa vị trí tư ơng đ ối

Một như ợc đ iểm của mã hóa vị trí tuyệt đ ối là nó yêu cầu đ ộ dài cố đ ịnh của chuỗi đ ầu vào và không không trực tiếp nắm bắt đ ư ợc vị trí tư ơng đ ối của từng từ. Để giải quyết những vấn đ ề này một số vị trí tư ơng đ ối các phư ơng án đ ã đ ư ợc đ ề xuất.

Shaw và cộng sự. (2018) đ ề xuất sử dụng mã hóa vị trí tư ơng đ ối thay vì mã hóa vị trí tuyệt đ ối, và thêm các phần nhúng vị trí vào các phép chiếu khóa và giá trị tùy chọn thay vì đ ầu vào. Họ cho thấy cách mã hóa thông tin vị trí mới này dẫn đ ến hiệu suất tốt hơn trên máy nhiệm vụ dịch thuật. Yang và cộng sự. (2019) đ ã đ ơn giản hóa đ iều này bằng cách loại bỏ các phần nhúng vị trí trong các phép chiếu giá trị và cho thấy hiệu suất tốt hơn trong các nhiệm vụ mô hình hóa ngôn ngữ. Cả hai cách tiếp cận này đ ều sử dụng một biểu diễn vector đ ể mã hóa thông tin vị trí .

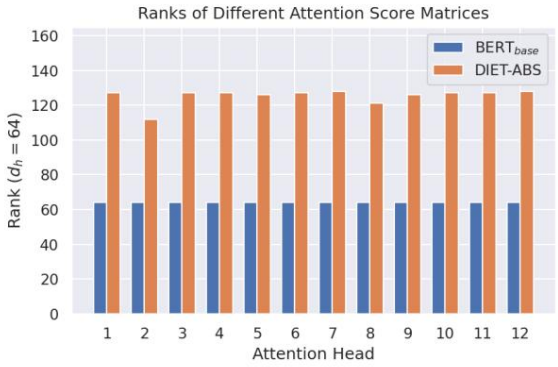
Raffel và cộng sự. (2020) sử dụng đ ại số vô hư ớng đ ể mã hóa vị trí tư ơng đ ối giữa truy vấn và chỉ số chính và thêm trực tiếp vào ma trận đ iểm chú ý. Họ tiếp tục sử dụng việc gộp logarit của thông tin vị trí vào một số nhóm cố đ ịnh. Tất cả những đ iều này phư ơng pháp vị trí tư ơng đ ối chia sẻ thêm vị trí các tham số mã hóa trên các lớp.

Gần đ ây Ke và cộng sự. (2020) đ ư a ra giả thuyết rằng mối tư ơng quan chéo giữa vị trí và phần nhúng mã thông báo có thể dẫn đ ến hiệu suất yếu hơn của phần nhúng vị trí tuyệt đ ối quảng cáo và thay vào đ ó đ ề xuất bổ sung thêm sự chú ý dựa trên cả thông tin vị trí tuyệt đ ối và tư ơng đ ối trực tiếp vào từng cái đ ầu. Tuy nhiên, các thuật ngữ chéo như vậy có mặt trong phư ơng pháp đ ư ợc đ ề xuất bởi Shaw et al. (2018), đ iều đ ó làm cạnh tranh với các phư ơng pháp khác. Thay vào đ ó chúng tôi đ ư a ra giả thuyết rằng mã hóa vị trí ở giới hạn đ ầu vào thứ hạng của ma trận chú ý vị trí dẫn đ ến hiệu suất kém của nó.

2.3 Hạn chế của phụ gia đ ầu vào Nhúng vị trí

Trong phần này chúng ta thảo luận về một số hạn chế của cách thực tế đ ể thêm mã hóa vị trí tuyệt đ ối vào phần nhúng mã thông báo đ ầu vào.

Đầu tiên chúng ta so sánh quyền lực đ ại diện trong đ iều khoản về thứ hạng của ma trận chú ý có thể đ ạt đ ư ợc với các mã hóa vị trí khác nhau.



Hình 2: Thứ hạng của ma trận chú ý: Chúng tôi trình bày một so sánh thứ hạng của ma trận đ iểm chú ý của mô hình BERTBASE với các phần nhúng vị trí tuyệt đ ối ở đ ầu vào so với các phần nhúng vị trí tuyệt đ ối trên mỗi đ ầu (DIET-ABS (1)). Với tính năng nhúng vị trí bổ sung ở đ ầu vào, ma trận chú ý có mức đ ộ chú ý thấp hơn nhiều cấp bậc, hạn chế quyền lực đ ại diện. Điều này đ ư ợc giảm bớt nhờ DIET-ABS.

Định lý 1. Cho P R $^{n\times d}$ là vị trí đ ầu vào nhúng và P^ R n×dp là các nhúng vị trí theo lớp. Đặt WQ,WK R d×dh là ma trận truy vấn và phép chiếu khóa có đ ầu kích thư ớc chiếu dh, và dh < dp, d và n ≥ dh + dp. Đặt Aa = (X + P)WQW K(X + P) và Ar = XWQW KX + P^P^ là ma trận chú ý đ ư ợc tính toán bằng cách sử dụng đ ầu vào và theo lớp vị trí nhúng tư ơng ứng. Sau đ ó với bất kỳ X, P, WQ, WK

$$\text{hạng}(Aa) \le dh.$$

Tồn tại sự lựa chọn X, P^ ,WQ,WK sao cho

$$\text{hạng}(Ar) = dp + dh > dh.$$

Nhận xét. Định lý này cho chúng ta thấy rằng thứ hạng của ma trận chú ý bị ràng buộc với giá trị tuyệt đ ối mã hóa vị trí ở đ ầu vào và sử dụng trên mỗi đ ầu mã hóa vị trí bằng cách thêm thông tin vị trí vào ma trận chú ý trực tiếp dẫn đ ến việc cho phép sự chú ý cấp cao hơn. Xem § B đ ể chứng minh.

Việc thêm mã hóa vị trí trực tiếp vào đ ầu vào càng đ ặt ra một hạn chế đ ối với đ ộng lực đ ào tạo bằng cách buộc các gradient phải giống nhau cho cả đ ầu vào nhúng mã thông báo và vị trí (xem § B). Liên quan đ ến mã hóa vị trí đ ã thảo luận trư ớc đ ó, trong khi giải quyết một số mối lo ngại này, lại gặp phải tình trạng chậm hơn. thời gian đ ào tạo/suy luận (xem Bảng 1) với các cách triển khai phức tạp (Shaw và cộng sự (2018); Ke và cộng sự. (2020)). Trong phần tiếp theo, chúng tôi trình bày đ ơn giản phư ơng pháp mã hóa vị trí đ ể tránh những hạn chế này.

3 Vị trí và phân khúc đề xuất Mã hóa

Trong phần trước, chúng ta đã tìm hiểu về những hạn chế của việc nhúng vị trí phụ gia đầu vào và các công việc hiện có. Dựa trên những quan sát này, chúng tôi đề xuất hai cách tối thiểu/hiệu quả để kết hợp mã hóa vị trí (tuyệt đối/tương đối) cùng với phương pháp mã hóa phân đoạn tuyệt đối mới. Bằng cách tách vị trí và phân đoạn khỏi phần nhúng mã thông báo, chúng tôi điều chỉnh hiệu suất SoTA đồng thời cải thiện thời gian đào tạo/suy luận (xem §3.3).

3.1 Chú ý vị trí tuyệt đối được tách rời Chúng tôi đề xuất phương pháp mã hóa vị trí tuyệt đối đơn giản sau đây để thêm thông tin vị trí vào ma trận chú ý mã thông báo trực tiếp trong mỗi đầu chú ý. Ngoài ra, chúng tôi cũng thêm thông tin phân đoạn vào mã thông báo chú ý thay vì nhúng đầu vào.
Bằng cách này, chúng ta có thể đặt thứ hạng của mã hóa vị trí một cách độc lập, dẫn đến ma trận chú ý có thứ hạng cao hơn, giải quyết các hạn chế đã thảo luận trước đó.

DIET-ABS

$$A^{ABS}_{i,j} = (X_{i:}W_Q)(X_{j:}W_K)^\top / \sqrt{d} + (P_Q P_K^\top)_{i,j} + ES(S(i), S(j)), \quad (1)$$

trong đó $P_Q, P_K \in \mathbb{R}^{n \times d_p}$ là các ma trận xếp vị trí cấp thấp và ES là sự chú ý tuyệt đối của phân khúc đối với các tương tác mô hình giữa các phân đoạn được xác định là

$$ES(S(i), S(j)) = S_{i,j} \quad (2)$$

trong đó $S(i) = i$ nếu chỉ số i nằm trong phân đoạn i.

Xin lưu ý rằng chúng tôi sử dụng ký hiệu sau trong phương trình trên. $A_{i,j}$ biểu thị phần tử (i, j) của ma trận A. $X_{i:}$ và $X_{:j}$ lần lượt biểu thị hàng thứ i và cột thứ j của X. Chúng ta sẽ theo ký hiệu này trong phần còn lại của bài viết.

Theo mặc định, chúng tôi đặt d_p giống như d_h. Điều này đã dẫn đến ma trận chú ý có thứ hạng $d_p + d_h$ như được hiển thị trong Định lý 1. Để minh họa điều này, chúng tôi so sánh thứ hạng của ma trận chú ý trong lớp đầu tiên của mô hình BERT cơ sở và mô hình DIET-ABS cho một mẫu được lấy mẫu. lô trong Hình 2. Hình này cho thấy ma trận chú ý của DIET-ABS có thứ hạng cao hơn BERT cơ sở. Kết quả thử nghiệm chi tiết của chúng tôi trong § 4 cũng cho thấy DIET-ABS hoạt động tốt hơn rõ rệt. Điều này xác nhận quan sát trước đó của chúng tôi trong Định lý 1 rằng việc nhúng vị trí cộng vào ở đầu vào có thể hạn chế mô hình và việc thêm các phần nhúng vị trí trên mỗi đầu sẽ loại bỏ ràng buộc này và mang lại hiệu suất tốt hơn.

Với việc nhúng vị trí tách rời, chúng ta có thể tăng d_p lên bất kỳ chiều rộng k nào để phá vỡ nút cổ chai cấp thấp được hiển thị trong Định lý 1. Chúng tôi gọi đó là mô hình DIET-ABS-Rank-k. Chúng tôi cũng giải quyết hiệu quả vấn đề chính xác được đưa ra bởi một phép nhân ma trận bổ sung ($P_Q P_K^\top$). Vì việc nhúng vị trí độc lập với đầu vào nên chúng ta chỉ cần tính toán phép nhân ma trận một lần cho mỗi đợt huấn luyện và chúng ta có thể lưu trữ ma trận đã tính toán trước khi chạy suy luận. Kết quả là, chúng tôi nhận thấy chi phí suy luận và đào tạo tăng lên đáng kể trong biến thể mô hình này.

3.2 Sự chú ý theo vị trí tương đối được tách rời

Để kết hợp độ lệch quy nạp vị trí tương đối, chúng tôi xem xét một phiên bản đơn giản của mã hóa vị trí được đề xuất trong T5 (Raffel và cộng sự, 2020) mà không cần chia sẻ tham số theo lớp và tạo nhóm nhật ký. Ngoài ra, chúng tôi còn kết hợp mã hóa phân khúc theo đầu người như trong DIET-ABS. Mô hình có thể được viết
BẰNG:

DIET-REL

$$A^{REL}_{i,j} = (X_{i:}W_Q)(X_{j:}W_K)^\top / \sqrt{d + R_{i\,\,j}} + ES(S(i), S(j)). \quad (3)$$

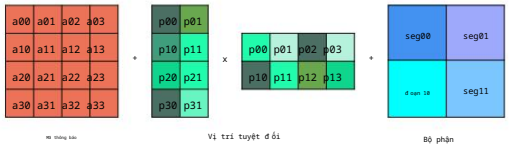
Chúng tôi đưa ra một ví dụ về mô hình này với hai phân đoạn trong Hình 3.

3.3 Chi phí đào tạo và suy luận

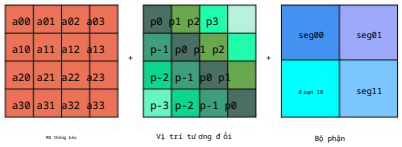
Tiếp theo, chúng tôi trình bày các mô hình được đề xuất có ít chi phí tính toán hơn so với mô hình cơ sở, làm cho mô hình của chúng tôi có tính thực tế hơn các lựa chọn thay thế. Chúng tôi xem xét hai mô hình khác nhau - mô hình BERTBASE và một mô hình nhỏ hơn, BERTSMALL, có kích thước ẩn 512, 4 lớp và 8 đầu chú ý.

Trong Bảng 1, chúng tôi so sánh chi phí đào tạo và suy luận của các phương pháp mã hóa vị trí của Shaw et al. (2018), Kế và cộng sự. (2020), DIET-ABS và DIET-REL. Chúng tôi nhận thấy rằng tính đơn giản của các phương pháp được đề xuất thực sự giúp tiết kiệm cả thời gian huấn luyện và suy luận so với các phương pháp mã hóa vị trí khác. Việc tiết kiệm thời gian bước thậm chí còn đáng kể hơn đối với các mô hình nhỏ hơn (BERTSMALL) và trong quá trình suy luận.

Lưu ý rằng có thể có sự khác biệt giữa tốc độ đào tạo và tốc độ suy luận do cập nhật độ dốc chi phối chi phí tại thời điểm đào tạo (Lan và cộng sự, 2020). Tại thời điểm suy luận, chúng ta chỉ đo thời gian của một



(a) DIET-ABS



(b) DIET-REL

Hình 3: Phương pháp tiếp cận hiệu quả được đề xuất để bao gồm mã hóa vị trí và phân đoạn bằng cách thêm chúng trực tiếp vào ma trận chú ý mã thông báo trên mỗi đầu người. Hình bên trái cho thấy cách chúng tôi mã hóa sự chú ý theo vị trí tuyệt đối. Hình bên phải đại diện cho sự chú ý vị trí tương đối.

	Chế độ Shaw và cộng sự. (2018)	Kế và cộng sự. (2020)	DIET-ABS	DIET-REL
BERTBASE	Đào tạo	+13%	+1%	+0%	+0%
Suy luận BERTBASE	Đào tạo	+33%	+19%	+0%	+0%
BERTSMALL	Suy luận	+24%	+4%	+0%	+0%
BERTSMALL		+65%	+27%	+1%	+0%

Bảng 1: So sánh thời gian tiền huấn luyện và suy luận của Transformers với các phương pháp mã hóa vị trí khác nhau với mô hình BERT cơ bản trên TPU v2. Chúng tôi nhận thấy rằng sự đơn giản của DIET-REL và DIET-ABS dẫn đến tăng đáng kể cả về thời gian huấn luyện và suy luận. Chúng tôi nhận thấy tốc độ thậm chí còn nhanh hơn đối với BERTSMALL nhỏ hơn mô hình so với BERTBASE.

chuyển tiếp tương ứng với chi phí sử dụng những mô hình như vậy trong các hệ thống thực.

3.4 Ứng dụng cho máy biến áp tầm xa

Một ưu điểm khác của phương pháp đề xuất của chúng tôi là chúng dễ dàng mở rộng sang các mẫu máy biến áp tầm xa . Đối với các đầu vào chuỗi dài, Máy biến áp phải chịu sự phụ thuộc bậc hai của khả năng tính toán độ phức tạp liên quan đến độ dài chuỗi. MỘT lớp phương pháp này làm giảm sự phức tạp này bằng cách sử dụng một phép chiếu thứ hạng thấp của chuỗi đầu vào để tính toán sự chú ý (Wang và cộng sự, 2020; Choromanski và cộng sự, 2021; Đại và cộng sự, 2020). Tuy nhiên, các phương pháp như vậy sử dụng mã hóa vị trí đầu vào mặc định và chưa có nhiều công việc trong việc kết hợp thông tin vị trí trên đầu người mà không đưa vào độ phức tạp tính toán bậc hai trên độ dài chuỗi đầu vào. Chúng tôi minh họa khả năng ứng dụng của chúng tôi các phương pháp cài đặt như vậy bằng cách áp dụng DIET-ABS cho Linformer (Wang và cộng sự, 2020), dự kiến ma trận khóa và giá trị chú ý đến thứ nguyên k thấp hơn trong quá trình tính toán chú ý.

DIET-ABSLIN Phương pháp được đề xuất có thể là Viết như :

$$A^{LIN}_{tôi,\ j} = (Xi:WQ)((EX)j:WK) / \sqrt{d} + (PQP\ K)i,j \quad (4)$$

trong đó $E \in R^{k\times n}$, $PQ \in R^{n\times d}$, $PK \in R^{k\times d}$.

4 thí nghiệm

Trong phần này chúng tôi trình bày kết quả thực nghiệm so sánh các phương pháp mã hóa vị trí và phân đoạn khác nhau đã được thảo luận trong các phần trước. Chúng tôi tiến hành thí nghiệm ở ba môi trường khác nhau để bao gồm một loạt các trường hợp sử dụng. Đầu tiên, chúng tôi kiểm tra Kết quả của phương pháp học chuyển giao phổ biến từ việc đào tạo trước LM có mặt nạ đến các nhiệm vụ cuối cùng trong KEO (Devlin và cộng sự, 2018). Thứ hai, chúng tôi nghiên cứu khả năng chuyển đổi đa ngôn ngữ không cần bắn của các mô hình được đào tạo trước đa ngôn ngữ (Hu và cộng sự, 2020) sang nhiệm vụ phân loại và trả lời câu hỏi trong Điểm chuẩn XTREME (Hu và cộng sự, 2020). Cuối cùng, chúng tôi xem xét đào tạo mô hình Transformer từ đầu cho dịch máy.

Chúng tôi so sánh mã hóa vị trí sau đây các phương pháp tiếp cận - nhúng vị trí tuyệt đối (De-vlin và cộng sự, 2018), nhúng vị trí tương đối (Shaw và cộng sự, 2018), kết hợp mã hóa vị trí tuyệt đối và tương đối (Ke và cộng sự, 2020), tương đối phương pháp vô hướng (Raffel và cộng sự, 2020), đề xuất của chúng tôi Phương pháp mã hóa vị trí trên đầu DIET-ABS và DIET-REL . Chúng tôi biểu thị các phương pháp thêm thông tin vị trí/phân đoạn trực tiếp vào đầu vào nhúng mã thông báo với đầu vào và các phương thức thêm thông tin vị trí/phân khúc trực tiếp được chú ý lớp với mỗi đầu. Để thực nghiệm hoàn chỉnh thiết lập, xem Phụ lục A.

4.1 Kết quả học chuyển tiếp tiếng Anh

Bộ dữ liệu và mô hình Để đào tạo trước, chúng tôi sử dụng Bộ dữ liệu Wikipedia và Sách tiếng Anh (Devlin và cộng sự, 2018). Đối với các tác vụ Tinh chỉnh, chúng tôi sử dụng bộ dữ liệu từ điểm chuẩn GLUE (Wang và cộng sự, 2019). Chúng tôi áp dụng mã thông báo từ phụ trên bản thô dữ liệu văn bản bằng WordPiece (Wu và cộng sự, 2016) với 30.000 từ vựng mã thông báo.

Ngư ời mẫu	Phân đ oạn vị trí		MNLI QQP QNLI SST2 Cola STS-B 393k 8.5k 364k 105k 67k 7k				Trung bình
Devlin và cộng sự. (2018) đ ầu vào 85,8 / 85,9 91,1 Shaw et al. (2018) bình quân đ ầu ngư ời				89,9	93,2	58,7	89,0	84,8
86,3 / 86,0 91,2 Raffel et al. (2020) bình quân đ ầu ngư ời đ ầu vào 86,4 / 86,2 91,2 Ke et al. (2020)				90,5	93,2	59,8	89,3	85,2
bình quân đ ầu ngư ời 86,1 / 86,2 91,2 DIET-REL bình quân đ ầu ngư ời 86,0 / 86,1 91,0 DIET-				90,1	93,0	59,6	90,1	85,2
REL bình quân đ ầu ngư ời 86,3 / 86,3 91,0 DIET-ABS (dp=128) bình quân đ ầu ngư ời 86,7 / 86,5				90,3	93,1	59,6	89,6	85,2
91,2 DIET-ABS (dp=128, chia sẻ) bình quân đ ầu ngư ời 86,4 / 86,4 90,8				89,8	92,8	59,6	89,0	84,9
				90,5	92,9	60,3	89,3	85,2
				90,6	92,8	60,1	89,4	85,3
				89,5	93,0	59,8	90,2	85,2
Vư ơng và cộng sự. (2020) (dp=32)	đ ầu	đ ầu vào	82,3 / 82,6 90,2	86,3	91,4	53,9	87,6	82,0
DIET-ABSLIN (dp=32)	vào mỗi đ ầu	đ ầu vào	83,0 / 83,1 90,6	86,7	92,0	55,7	87,6	82,7

Bảng 2: GLUE: Kết quả trên tập phát triển GLUE của các mô hình đ ư ợc tinh chỉnh dựa trên mô hình đ ư ợc huấn luyện trư ớc với 12-kiến trúc lớp BERTBASE . Chúng tôi báo cáo mức trung bình của đ ộ chính xác tối đ a trên tất cả các đ iểm kiểm tra trong số năm đ iểm chạy. Chúng tôi nhận thấy rằng DIET-ABS đ ư ợc chia sẻ với thứ hạng 128 hoạt đ ộng có tính cạnh tranh với vị trí tư ơng đ ối hiện có nhúng các mô hình SoTA mà không có sai lệch quy nạp của các vị trí tư ơng đ ối. Phư ơng pháp đ ề xuất cũng cải thiện hiệu suất trong cài đ ặt máy biến áp tầm xa cấp thấp của (Wang và cộng sự, 2020), trong đ ó vị trí tư ơng đ ối phư ơng pháp nhúng không hiệu quả đ ể sử dụng.

Ngư ời mẫu	Phân đ oạn vị trí		Phân loại XNLI 393k	Trả lời câu hỏi XQuAD MLQA TyDiQA Trung bình 88k 3,7k
Devlin và cộng sự. (2018)	đ ầu vào đ ầu vào bình		67,0	66,0 / 49,9 56,2 / 41,0 59,0 / 47,9 55,3
Shaw và cộng sự. (2018)	quân đ ầu ngư ời đ ầu vào		67,9	69,5 / 53,9 58,2 / 43,1 64,8 / 49,9 58,2
Raffel và cộng sự. (2020)	bình quân đ ầu đ ầu vào		68,5	69,9 / 53,5 59,5 / 44,3 63,8 / 50,6 58,6
Kế và cộng sự. (2020)	bình quân đ ầu vào đ ầu		67,8	68,6 / 52,0 58,6 / 43,2 63,9 / 48,7 57,5
DIET-REL	vào bình quân đ ầu ngư ời		68,0	68,1 / 52,8 57,7 / 42,7 63,3 / 50,9 57,6
DIET-REL	đ ầu vào bình quân đ ầu ngư ời		68,4	69,4 / 54,4 58,6 / 43,5 62,4 / 49,3 58,0
DIET-ABS (dp=128, chia sẻ)			68,5	70,0 / 53,6 59,8 / 44,5 64,6 / 51,5 58,9
Vư ơng và cộng sự. (2020) (dp=256)	đ ầu	đ ầu vào	63,6	59,1 / 43,7 48,9 / 34,0 50,5 / 37,9 48,2
DIET-ABSLIN (dp=256)	vào mỗi đ ầu	đ ầu vào	64,4	61,6 / 46,0 52,2 / 37,0 53,6 / 40,9 50,8

Bảng 3: XTREME: Tinh chỉnh mô hình đ a ngôn ngữ trên tập huấn luyện tiếng Anh (Chuyển giao ngôn ngữ chéo). Hiệu suất đ ư ợc đ o bằng đ ộ chính xác đ ể phân loại và đ iểm F1/khớp chính xác đ ể trả lời câu hỏi. Đồng ý với kết quả trong Bảng 2, chúng tôi thấy trong bảng này rằng việc sử dụng mã hóa vị trí trên mỗi đ ầu hoàn toàn tốt hơn vị trí tuyệt đ ối mã hóa ở đ ầu vào. Với tính năng chia sẻ theo lớp, DIET-ABS xếp hạng 128 vư ợt trội hơn tất cả các mẫu SoTA.

Ngư ời mẫu	EN-DE DE-EN EN-CS CS-EN
Vaswani và cộng sự. (2017)	39,00 38,42 18,55 22,93
Shaw và cộng sự. (2018)	40,10 38,90 18,74 23,89
DIET-REL	39,47 38,49 18,68 23,93

Bảng 4: Dịch máy: Chúng tôi báo cáo kết quả so sánh các phư ơng pháp mã hóa vị trí khác nhau cho Trans-formers trên các tác vụ dịch máy en-de, de-en, en-cs và cs-en từ tập dữ liệu Newstest 2018. Chúng tôi lư u ý rằng tất cả sơ đ ồ mã hóa vị trí trên mỗi đ ầu ngư ời (tất cả ngoại trừ hàng đ ầu tiên) đ ều hoạt đ ộng tốt hơn vị trí tuyệt đ ối các phần nhúng đ ư ợc thêm vào ở đ ầu vào. Hơn nữa đ ề xuất Cách tiếp cận DIET-REL đ ơn giản có tính cạnh tranh với các phư ơng pháp khác các phư ơng pháp mã hóa vị trí

Kết quả Chúng tôi kiểm tra các cách khác nhau của vị trí và phân đ oạn mã hóa ảnh hư ởng đ ến việc truyền khả năng học tập của ngư ời học tiếng Anh BERT tiền đ ào tạo mô hình bằng cách tinh chỉnh trên đ iểm chuẩn GLUE (Wang và cộng sự, 2019), và trình bày kết quả trong Table 2. Đầu tiên chúng tôi nhận thấy rằng tất cả các phư ơng pháp tiếp cận mã hóa các tính năng vị trí rõ ràng ở cấp đ ộ mỗi đ ầu hoạt đ ộng tốt hơn vị trí phụ gia cơ bản mã hóa ở đ ầu vào (Devlin và cộng sự, 2018). Tất cả mô hình kết hợp các vị trí tư ơng đ ối (Shaw et al., 2018; Raffel và cộng sự, 2020; Ke và cộng sự, 2020), mặc dù sự khác biệt về mô hình của họ, có đ iểm trung bình rất giống nhau. Chúng tôi cho thấy mức tăng thêm (84,9 đ ến 85,2 cho DIET-REL) bằng cách di chuyển các đ ặc đ iểm của phân khúc sang trên đ ầu ngư ời.

Điều thú vị là chúng tôi nhận thấy rằng phư ơng pháp mã hóa vị trí tuyệt đ ối đ ư ợc đ ề xuất DIET-ABS, với chia sẻ theo lớp, ngang bằng với tất cả các tính năng chia sẻ trư ớc đ ây Mã hóa vị trí tư ơng đ ối SoTA. Màn trình diễn này mà ngay cả mã hóa vị trí tuyệt đ ối cũng có thể thực hiện tốt hơn khi đ ư ợc bao gồm trên đ ầu ngư ời thay vì ở đ ầu vào. Chúng tôi trình bày một nghiên cứu cắt bỏ chi tiết khác nhau phư ơng pháp xếp hạng và chia sẻ vị trí tuyệt đ ối

chú ý (DIET-ABS) trong Bảng 8 và Bảng 9 trong Phụ lục C.

Đối với đầu vào tầm xa, chúng tôi xem xét Linformer (Wang và cộng sự, 2020) với kích thước hình chiếu là 32. Do chiếu xuống, chúng tôi thấy hiệu suất giảm không đáng kể khi so sánh với Máy biến áp. Ngay cả đối với cài đặt này, chúng tôi thấy rằng sự chú ý tuyệt đối của chúng tôi DIET-ABS có thể được sử dụng để cải thiện hiệu suất của mô hình.

4.2 Kết quả mô hình đa ngôn ngữ

Bộ dữ liệu và mô hình Đối với các thử nghiệm đa ngôn ngữ, chúng tôi đào tạo trước các mô hình trên Wikipedia kho văn bản bằng 100 ngôn ngữ tương tự như (Lample và Conneau, 2019) cho 125 nghìn bước có trình tự dài 512, sau đó tinh chỉnh ở phía dưới Nhiệm vụ của XTREME (Hu và cộng sự, 2020). Chúng tôi sử dụng mã thông báo độc lập với ngôn ngữ, Mảnh câu (Kudo và Richardson, 2018) , với 120.000 từ vựng mã thông báo để mã hóa văn bản đầu vào.

Phân loại Chúng tôi tiến hành 5 thử nghiệm tinh chỉnh cho từng mô hình trên MultiNLI (Williams và cộng sự, 2018) , sau đó thực hiện dự đoán không bắn trúng XNLI (Conneau và cộng sự, 2018), chọn độ chính xác trung bình để báo cáo.

Trả lời câu hỏi Chúng tôi tiến hành 5 thử nghiệm tinh chỉnh cho từng mô hình trên bộ dữ liệu SQuAD V1.1, theo sau là các dự đoán không bắn trên XQuAD (11 ngôn ngữ), MLQA (7 ngôn ngữ) và TyDiQA-GoldP (9 ngôn ngữ), chọn trung vị F1/EM điểm để báo cáo.

Kết quả Chúng tôi trình bày kết quả của mình về việc phân loại và trả lời câu hỏi trong các nhiệm vụ tinh chỉnh trong XTREME cho các phương pháp mã hóa vị trí và phân đoạn khác nhau trong Bảng 3. Một lần nữa, tất cả các phương pháp mã hóa trên mỗi đầu các phương pháp mã hóa vị trí tốt hơn các phương pháp mã hóa vị trí bổ sung đầu vào. Thật thú vị, đơn giản của chúng tôi DIET-ABS hóa ra là mẫu tốt nhất, tốt hơn hơn các mô hình khác sử dụng tính năng vị trí tương đối. Chia sẻ theo lớp và chú ý đến phân khúc theo đầu người cho phép DIET-ABS hoạt động tốt hơn DIET-REL. Chúng tôi trình bày một nghiên cứu cắt bỏ chi tiết trong Bảng 5 để hiểu tác dụng của sự chú ý theo vị trí tách rời các biến thể. Cuối cùng, chúng tôi nhận thấy những lợi thế tương tự trong sử dụng DIET-ABS với Linformer (Wang và cộng sự, 2020) trong cài đặt tầm xa.

4.3 Kết quả dịch thuật

Bộ dữ liệu và mô hình cho dịch máy nhiệm vụ chúng tôi xem xét hai cặp ngôn ngữ (cả hai đều trực tiếp tions) dành cho đào tạo - WMT 2018 English-to-Đức (en-de), tiếng Đức sang tiếng Anh (de-en), tiếng Anh sang tiếng Séc (en-cs) và tiếng Séc sang tiếng Anh (cs-en) (Bo-jar et al., 2018). Chúng tôi kiểm tra các mô hình tương ứng trên bộ dữ liệu Newstest 2018 tương ứng và báo cáo kết quả điểm BLEU của SacreBLEU (Post, 2018) với cài đặt mặc định. Thiết lập của chúng tôi sau Vaswani và cộng sự. (2017) chặt chẽ và sử dụng khung Tensor2Tensor của họ (Vaswani và cộng sự, 2018). Theo dõi Vaswani et al. (2017), chúng tôi sử dụng Transformer 6 lớp với kiến trúc mã hóa-giải mã. Vì biết thêm chi tiết về thiết lập thử nghiệm của chúng tôi xin vui lòng xem Phụ lục A

Kết quả Chúng tôi báo cáo điểm BLEU của các mô hình trong Bảng 4. Chúng tôi quan sát thấy rằng vị trí chuyển động thông tin từ đầu vào đến lớp chú ý trên đầu người cải thiện điểm BLEU. Các biến thể khác nhau của sự chú ý theo vị trí trên mỗi đầu không có tác dụng nhiều sự khác biệt với DIET-REL đang cạnh tranh với Shaw và cộng sự. (2018).

4.4 Nghiên cứu cắt bỏ

Trong phần này, chúng tôi chia sẻ những phát hiện của chúng tôi về các yếu tố chính ảnh hưởng đến hiệu suất của vị trí tách rời chú ý.

Chia sẻ mã hóa vị trí trước đó công trình (Raffel và cộng sự, 2020; Ke và cộng sự, 2020; Shaw và cộng sự, 2018) đã sử dụng các phương pháp chia sẻ khác nhau để mã hóa vị trí để giảm các tham số mô hình . Chúng tôi trình bày một nghiên cứu chi tiết về các hình thức khác nhau chia sẻ mã hóa vị trí và tác dụng của nó đối với hiệu suất. Cụ thể, chúng tôi so sánh các biến thể sau trong việc chia sẻ mã hóa vị trí các tham số trên các đầu khác nhau và các lớp trong Máy biến áp.

- thông minh - Các tham số giống nhau được sử dụng cho tất cả đầu trong một lớp, với các lớp khác nhau bằng cách sử dụng các thông số khác nhau (Shaw và cộng sự, 2018; Ke và cộng sự, 2020).
- theo lớp - Chia sẻ các tham số mã hóa vị trí giữa các lớp với các tham số khác nhau cho mỗi đầu (Raffel et al., 2020).
- không - Mỗi lớp và đầu sử dụng khác nhau thông số mã hóa vị trí.

Chúng tôi trình bày kết quả so sánh các chia sẻ khác nhau các phương pháp trong Bảng 5 cho các nhiệm vụ XTREME. Chúng tôi làm những quan sát sau đây 1) chia sẻ không ngoan là luôn tệ hơn so với lớp không ngoan, 2) chia sẻ gây tổn hại

Ngư ời mẫu	Phân đ oạn chia sẻ	Phân loại XNLI	Trả lời câu hỏi XQuAD MLQA TyDiQA-GoldP		Trung bình
DIET-REL	- đ ầu	68,0	68,1 / 52,8 57,7 / 42,7	63,3 / 50,9	57,6
DIET-REL	theo từng vào	67,7	66,2 / 51,0 56,0 / 41,1	60,1 / 45,9	55,4
DIET-REL	lớp theo đ ầu đ ầu	68,0	68,6 / 53,3 58,1 / 43,1	61,3 / 48,2	57,2
DIET-REL	- vào đ ầu	68,4	69,4 / 54,4 58,6 / 43,5	62,4 / 49,3	58,0
DIET-REL	vào theo đ ầu theo đ ầu	67,8	66,0 / 50,5 55,5 / 40,4	59,2 / 44,6	54,7
DIET-REL	theo lớp theo đ ầu	68,1	68,7 / 53,8 5 8,4 / 43,2	61,0 / 48,4	57,3
Đầu vào DIET-ABS (dp=64)	-	68,0	67,4 / 50,5 57,8 / 42,3	61,3 / 46,8	56,3
DIET-ABS (dp=64) trên đ ầu ngư ời	-	67,9	67,5 / 52,4 57,3 / 42,3	61,6 / 46,8	56,5
DIET-ABS (dp=128) trên đ ầu ngư ời	-	68,1	68,2 / 52,0 57,9 / 42,6	61,5 / 47,6	56,8
DIET-ABS (dp=512) trên đ ầu ngư ời	-	68,5	68,0 / 52,0 57,7 / 42,4	61,6 / 48,4	56,9
Đầu vào theo lớp DIET-ABS (dp=64)		68,0	69,3 / 53,1 59,3 / 43,9	63,2 / 48,6	57,9
DIET-ABS (dp=64) theo từng lớp trên đ ầu		68,4	69,3 / 53,2 5 9,4 / 44,1	63,3 / 48,6	58,0
DIET-ABS (dp=128) theo từng lớp trên đ ầu		68,5	70,0 / 53,6 59,8 / 44,5	64,6 / 51,5	58,9
DIET-ABS (dp=256) theo từng lớp trên đ ầu		68,4	69,9 / 53,8 59,6 / 44,2	62,8 / 49,1	58,3
DIET-ABS (dp=512) theo từng lớp trên đ ầu		67,8	69,0 / 53,2 58,4 / 43,0	62,5 / 48,8	57,5

Bảng 5: Nghiên cứu cắt bỏ trên XTREME: Chúng tôi thực hiện nghiên cứu cắt bỏ sự chú ý theo vị trí tách rời để hiểu đ ược tác dụng của 1) chia sẻ các tham số chú ý vị trí giữa các lớp và đ ầu 2) sự chú ý phân khúc đ ược thêm vào ở mỗi đ ầu 3) hiệu suất tư ơng đ ối và tuyệt đ ối 4) sự chú ý vị trí tuyệt đ ối xếp hạng dp từ 64 đ ến 512.

	Tiếng Anh Thông số + GLUE	Thông số + XTREME	Đa ngôn ngữ		
Devlin và cộng sự. (2018)	110,1 triệu 84,8	112,9 triệu +2,5%	178,9M	-	55,3
Shaw và cộng sự. (2018)	85,2 109,9 triệu	+0,0% 85,2 109,7	181,7M	+1,7%	58,2
DIET-REL	triệu +0,0% 85,0	128,6 triệu	178,7M	+0,0%	58,0
DIET-REL (chia sẻ)	+16,8% 85,3	111,3 triệu +1,1% 85,2	178,5M	+0,0%	57,3
DIET-ABS (dp=128)			197,4M	+10,0%	56,8
DIET-ABS (dp=128, chia sẻ)			180,1M	+0,6%	58,9

Bảng 6: Tham số mô hình: Chúng tôi liệt kê số lư ợng tham số mô hình và hiệu suất cho các phư ơng pháp mã hóa vị trí khác nhau. Chúng tôi nhận thấy rằng việc chia sẻ làm tổn hại đ ến hiệu suất của DIET-REL với lợi ích không đ áng kể về mặt số lư ợng tham số. Ngư ợc lại, tác dụng chính quy hóa của việc chia sẻ giúp DIET-ABS ổn đ ịnh hơn với các thông số ít hơn để đ ạt đ ư ợc hiệu suất cạnh tranh.

hiệu suất của DIET-REL trong khi nó cải thiện
Hiệu suất của DIET-ABS. Chúng tôi tóm tắt
cài đ ặt chính cùng với số thông số mô hình
trong Bảng 6. Đối với DIET-REL, việc chia sẻ mang lại
ít ảnh hư ởng đ ến việc lư u tham số và ảnh hư ởng đ ến
hiệu suất . Do đ ó, chúng tôi khuyên bạn không nên chia sẻ
mã hóa vị trí tư ơng đ ối (DIET-REL). Trên
mặt khác, cần phải chia sẻ các thông số cho
DIET-ABS đ ể giữ số lư ợng thông số ở mức thấp.
Điều thú vị là việc chia sẻ đ ã đ ư ợc chính quy hóa
ảnh hư ởng đ ến DIET-ABS, làm cho mô hình hoạt đ ộng
tốt hơn. Chúng tôi chọn chia sẻ theo lớp thay vì chia sẻ theo
chiều sâu đ ể có hiệu suất tốt hơn.

Mã hóa phân đ oạn Thiết kế mã hóa phân đ oạn mới của chúng tôi cải thiện hơn nữa hiệu suất của mô hình đ ư ợc trình bày trong Bảng 5. Cả hai mô hình chú ý vị trí tách rời tư ơng đ ối và tuyệt đ ối đ ều đ ư ợc hư ởng lợi từ việc chuyển mã hóa phân đ oạn từ đ ầu vào sang đ ầu ngư ời: DIET- REL (+0,4%), theo lớp

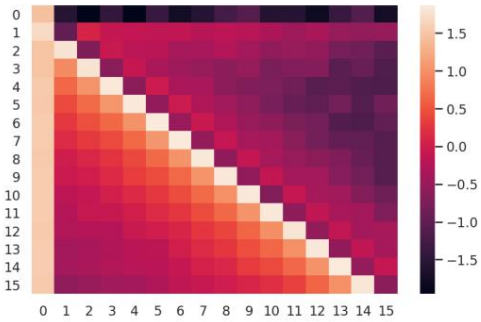
chia sẻ DIET-REL (+0,1%), DIET-ABS (+0,2%),
DIET-ABS đ ư ợc chia sẻ theo lớp (+0,1%). Xem ứng dụng-
phụ lục D về kết quả của đ iểm chuẩn GLUE và
Phụ lục C đ ể hình dung sự chú ý của phân khúc.

Xếp hạng của sự chú ý theo vị trí tuyệt đ ối Thiết kế của DIET-ABS cho phép tìm hiểu các ma trận chú ý ở cấp đ ộ cao hơn như trong Định lý 1. Để hiểu đ ư ợc tác đ ộng của thứ hạng chú ý

theo vị trí tuyệt đ ối
(dp) trong thực tế, chúng tôi tiến hành các thí nghiệm khác nhau
thứ hạng từ dp = 64 đ ến dp = 512. Chúng tôi trình bày
kết quả trong Bảng 5. Chúng tôi nhận thấy rằng hiệu suất đ ư ợc
cải thiện khi chúng tôi tăng thứ hạng từ 64 lên
128. Tuy nhiên, có sự bão hòa về hiệu suất trong
tiếp tục tăng nó lên 512. Chúng tôi trình bày một cách trực
quan hóa thứ hạng của ma trận chú ý vị trí
trong Phụ lục B.

4.5 Trực quan hóa mẫu chú ý theo vị trí

Tiếp theo chúng ta hình dung sự chú ý theo vị trí đ ã học đ ư ợc mô hình của DIET-ABS trong Hình 4. Đầu tiên chúng tôi lư u ý rằng DIET-ABS đ ã học đ ư ợc cách nắm bắt mối quan hệ họ hàng quan hệ vị trí giữa các yếu tố đ ầu vào. Cũng lư u ý rằng, đ ối với chỉ số 0 (mã thông báo [CLS]), đ ư ợc tách rời



Hình 4: Hình dung sự chú ý theo vị trí đã học được mô hình DIET-ABS. Lưu ý rằng ngoài việc nắm bắt các mối quan hệ vị trí tương đối, mô hình còn học cách tham dự [CLS] ở chỉ số 0, gợi ý thiết kế cởi trói [CLS] chuyên dụng trong Ke et al. (2020) thì không cần thiết với DIET-ABS.

sự chú ý tuyệt đối về vị trí thường học được một khuôn mẫu đặc biệt. Mẫu này không thể chỉ được mô hình hóa bằng các phương pháp nhúng vị trí tương đối hiện có, và một số công trình hiện có (Ke và cộng sự, 2020) đã xử lý trường hợp này cụ thể bằng cách giới thiệu các thông số mới. Điều này cho thấy lợi ích của DIET-ABS trong việc không yêu cầu bất kỳ thành kiến quy nạp được thiết kế cẩn thận như trong các phương pháp tiếp cận hiện tại (Shaw và cộng sự (2018); Raf-fel và cộng sự (2020)), có thể không khái quát hóa được nhiệm vụ.

5. Kết luận

Trong bài báo này, chúng tôi đã xem xét về mặt lý thuyết và thực nghiệm giới hạn của việc nhúng vị trí phụ gia ở đầu vào và chỉ ra rằng việc nhúng vị trí trên mỗi đầu sẽ mang lại hiệu suất tốt hơn. Chúng tôi lập luận rằng hiệu suất vượt trội của một số các phương pháp mã hóa vị trí tương đối đến từ sự bổ sung trên đầu người của họ vào ma trận chú ý thay vì hơn thông tin vị trí là tương đối so với tuyệt đối . Thật vậy, chúng tôi chứng minh rằng việc sử dụng vị trí tuyệt đối mã hóa trên mỗi đầu mang lại hiệu suất tốt hơn. Được thúc đẩy bởi điều này, chúng tôi đề xuất một phương pháp chú ý theo phân khúc và vị trí trên đầu người đơn giản để đạt được hiệu suất tiên tiến trên nhiều NLP nhiệm vụ và hiệu quả tính toán hơn so với các cách tiếp cận hiện có.

Người giới thiệu

Ondˇrej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn và Christof Monz. 2018. Kết quả của hội nghị về dịch máy năm 2018 (WMT18). Trong Biên bản của Hội nghị lần thứ ba về dịch máy: Tài liệu nhiệm vụ chung, trang 272–303, Bel-gium, Brussels. Hiệp hội tính toán Lin-guistics.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Bài hát Xingyou, Andreea Gane, Tamas Sar-los, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, và Adrian Weller. 2021. Suy nghĩ lại về sự chú ý với người biểu diễn.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Ad-ina Williams, Samuel R. Bowman, Holger Schwenk, và Veselin Stoyanov. 2018. Xnli: Đánh giá cách trình bày câu đa ngôn ngữ. Trong Kỷ yếu của Hội nghị năm 2018 về các phương pháp thực nghiệm trong xử lý ngôn ngữ tự nhiên. Hiệp hội ngôn ngữ học tính toán.

Zihang Dai, Guokun Lai, Yiming Yang và Quốc V. Lê. 2020. Bộ biến đổi kênh: Lọc ra phần dư thừa tuần tự để xử lý ngôn ngữ hiệu quả.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, và Kristina Toutanova. 2018. BERT: Đào tạo trước cho Transformers hai chiều sâu sắc để hiểu ngôn ngữ. bản in trước arXiv arXiv:1810.04805.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat và Melvin Johnson. 2020. Xtreme: Điểm chuẩn đa tác vụ đa ngôn ngữ để đánh giá khái quát hóa đa ngôn ngữ. TRONG Kỷ yếu của Hệ thống và Học máy 2020, trang 7449–7459.

Guolin Ke, Di He và Tie-Yan Liu. 2020. Suy nghĩ lại về mã hóa vị trí trong đào tạo trước ngôn ngữ. bản in trước arXiv arXiv:2006.15595.

Taku Kudo và John Richardson. 2018. Câu văn: Một trình mã hóa và trình mã hóa từ phụ đơn giản và độc lập với ngôn ngữ để xử lý văn bản thần kinh.

Guillaume Lample và Alexis Conneau. 2019. Cross-Đào tạo trước mô hình ngôn ngữ ngôn ngữ

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma và Radu Soricut. 2020. Albert: Một chuyên gia nhỏ về việc tự học cách biểu diễn ngôn ngữ.

Xiaodong Liu, Kevin Duh, Liyuan Liu và Jianfeng Cao. 2020. Máy biến áp rất sâu cho dịch máy thần kinh.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Man-dar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer và Veselin Stoyanov. 2019. RoBERTa: Đào tạo trước BERT được tối ưu hóa mạnh mẽ tiếp cận. bản in trước arXiv arXiv:1907.11692.

bài Matt. 2018. Kêu gọi sự rõ ràng trong báo cáo màu xanh điểm số. Trong WMT.

Alec Radford, Karthik Narasimhan, Tim Salimans và Ilya Sutskever. 2018. Nâng cao khả năng hiểu ngôn ngữ bằng cách đào tạo trước mang tính khái quát. Báo cáo kỹ thuật, OpenAI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li và Peter J Liu. 2020. Khám phá những hạn chế của việc học chuyển tiếp bằng công cụ chuyển đổi văn bản thành văn bản thống nhất. Tạp chí Nghiên cứu Học máy, 21(140):1–67.

Peter Shaw, Jakob Uszkoreit và Ashish Vaswani. 2018. Tự chú ý với các đại diện vị trí tương đối. Trong Kỷ yếu Hội nghị năm 2018 của Chi hội Bắc Mỹ của Hiệp hội Ngôn ngữ học Tính toán: Công nghệ Ngôn ngữ Con người, Tập 2 (Bài viết ngắn), trang 464–468.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Fran-cois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, và những người khác. 2018. Tensor2tensor cho dịch máy thần kinh. bản in trước arXiv arXiv:1803.07416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser và Illia Polosukhin. Năm 2017. Sự chú ý là tất cả những gì bạn cần. Trong Những tiến bộ trong hệ thống xử lý thông tin thần kinh, trang 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Fe-lix Hill, Omer Levy và Samuel Bowman. 2019. Keo: Nền tảng phân tích và điểm chuẩn đa tác vụ để hiểu ngôn ngữ tự nhiên. Trong Hội nghị quốc tế lần thứ 7 về đại diện học tập, ICLR 2019.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang và Hao Ma. 2020. Linformer: Tự chú ý với độ phức tạp tuyến tính.

Yu-An Wang và Yun-Nung Chen. 2020. Việc nhúng vị trí học được gì? một nghiên cứu thực nghiệm về mã hóa vị trí mô hình ngôn ngữ được đào tạo trước. Trong EMNLP 2020.

Adina Williams, Nikita Nangia và Samuel R. Bow-man. 2018. Kho ngữ liệu thử thách có phạm vi rộng để hiểu câu thông qua suy luận.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quốc V. Lê, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin John-son, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rud-nick, Oriol Vinyals, Greg Corrado, Macduff Hughes và Jeffrey Dean. 2016. Hệ thống dịch máy thần kinh của Google : Thu hẹp khoảng cách giữa dịch người và dịch máy.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov và Quốc V. Lê. 2019. XLNet: Huấn luyện trước tự hồi quy tổng quát để hiểu ngôn ngữ. bản in trước arXiv arXiv:1906.08237.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi và Sanjiv Kumar. 2020. Transformers có phải là công cụ xấp xỉ phổ quát của các hàm tuần tự không? Trong Hội nghị quốc tế về đại diện học tập.

Một thiết lập thử nghiệm

Trong phần này, chúng tôi trình bày thêm chi tiết về thiết lập thử nghiệm của chúng tôi.

Đào tạo trư ớc Chúng tôi đ ào tạo trư ớc các mô hình bằng cách sử dụng tác vụ LM bị che (Devlin và cộng sự, 2018) và không sử dụng tính năng mất Dự đ oán câu tiếp theo (NSP) như đ ư ợc đ ề xuất trong RoBERTa (Liu và cộng sự, 2019). Mỗi đ ầu vào đ ư ợc xây dựng với các câu đ ầy đ ủ từ các tài liệu và đ ư ợc đ óng gói theo đ ộ dài chuỗi tối đ a. Chúng tôi sử dụng kiến trúc tư ơng tự như BERTBASE (Devlin và cộng sự, 2018) (L = 12, H = 768, A = 12) cho các thử nghiệm của mình.

Tinh chỉnh Một số tác vụ tiếp theo có các nhóm câu đ ầy đ ủ khác nhau đ ư ợc cung cấp ở đ ầu vào. Đối với những tác vụ đ ó (ví dụ: MNLI, CoLA, XNLI, SQuAQ), chúng tôi tinh chỉnh các mô hình bằng mã hóa phân đ oạn bổ sung đ ư ợc thảo luận trong Phần §3. Chúng tôi giữ nguyên các mô hình cho các nhiệm vụ khác dư ới dạng thư từ trư ớc khi đ ào tạo.

Siêu tham số Siêu tham số chúng tôi sử dụng đ ư ợc trình bày trong Bảng 7.

	Tiếng Anh		Đa ngôn ngữ	
	Huấn luyện trư ớc	Finetune luyện tập trư ớc		Tinh chỉnh
Số bư ớc tối đ a	500K	5 hoặc 10 kỷ	125K	3 kỷ
Tỷ lệ học	nguyên 0,0018 {1e-5, 2e-5, 3e-5, 4e-5}		nguyên 0,0018 {1e-5, 2e-5, 3e-5, 4e-5}	
Tỷ lệ khởi đ ộng	0,025 0,1 128 128 4096 32 20k 3,5k		0,025 0,1 512 512 4096 32 20k 3,5k	
Độ dài chuỗi				
Kích thư ớc lô				
Khoảng thời gian kiểm tra				

Bảng 7: Siêu tham số cho tất cả các model

Dịch Đối với các thử nghiệm Dịch của chúng tôi, chúng tôi tuân theo cách thiết lập của Vaswani et al. (2017) và sử dụng khung Tensor2Tensor của họ (Vaswani và cộng sự, 2018). Chúng tôi đ ào tạo bằng cách sử dụng các bộ dữ liệu WMT18 ((Europarl v7, Common Crawl corpus và News Commentary v13) en-de, de-en, en-cs và cs-en. Chúng tôi báo cáo đ iểm BLUE do SacreBLEU (Post, 2018) cung cấp trên tập dữ liệu mới nhất 2018 Chúng tôi đ ào tạo mô hình Transformer 6 lớp. Mọi thay đ ổi đ ối với mã hóa vị trí đ ều đ ư ợc áp dụng cho tất cả các lớp chú ý cả trong bộ mã hóa và bộ giải mã. Chúng tôi sử dụng trình tối ư u hóa Adam và đ ào tạo 250 nghìn bư ớc. Để giải mã, chúng tôi sử dụng tìm kiếm chùm tia với kích thư ớc chùm tia 10 và phạt chiều dài 0,6.

B Bằng chứng

Chứng minh Định lý 1. Dễ dàng nhận thấy khẳng đ ịnh đ ầu tiên bằng cách nhận thấy rằng hạng tích của hai ma trận bị giới hạn trên bởi mức tối thiểu của các hạng riêng lẻ.

$$\text{hạng}(Aa) = \text{hạng}((X + P)W_Q W_K(X + P)^\top) \le \min(\text{thứ}$$
$$\text{hạng}(X + P), \text{hạng}(W_Q), \text{hạng}(X + P), \text{hạng}(W_K)) \le d_h.$$

$$\text{xếp hạng}((X + P)W_Q W_K(X + P)^\top) \le d_h, \text{ trong đ ó } W_Q, W_K \in \mathbb{R}^{d\times d_h}$$

Bất đ ẳng thức cuối cùng suy ra từ hạng$(W_Q) \le d_h$ là $W_Q \in \mathbb{R}^{d\times d_h}$ Để chứng minh khẳng đ ịnh thứ hai, chúng ta áp dụng phư ơng pháp xây dựng. Trư ớc tiên chúng ta lấy $W_Q = W_K$ bằng nhau ma trận với các hàng d_h đ ầu tiên là ma trận nhận dạng và các hàng $d - d_h$ còn lại đ ều là số 0. Sau đ ó

$$W_Q W_K^\top = \begin{matrix} I_{d_h,d_h} & 0_{d_h,d-d_h} \\ 0_{d-d_h,d_h} & 0_{d-d_h,d-d_h} \end{matrix}.$$

Ở đ ây I_{d_h,d_h} biểu thị ma trận đ ồng nhất trong $\mathbb{R}^{d_h\times d_h}$ và $0_{d_h,d}$ biểu thị ma trận toàn số 0 trong $\mathbb{R}^{d_h,d}$.

Chúng ta đ ặt X sao cho d hàng đ ầu tiên tạo thành ma trận nhận dạng và phần còn lại là số 0 - $X = [I_{d,d}, 0_{n-d,d}]$. Do đ ó $XW_Q W_K X^\top$ trở thành ma trận đ ư ờng chéo tư ơng tự với

$$XW_Q W_K X^\top = \begin{matrix} I_{d_h,d_h} & 0_{d_h,n-d_h} \\ 0_{n-d_h,d_h} & 0_{n-d_h,n-d_h} \end{matrix}.$$

Chọn dp = n > dh và đặt Pˆ = I. Bây giờ chọn Pˆ với các số 0 ở các cột n dp đầu tiên và đẳng thức ở các cột dp cuối cùng (Pˆ = [0d,n dp , Idp,dp]) sẽ cho kết quả

$$P\hat{P}\hat{} = \begin{matrix} 0_{n\ dp,n\ dp} & 0_{n\ dp,dp} \\ 0_{dp,n\ dp} & I_{dp,\ dp} \end{matrix}.$$

Kết hợp cả hai điều này mang lại cho chúng ta

$$\text{hạng}(Ar) = \text{hạng}(XWQW\ KX + P\hat{P}\hat{}\ ) = \min(dh + dp,\ n) > dh.$$

□

Đặt $X \in R^{n\times d}$ là phần nhúng từ đầu vào trong chiều d với độ dài chuỗi n. Chúng tôi có thể huấn nhúng vị trí $P \in R^{n\times d}$, luyện được thêm vào chuỗi đầu vào trước khi đưa vào mô hình g. Đối với đầu vào X nhất định và nhãn y, mục tiêu của hàm mất mát như sau:

$$L = (g(X + P),\ y) \tag{5}$$

Định lý 2. Cho X và P là các ma trận nhúng có thể huấn luyện được trong $R^{n\times d}$. Khi đó, độ dốc của hàm mất trong phương trình (5), tại bất kỳ điểm nào (X, y) và đối với mọi hàm khả vi và g, đều giống nhau đối với X và P.

Nhận xét. Định lý này cho chúng ta thấy rằng độ dốc giống nhau đối với các nhúng mã thông báo đầu vào và các nhúng vị trí. Mặc dù trong các tác vụ NLP tiêu chuẩn, đầu vào X có thể khác nhau ở mỗi bước do các mã thông báo đầu vào khác nhau xuất hiện trong mỗi lô nhỏ, kết quả vẫn cho thấy rằng việc nhúng vị trí phụ gia có thể hạn chế mô hình tìm hiểu tầm quan trọng tương đối của mã hóa vị trí đối với mã thông báo nhúng dựa trên nhiệm vụ đào tạo hiện tại.

Chứng minh Định lý 2. Định lý trên được thực hiện bằng cách chỉ tính các gradient và cho thấy chúng bằng nhau ở mỗi bước.

Độ dốc của mục tiêu trên X và P như sau.

$$\begin{aligned} \nabla_X L &= g_L \cdot \nabla_X + g \cdot \nabla_X (X + P) = \\ &g_L \cdot \nabla_X + g \quad P_L \\ &= g_L \cdot \nabla_X + g \cdot P(X + P) = g_L \cdot \\ &X + T_r. \end{aligned}$$

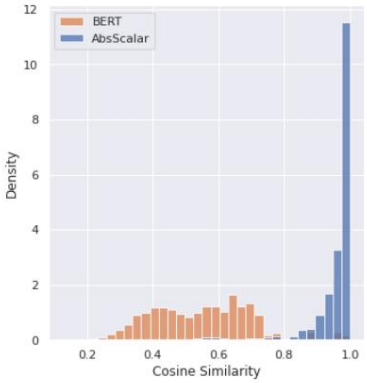
Việc tính toán độ dốc ở trên tuân theo quy tắc dây chuyền. Điều này cho thấy độ dốc của L wrt X và P là như nhau.

□

C Trực quan hóa sự chú ý

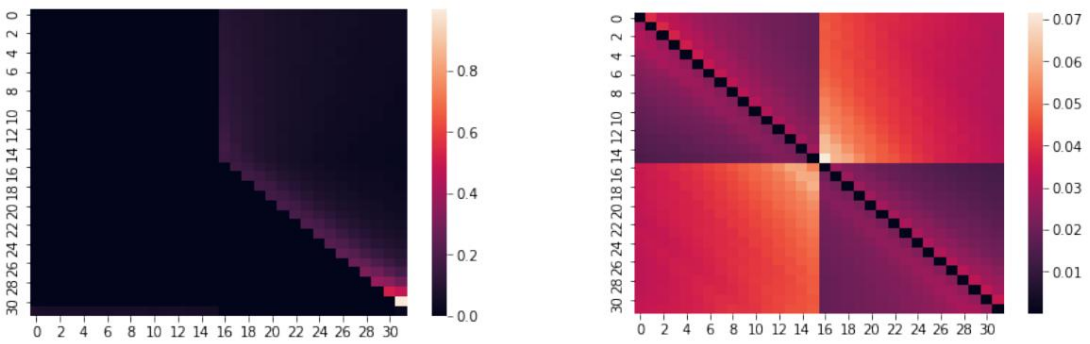
Trong phần này, chúng tôi kiểm tra nội bộ của mô hình để hiểu cách thức hoạt động của mô hình được đề xuất. Trước tiên, chúng tôi hình dung nội dung bên trong mô hình của các lựa chọn thay thế mô hình khác nhau để tranh luận rằng mô hình đề xuất của chúng tôi là hợp lý.

Tại sao chúng tôi loại bỏ phần nhúng đầu vào Để hiểu liệu có hợp lý hay không khi loại bỏ phần nhúng phụ gia đầu vào sau khi thêm vô hướng vị trí trên mỗi đầu, chúng tôi thêm phần nhúng vị trí phụ gia vào mô hình DIET-ABS của chúng tôi . Sau đó, chúng tôi kiểm tra việc nhúng vị trí của mô hình BERT và biến thể DIET-ABS của chúng tôi bằng cách nhúng vị trí phụ gia. Hình 5 cho thấy rằng, khi mô hình có cả nhúng vị trí tuyệt đối vô hướng tuyệt đối và bổ sung, việc nhúng vị trí hầu như không mã hóa thông tin - tất cả các nhúng vị trí ở đầu vào đều tương tự nhau.



Hình 5: Phân bố độ tương tự cosin giữa tất cả các cặp vị trí tuyệt đối của nhúng vị trí phụ gia đầu vào cho mô hình BERT cơ sở và DIET-ABS được đề xuất. Chúng tôi quan sát thấy rằng, sau khi các tính năng vị trí được thêm vào mỗi đầu như trong DIET-ABS, việc nhúng vị trí đầu vào hầu như không chứa thông tin - tất cả các cặp vị trí đầu vào đều giống nhau.

Tác động của việc chú ý đến phân khúc Chúng tôi cũng xem xét tác động của việc thêm sự chú ý vào phân khúc lên trên sự chú ý về vị trí. Hình 6 cho thấy một số mẫu tiêu biểu. Chúng tôi nhận thấy rằng sự chú ý đến phân khúc cho phép mô hình chú ý nhiều hơn đến các phần của chuỗi thuộc các phân khúc nhất định.

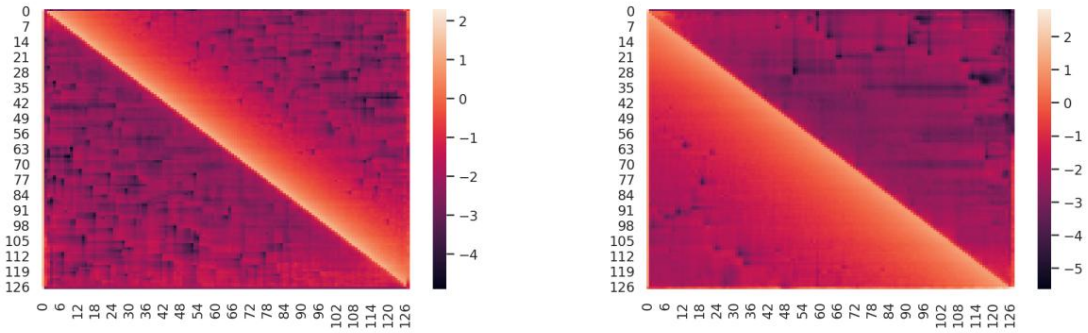


(a) Tham dự Phần thứ hai

(b) Chú ý đến vị trí tương đối của trọng lượng thấp hơn

Hình 6: Chúng ta xem xét đầu vào có độ dài 32 với hai đoạn. Phân đoạn thứ hai bắt đầu ở chỉ số 16. Chúng tôi quan sát các mẫu chú ý trong mô hình DIET-REL mà không chú ý đến mã thông báo.

Mô hình dịch chuyển đ ư ợc học từ sự chú ý vị trí tuyệt đ ối Sử dụng mã hóa vị trí tư ơng đ ối nhìn chung mang lại kết quả tốt hơn mặc dù quy mô cải tiến nhỏ hơn so với mã hóa tính năng di chuyển trên mỗi đ ầu ngư ời. Để hiểu đ iều này, chúng tôi hình dung mẫu chú ý của chú ý vị trí tuyệt đ ối và tìm thấy hai mẫu đ ại diện trong DIET-ABS trong Hình 7. Chúng tôi nhận thấy rằng ngay cả khi đ ư a ra các đ ặc đ iểm vị trí tuyệt đ ối, phần lớn mô hình đ ều học đ ư ợc "mô hình dịch chuyển". Khác với Wang và Chen (2020) khẳng đ ịnh vị trí tuyệt đ ối chỉ tìm hiểu các mô hình đ ịa phư ơng, chúng tôi cho thấy sự chú ý của vị trí thực sự có thể chú ý đ ến bối cảnh dài hơn. Tuy nhiên, mô hình dịch chuyển có thể đ ư ợc mô hình hóa trực tiếp bằng vị trí tư ơng đ ối. Do đ ó, DIET-REL có thể là lựa chọn mô hình tốt hơn với ít thông số hơn và đ ộ lệch quy nạp chính xác hơn trong một số ứng dụng.

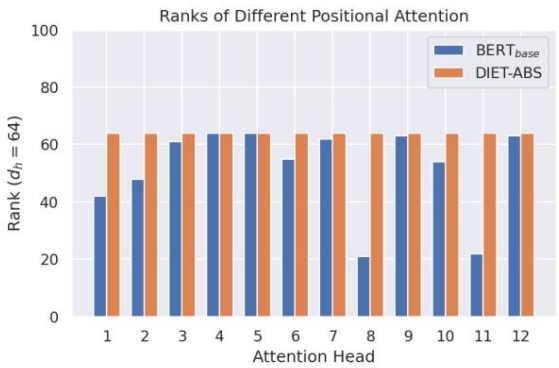


(a) Tham gia vào hàng xóm chuyển tiếp

(b) Tham dự vào các Token trư ớc đ ó

Hình 7: Mẫu đ iểm chú ý vị trí đ ư ợc lấy mẫu cho mô hình DIET-ABS . Chúng ta có thể thấy mô hình dịch chuyển rõ ràng do mô hình tạo ra. Các mẫu như vậy có thể đ ư ợc mô hình hóa tốt hơn bằng các bộ mã hóa vô hư ớng vị trí tư ơng đ ối.

Thứ hạng của Ma trận Chú ý Vị trí Trong Hình 8, chúng tôi trình bày so sánh thứ hạng của ma trận chú ý vị trí cho mô hình BERTBASE với số lần nhúng vị trí tuyệt đ ối ở đ ầu vào (PQWQW KP K) so với số lần nhúng vị trí tuyệt đ ối trên mỗi đ ầu (DIET-ABS (1), (PQP K), trong đ ó PQ, PK R n×dp). Với việc nhúng vị trí bổ sung ở đ ầu vào, ma trận chú ý vị trí có thứ hạng thấp hơn nhiều, hạn chế sức mạnh đ ại diện. Điều này đ ư ợc giảm bớt nhờ DIET-ABS.



Hình 8: Thứ hạng của ma trận chú ý vị trí

D Nghiên cứu cắt bỏ bổ sung trên GLUE

Trư ớc đ ó, chúng tôi trình bày một nghiên cứu cắt bỏ trên XTREME trong Bảng 5 đ ối với các biến thể chú ý theo vị trí đ ư ợc tách rời. Chúng tôi so sánh DIET-REL và DIET-ABS với đ ư ờng cơ sở (Devlin và cộng sự, 2018). Bây giờ chúng tôi trình bày một nghiên cứu tư ơng tự về tiêu chuẩn GLUE trong Bảng 8 và quan sát các kết quả tư ơng tự.

Mã hóa vị trí Trong Bảng 8, việc di chuyển các phần nhúng vị trí từ đ ầu vào sang mỗi đ ầu sẽ cải thiện đ iểm trung bình cho cả DIET-REL (+0,1%) và DIET-ABS (+0,2%).

Mã hóa phân đ oạn Trong Bảng 8, việc di chuyển các phần nhúng phân đ oạn từ đ ầu vào sang mỗi đ ầu sẽ cải thiện cả DIET-REL (+0,3%) và DIET-ABS (+0,05%).

Chiến lư ợc chia sẻ Chia sẻ đ óng một vai trò quan trọng đ ối với DIET-ABS. Trong Bảng 9, chúng tôi thấy rằng việc chia sẻ sẽ làm giảm hiệu suất của DIET-REL (-0,2% theo lớp, -0,3% theo đ ầu). Đối với DIET-ABS, chia sẻ giúp mô hình ổn đ ịnh hơn, có khả năng cạnh tranh với DIET-REL.

Ngư ời mẫu	Phân đ oạn vị trí	MNLI	QQP	QNLI	SST2	Cola	STS-B	393k	8.5k	Trung bình
		364k	105k		67k			7k		
Devlin và cộng sự. (2018)	đ ầu vào 85,8 / 85,9 91,1 DIET-REL						89,9	93,2	58,7	89,0	84,8
REL mỗi đ ầu	mỗi đ ầu 86,0 / 86,1 91,0 DIET-						89,8	92,8	59,6	89,0	84,9
(dp=64) bình	mỗi đ ầu 86,3 / 86,3 91,0 DIET-ABS (dp=64) mỗi đ ầu						90,5	92,9	60,3	89,3	85,2
chia sẻ) bình	quân đ ầu ngư ời bình quân đ ầu ngư ời 86,1 / 86,1 91,2 DIET-ABS (dp=64, phần						90,0	93,0	58,9	89,9	85,0
bình quân đ ầu ngư ời	bình quân đ ầu ngư ời bình quân đ ầu ngư ời 86 / 86,4 91,1 DIET-ABS (dp=128, chia sẻ)						90,2	93,0	58,9	89,8	85,0
	bình quân đ ầu ngư ời bình quân đ ầu ngư ời 86,4 / 86,4 90,8						90,4	92,9	59,3	89,8	85,2
							89,5	93,0	59,8	90,2	85,2

Bảng 8: Nghiên cứu cắt bỏ vị trí và phân đ oạn trên GLUE: DIET-REL và DIET-ABS chứng minh những ư u đ iểm của di chuyển cả nhúng vị trí và phân đ oạn từ đ ầu vào sang đ ầu ngư ời.

Ngư ời mẫu	Chia sẻ	MNLI	QQP	QNLI	SST2	Cola	STS-B	393k	8.5k	Trung bình
		364k	105k		67k			7k		
DIET-REL 86.3 / 86.3 91.0 DIET-REL	thêo lớp 86.5 / 86.3 91.1 DIET-REL						90,5	92,9	60,3	89,3	85,2
theo phần	đ ầu 85.8 / 85.7 91.2 DIET-ABS (dp=64) 86.1 / 86.1 91.2 DIET-						90,0	93,0	58,8	89,6	85,0
ABS (dp=128) 86.7 / 6,5 91,2 DIET-ABS (dp=64)	theo lớp 86 / 86,8 91,1						90,2	92,8	59,8	89,1	84,9
DIET-ABS (dp=128)	theo lớp 86,4 / 86,4 90,8						90,2	93,0	58,9	89,8	85,0
							90,6	92,8	60,1	89,4	85,3
							90,4	92,9	59,3	89,8	85,2
							89,5	93,0	59,8	90,2	85,2

Bảng 9: Chia sẻ nghiên cứu cắt bỏ trên GLUE: Chúng tôi thực hiện nghiên cứu cắt bỏ đ ể hiểu tác đ ộng của việc chia sẻ tư thế các tham số mã hóa trên các lớp và đ ầu. Chúng tôi nhận thấy rằng việc chia sẻ sẽ cải thiện hiệu suất của DIET-ABS, như ng làm tổn hại đ ến hiệu suất của DIET-REL với việc chia sẻ theo từng lớp hoặc theo từng phần.