TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Nguyễn Trương Hoàng Thái
Nguyễn Thiên Phúc

PHƯƠNG PHÁP XÁC ĐỊNH VÀ XOÁ NƠ-RON TRI THỨC TRONG MÔ HÌNH TRANSFORMER

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT
CHƯƠNG TRÌNH CHẤT LƯỢNG CAO

Tp. Hồ Chí Minh, tháng 08/2024

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Nguyễn Trương Hoàng Thái - 20127625
Nguyễn Thiên Phúc - 20127681

PHƯƠNG PHÁP XÁC ĐỊNH VÀ XOÁ NƠ-RON TRI THỨC TRONG MÔ HÌNH TRANSFORMER

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT
CHƯƠNG TRÌNH CHẤT LƯỢNG CAO

GIÁO VIÊN HƯỚNG DẪN
PGS.TS. Lê Hoàng Thái

Tp. Hồ Chí Minh, tháng 08/2024

Lời cảm ơn

Chúng em xin gửi lời tri ân sâu sắc đến quý thầy cô giáo đã nhiệt tình hướng dẫn chúng em trong quá trình thực hiện đồ án tốt nghiệp. Nhờ sự tận tâm và dìu dắt của quý thầy cô, chúng em đã hoàn thành đồ án này một cách tốt nhất.

Chúng em xin gửi lời cảm ơn chân thành đến Ban Giám hiệu nhà trường, các thầy cô giáo trong Khoa Công nghệ Thông tin đã truyền đạt kiến thức và kỹ năng cho chúng em trong suốt quá trình học tập tại Trường Đại học Khoa học Tự nhiên - ĐHQG TP.HCM.

Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc nhất tới giảng viên hướng dẫn - PGS.TS. Lê Hoàng Thái, giảng viên Bộ môn Khoa học máy tính - Trường Đại học Khoa học Tự nhiên - ĐHQG TP.HCM đã tận tình hướng dẫn và chỉ bảo chúng em trong suốt quá trình thực hiện khoá luận.

Chúng em xin gửi lời cảm ơn đến gia đình và bạn bè đã luôn động viên, hỗ trợ chúng em về mặt tinh thần và vật chất trong suốt quá trình thực hiện đồ án. Cùng với đó, chúng em xin cảm ơn các anh/chị khoá trên đã nhiệt tình cung cấp thông tin và tài liệu cho chúng em trong quá trình nghiên cứu.

Do điều kiện về thời gian cũng như lượng kiến thức về đề tài rất rộng, và kinh nghiệm còn hạn chế của chúng em, đồ án này không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo và đóng góp ý kiến của quý thầy cô để có thể bổ sung, nâng cao ý thức và kiến thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Chúng em xin trân trọng cảm ơn!

Mục lục

Danh sách hình

Danh sách bảng

Danh sách chữ viết tắt

ATTATTR phương pháp self-attention attribution.

BERT Bidirectional Encoder Representations from Transformers (mô hình học máy dựa trên cấu trúc Transformer).

DeepLIFT Deep Learning Important FeaTures.

GELU Gaussian Error Linear Unit (hàm kích hoạt được sử dụng trong mạng nơ-ron truyền thẳng).

LRP Layer-wise Relevance Propagation.

ReLU Rectified Linear Unit (hàm kích hoạt được sử dụng trong mạng nơ-ron truyền thẳng).

Tóm tắt

Những mô hình học máy nắm giữ lượng lớn dữ liệu cá nhân của người dùng, tiềm ẩn các vấn đề về bảo mật và quyền riêng tư. Quá trình xoá bỏ dữ liệu từ phía cơ sở dữ liệu không đảm bảo rằng những mô hình học máy sẽ loại bỏ hoàn toàn lượng dữ liệu. Đề tài nghiên cứu của chúng tôi được thực hiện với mục tiêu xác định và loại bỏ những tri thức được chỉ định trong mô hình ngôn ngữ dựa trên cấu trúc Transformer [22], với mục đích đảm bảo quyền riêng tư dữ liệu mà không yêu cầu tái huấn luyện mô hình học máy. Phương pháp được chúng tôi nghiên cứu tính toán điểm phân bổ áp dụng độ dốc dựa trên giá trị kích hoạt của các nơ-ron. Phương pháp được đề xuất ở nghiên cứu của Damai Dai 2022 [5], trong đó những nơ-ron có điểm phân bổ cao là những nơ-ron nắm giữ dữ liệu được chỉ định loại bỏ. Kết quả thực nghiệm thể hiện sự hiệu quả của phương pháp, nổi bật là quá trình loại bỏ dữ liệu về nghề nghiệp (quan hệ dữ liệu P106). Sau khi áp dụng phương pháp loại bỏ các nơ-ron chứa tri thức, độ đo perplexity của mô hình khi dự đoán nhãn đúng về dữ liệu này tăng lên đáng kể (+142.9%) trong khi độ đo perplexity khi dự đoán các dữ liệu khác chịu ảnh hưởng rất nhỏ (+5.8%). Tuy phương pháp thể hiện được sự hiệu quả với mục tiêu đảm bảo quyền riêng tư và tiết kiệm tài nguyên huấn luyện mô hình, song vẫn còn nhiều thách thức về phạm vi mô hình và dữ liệu thực tế. Mã nguồn của đề tài có thể được truy cập ở đường dẫn: `https://github.com/Thaifitus/determining_and_erasing_kns_in_transformer_thesis`.

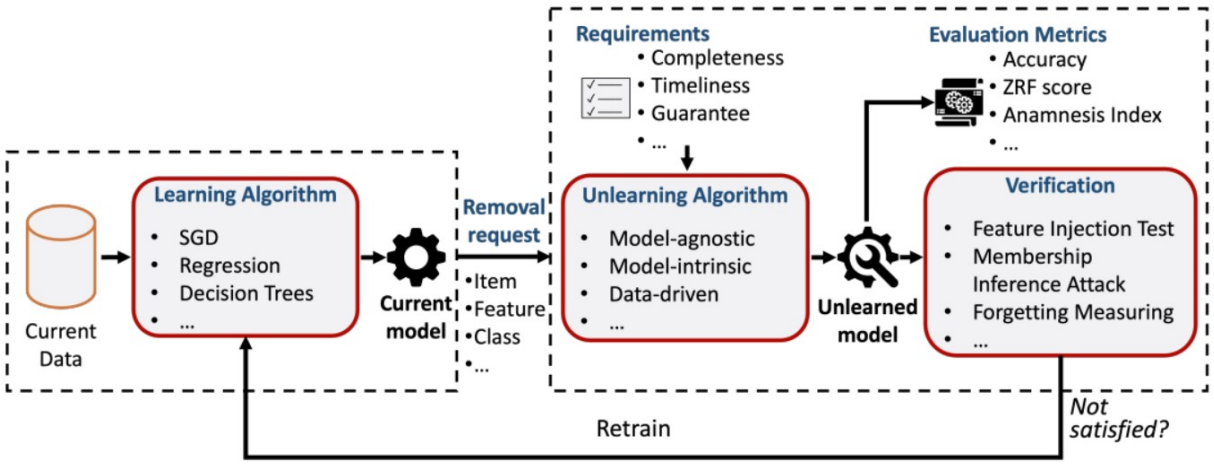
Chương 1

Giới thiệu đề tài

1.1 Đặt vấn đề

Ngày nay, những hệ thống máy tính nắm giữ lượng lớn dữ liệu cá nhân của người dùng. Bên cạnh lợi ích về mặt cải tiến những mô hình học máy còn tiềm ẩn nguy cơ làm rò rỉ dữ liệu của người dùng [20]. Quá trình xoá bỏ dữ liệu từ phía cơ sở dữ liệu không đảm bảo rằng những mô hình học máy sẽ loại bỏ hoàn toàn lượng dữ liệu dẫn đến tình trạng mô hình vẫn còn "ghi nhớ" lượng thông tin đó [16]. Các cuộc tấn công đối kháng (adversarial attack) nhắm đến việc tấn công vào dữ liệu huấn luyện của mô hình nhằm trích xuất thông tin đặc trưng của dữ liệu [4]. Tình trạng các mô hình học máy "ghi nhớ" thông tin cá nhân của người dùng tiềm ẩn những cuộc tấn công trong tương lai xa. Do đó, việc cho phép người dùng loại bỏ hoàn toàn dữ liệu riêng tư của mình khỏi mô hình học máy được gọi là lĩnh vực Machine Unlearning. Lĩnh vực này cho phép loại bỏ hoàn toàn dữ liệu khỏi mô hình học máy mà không yêu cầu quá trình huấn luyện lại mô hình. Hình ảnh 1.1 trực quan khung sườn tổng quát của phương pháp loại bỏ tri thức.

Đề tài này nghiên cứu về phương pháp xác định và xóa các nơ-ron tri thức trong mô hình Transformer dựa trên cơ chế tự chú ý (Self-Attention) và mạng nơ-ron truyền thẳng (Feed Forward Network) được đề xuất ở



Hình 1.1: Hình ảnh được tham khảo từ bài báo *A Survey of Machine Unlearning* [16]. Hình ảnh trực quan khung sườn tổng quát của phương pháp loại bỏ tri thức, bao gồm thành phần học (bên trái) và thành phần loại bỏ tri thức (bên phải). Ở thành phần học, mô hình ban đầu được huấn luyện với tập dữ liệu cụ thể sử dụng các phương pháp huấn luyện. Mô hình tiền huấn luyện sau đó được áp dụng phương pháp loại bỏ tri thức (Unlearning) với các tiêu chuẩn được sử dụng. Kết quả cuối cùng là mô hình học máy với tri thức đã loại bỏ được đánh giá và kiểm định. Mô hình sau cùng được sử dụng cho các tác vụ cụ thể nếu đáp ứng các tiêu chí kiểm định hoặc sẽ phải tái huấn luyện nếu không đáp ứng được các tiêu chí này.

nghiên cứu của Damai Dai 2022 [5]. Cụ thể, chúng tôi thực hiện tìm hiểu phương pháp giúp nhận diện các nơ-ron tri thức, từ đó có thể thực hiện thao tác xoá đi các nơ-ron tri thức này mà không cần tái huấn luyện lại mô hình. Cuối cùng, chúng tôi tiến hành thực nghiệm nhằm đánh giá và so sánh hiệu suất đối với bài toán điền từ vào ô trống. Bài toán điền từ vào ô trống (Cloze Task) [12] là một tác vụ trong Xử lý ngôn ngữ tự nhiên tập trung vào việc dự đoán từ phù hợp nhất với chỗ trống trong một câu văn dựa trên ngữ cảnh của các từ xung quanh. Kết quả nghiên cứu của đề tài thể hiện sự hiệu quả khi thực hiện loại bỏ tri thức về các quan hệ dữ liệu của mô hình, nổi bật là quan hệ nghề nghiệp (P106) - một quan hệ lưu trữ nhiều thông tin nhạy cảm của một cá nhân. Sau khi áp dụng

phương pháp loại bỏ tri thức thuộc 30 nơ-ron khác nhau với quan hệ P106, độ chính xác của mô hình khi dự đoán nhãn đúng với các câu truy vấn liên quan đến quan hệ này giảm đi đáng kể so với ban đầu (-36.13%) trong khi độ chính xác về tri thức thuộc các quan hệ còn lại chịu ảnh hưởng rất nhỏ (-1.422%). Kết quả này đáp ứng được yêu cầu đảm bảo quyền riêng tư dữ liệu và không yêu cầu tái huấn luyện mô hình học máy vì chỉ có tri thức được chỉ định bị loại bỏ khỏi mô hình, trong khi các tri thức khác không bị ảnh hưởng.

Đề tài có ý nghĩa đối với các mô hình học máy và sự riêng tư của người dùng, các ý nghĩa nổi bật có thể được đề cập như:

- Tăng cường tính bảo mật cho hệ thống: giảm nguy cơ bị tấn công bởi các cuộc tấn công đối kháng nhắm vào dữ liệu huấn luyện [4, 3].
- Tăng cường niềm tin của người dùng vào các hệ thống trí tuệ nhân tạo: khi người dùng có khả năng kiểm soát dữ liệu cá nhân, sự tin tưởng của người dùng dành cho hệ thống sẽ được cải thiện đáng kể.
- Việc áp dụng các phương pháp Machine Unlearning giúp tiết kiệm tài nguyên vì các phương pháp đảm bảo được khả năng của mô hình mà không yêu cầu quá trình tái huấn luyện.
- Thúc đẩy sự phát triển của các hệ thống trí tuệ nhân tạo có đạo đức: việc bảo vệ quyền riêng tư của người dùng là một yếu tố quan trọng trong việc phát triển các hệ thống trí tuệ nhân tạo có đạo đức [20].

1.2 Mục tiêu và phạm vi đề tài

1.2.1 Mục tiêu đề ra

Chúng tôi thực hiện đề tài nghiên cứu một phương pháp loại bỏ tri thức (phương pháp Machine Unlearning) được đề xuất ở nghiên cứu của Damai Dai 2022 [5] trong mô hình Transformer ở tác vụ xử lý ngôn ngữ

tự nhiên là điền vào ô trống [12], với mục đích đảm bảo quyền riêng tư dữ liệu mà không yêu cầu tái huấn luyện mô hình học máy.

Đây là một phương pháp trong lĩnh vực Machine Unlearning, giúp loại bỏ dữ liệu khỏi mô hình học máy mà không yêu cầu huấn luyện lại mô hình. Phương pháp ứng dụng tích hợp độ dốc được đề xuất ở nghiên cứu của Sundararajan, Taly và Yan 2017 [21] trong quá trình chọn lọc tri thức giúp chắt lọc được lượng nơ-ron tri thức chất lượng, làm tiền đề cho quá trình loại bỏ tri thức hiệu quả. Cụ thể, mục tiêu của đề tài bao gồm:

- Xác định các nơ-ron tri thức có thể hiện thực tế quan hệ (relational facts) thuộc tầng ẩn trong các mạng nơ-ron truyền thẳng ở mô hình Transformer tiền huấn luyện dựa trên phương pháp tích hợp độ dốc.
- Áp dụng phương pháp xoá bỏ tri thức về quan hệ dữ liệu đối với các nơ-ron đã xác định được.
- Thực hiện khảo sát và đánh giá hiệu quả của phương pháp đối với bài toán điền vào ô trống.

1.2.2 Phạm vi tiếp cận

Trong đề tài này, quá trình xác định nơ-ron tri thức dựa trên điểm phân bổ được giới hạn ở hai phương pháp là phương pháp cơ sở và phương pháp ứng dụng tích hợp độ dốc được đề xuất ở nghiên cứu của Damai Dai 2022 [5]. Phương pháp cơ sở sử dụng giá trị kích hoạt tương ứng với câu truy vấn đầu vào của các nơ-ron thuộc mô hình tiền huấn luyện để phục vụ cho việc tính toán. Phương pháp được Damai Dai 2022 đề xuất sử dụng kết hợp giá trị kích hoạt của các nơ-ron thuộc mô hình tiền huấn luyện với độ dốc và được tính toán sử dụng xấp xỉ Riemann. Bên cạnh đó, chiến lược chắt lọc nơ-ron cũng được thực hiện dựa trên đề xuất của Damai Dai 2022 [5]. Chiến lược này bao gồm hai bước tiến hành chính là chắt lọc theo tham số "t" - phần trăm giá trị điểm phân bổ lớn nhất của các nơ-ron với

câu truy vấn đầu vào tương ứng và tham số "p" - phần trăm số lượng câu truy vấn đầu vào thuộc mẫu câu tương ứng.

Có rất nhiều mô hình Transformer tiền huấn luyện có cấu trúc gồm hai thành phần chính là tầng tự chú ý và mạng nơ-ron truyền thẳng hoạt động với các tác vụ xử lý ngôn ngữ tự nhiên, nổi bật như Generative Pre-trained Transformer (OpenAI's GPT-1) [17], RoBERTa [13]. Với tiêu chí lựa chọn mô hình có cấu trúc phù hợp với phương pháp đồng thời có kích thước phù hợp cho quá trình thực nghiệm, mô hình được sử dụng trong đề tài là mô hình Transformer tiền huấn luyện BERT-base-cased [7].

Tập dữ liệu được sử dụng trong đề tài là tập dữ liệu PARAREL [8]. PARAREL ban đầu là một tập dữ liệu dành cho bài toán điền vào ô trống được xây dựng bởi các chuyên gia, bao gồm nhiều mẫu câu cho 38 quan hệ khác nhau. Tuy nhiên với mục đích đảm bảo tính đa dạng cho tập dữ liệu, nhóm tác giả ở bài báo của Damai Dai 2022 [5] đã phân tích và chỉ giữ lại 34 quan hệ từ tập dữ liệu ban đầu. Các quan hệ thỏa tính đa dạng là những quan hệ có nhiều hơn 3 mẫu câu, góp phần tạo ra sự thuận lợi cho phương pháp vì đảm bảo rằng các câu truy vấn không chia sẻ cùng các nơ-ron tri thức dương tính giả (false-positive). Quá trình áp dụng phương pháp cũng như kết quả thực nghiệm của đề tài được ràng buộc chặt chẽ với tập dữ liệu PARAREL đã qua xử lý này.

1.3 Đóng góp của đê tài

Đề tài nghiên cứu của chúng tôi có những kết quả đóng góp cụ thể như sau:

- Xác định được số lượng nơ-ron tri thức giao nhau giữa hai quan hệ bất kỳ thuộc tập dữ liệu là rất nhỏ (0.09 nơ-ron). Kết quả xác định này tạo tiền đề cho quá trình loại bỏ tri thức ở từng quan hệ khác nhau là độc lập nhau, giúp tiết kiệm tài nguyên tái huấn luyện mô hình.

- Quá trình loại bỏ tri thức ở mô hình BERT-base-cased về các quan hệ dữ liệu mang lại hiệu quả nhất định, mô hình trước và sau khi áp dụng phương pháp có sự thay đổi chênh lệch đối với tri thức quan hệ. Khi thực hiện loại bỏ tri thức ở 30 nơ-ron khác nhau về quan hệ nghề nghiệp (P106), cả hai số liệu về độ chính xác và độ đo perplexity của mô hình về quan hệ P106 giảm đi đáng kể lần lượt là -36.13% và +142.9%. Trong khi đó, tri thức đối với các quan hệ dữ liệu khác có sự thay đổi không đáng kể với số liệu về độ chính xác và độ đo perplexity của mô hình lần lượt là -1.422% và +5.842%.

- Quá trình loại bỏ tri thức hiệu quả đối với từng quan hệ dữ liệu giúp tăng cường tính bảo mật, quyền riêng tư và giảm thiểu tài nguyên tái huấn luyện mô hình.

- Kết quả thực nghiệm với các giá trị tham số khác nhau sau khi áp dụng phương pháp được Damai Dai 2022 [5] đề xuất.

1.4 Cấu trúc khoá luận

- Chương 1: "Giới thiệu đề tài" - Giới thiệu tổng quan về vấn đề Machine Unlearning, nội dung và các đóng góp cụ thể của đề tài ở tác vụ loại bỏ tri thức.

- Chương 2: "Các công trình liên quan" - Đề cập và bàn luận về các công trình nghiên cứu có liên quan và được tham khảo trong đề tài.

- Chương 3: "Phương pháp tìm hiểu" - Trình bày cơ sở lập luận và phương pháp được tìm hiểu ở đề tài.

- Chương 4: "Kết quả thực nghiệm" - Phân tích khám phá tập dữ liệu thực nghiệm, chi tiết cách thức thực nghiệm (mô hình, phương pháp, khảo sát) và các kết quả thống kê thu được.

- Chương 5: "Kết luận và hướng phát triển" - Nhận xét và đánh giá chi tiết về tính khả thi cũng như về ưu nhược điểm của quá trình thực nghiệm và kết quả thu được. Qua đó đề xuất hướng phát triển sau này cho đề tài.

- Tài liệu tham khảo - Mục trích dẫn các công trình nghiên cứu được tham khảo và dẫn xuất trong khóa luận.

Chương 2

Các công trình liên quan

2.1 Phương pháp tích hợp độ dốc

Các nghiên cứu trước đây tập trung vào đánh giá hiệu quả của phương pháp tính toán độ phân bổ thông qua phương pháp thực nghiệm. Phương pháp thực nghiệm trong lĩnh vực học máy là phương pháp đánh giá hoặc thử nghiệm dựa trên những quan sát, kết quả thực nghiệm hoặc kinh nghiệm thực tế thay vì dựa trên lý thuyết hoặc giả định. Tuy nhiên, các phương pháp đánh giá thực nghiệm có những hạn chế nhất định. Ví dụ: Wojciech Samek 2015 [18] đề xuất phương pháp hoạt động dựa trên sự phân bổ mức độ liên quan từ lớp đầu ra của mạng nơ-ron ngược về lớp đầu vào. Mỗi giá trị điểm ảnh (pixel) được gán một giá trị phân bổ, cho biết mức độ đóng góp của giá trị điểm ảnh đối với quyết định phân loại cuối cùng. Các điểm ảnh có giá trị liên quan cao hơn được coi là quan trọng hơn.

Để đánh giá mức độ quan trọng của các điểm ảnh, tác giả đã lựa chọn k điểm ảnh có giá trị liên quan cao nhất (ví dụ: $k = 100$) và giá trị cường độ của các điểm ảnh này sẽ được thay đổi ngẫu nhiên. Sau khi thực hiện thay đổi giá trị cường độ của các điểm ảnh dữ liệu, tác giả thực hiện đo lường mức độ giảm điểm số (độ chính xác) của mô hình trong tác vụ nhận dạng đối tượng. Nếu các điểm ảnh được chọn thực sự quan trọng, việc

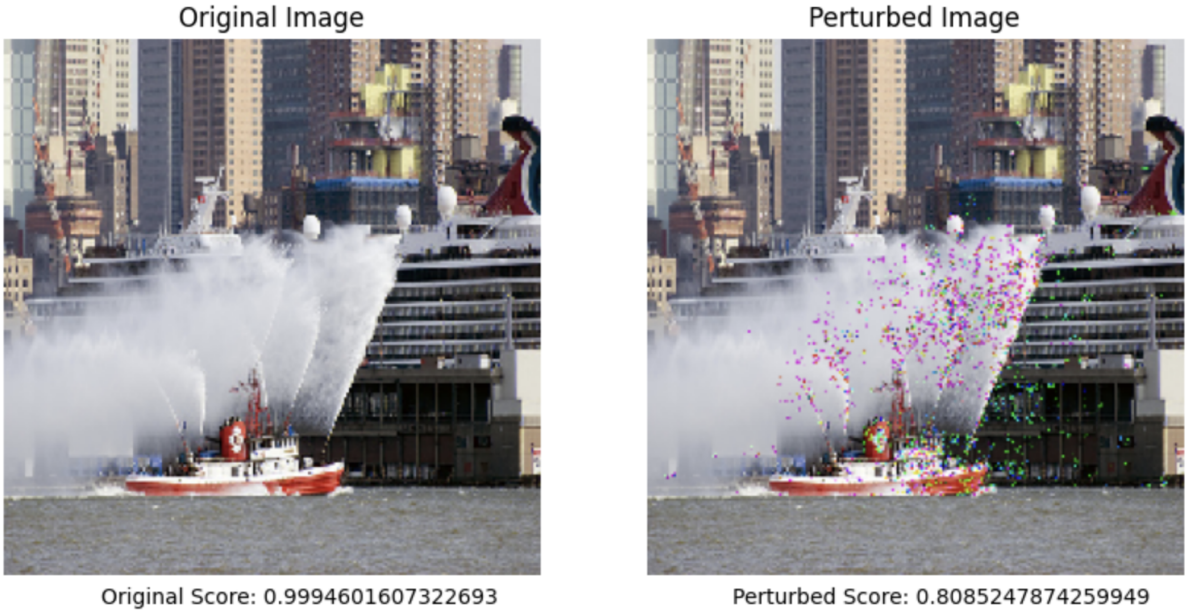
thay đổi giá trị của điểm ảnh sẽ dẫn đến sự giảm đáng kể hiệu suất của mô hình. Phương pháp được Wojciech Samek 2015 [18] trình bày cũng có những hạn chế như yêu cầu tính toán phức tạp, đặc biệt là đối với các mô hình có cấu trúc mạng nơ-ron lớn và sâu. Việc lan truyền ngược và tính toán mức độ liên quan cho từng điểm ảnh có thể tốn kém về mặt tính toán. Bên cạnh đó, hiệu quả của phương pháp phụ thuộc vào kiến trúc cụ thể của mô hình và việc áp dụng phương pháp yêu cầu tác động đến dữ liệu ban đầu. Ngoài ra, một nhược điểm có thể được đề cập của phương pháp đó là hình ảnh sau khi được áp dụng phương pháp cũng có sự thay đổi không tự nhiên như hình ảnh dữ liệu ban đầu.

Chúng tôi tiến hành áp dụng phương pháp của Wojciech Samek 2015 [18] trên bộ dữ liệu ImageNet [6] với mục đích hiểu rõ hơn về phương pháp điểm phân bổ. Bộ dữ liệu ImageNet là bộ dữ liệu quy mô lớn hình ảnh chú thích, được thiết kế để sử dụng trong nghiên cứu nhận dạng đối tượng trực quan. Bộ dữ liệu này chứa hơn 14 triệu hình ảnh, với mỗi hình ảnh được chú thích bằng bộ dữ liệu WordNet [15] - một hệ cơ sở tri thức khổng lồ về ngôn ngữ học của từ vựng tiếng Anh. Mô hình được sử dụng trong thực nghiệm là mô hình phân loại hình ảnh ResNet50. Đây là một mạng thần kinh tích chập với độ sâu 50 lớp, được Microsoft xây dựng và đào tạo vào năm 2015. Mô hình này đã được huấn luyện trên hơn 1 triệu hình ảnh từ cơ sở dữ liệu ImageNet, với các hình ảnh màu có kích thước 224×224 điểm ảnh và khả năng phân loại lên tới 1000 đối tượng khác nhau.

Trong nghiên cứu này, chúng tôi tiến hành thực nghiệm bằng cách áp dụng mô hình ResNet50 tiền huấn luyện trên tập dữ liệu ImageNet nhằm xác định các vùng quan trọng trong ảnh dữ liệu ban đầu có ảnh hưởng đến kết quả dự đoán của mô hình. Hình ảnh được chọn để thực nghiệm có nhãn tương ứng là tàu chữa cháy (fireboat) từ tập dữ liệu ImageNet. Phương pháp của Wojciech Samek (2015) [18] được áp dụng để nhận diện các điểm ảnh có giá trị phân bổ cao nhất và cường độ của các điểm ảnh này sau đó được thay đổi ngẫu nhiên để tạo ra hình ảnh mới.

Trong quá trình thực nghiệm, phương pháp của Wojciech Samek 2015

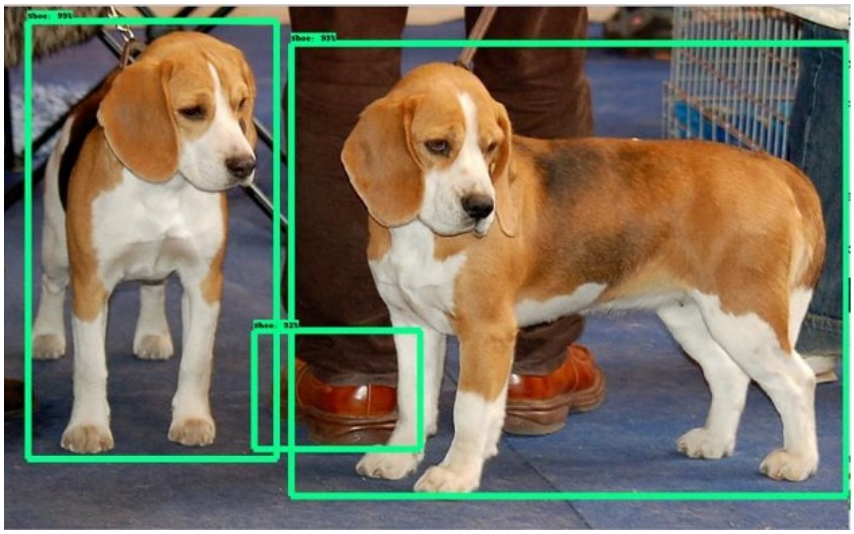
[18] tính toán điểm phân bổ cho ảnh đầu vào và từ những điểm ảnh có điểm phân bổ cao tiến hành làm nhiễu các điểm ảnh quan trọng này bằng cách thay đổi giá trị ngẫu nhiên. Sau đó, chúng tôi thực hiện đo lường sự thay đổi ở kết quả dự đoán của mô hình. Ảnh gốc và ảnh đã bị làm nhiễu được hiển thị để so sánh, cùng với điểm số dự đoán của mô hình trước và sau khi làm nhiễu. Hình 2.1 thể hiện điểm dự đoán của mô hình, hình ảnh dữ liệu trước và sau khi áp dụng phương pháp.



Hình 2.1: Hình ảnh trước (ảnh bên trái) và sau (ảnh bên phải) khi áp dụng phương pháp thay đổi ngẫu nhiên giá trị của các điểm ảnh có giá trị liên quan cao nhất. Quá trình thay đổi ngẫu nhiên các điểm ảnh quan trọng đối với kết quả dự đoán của mô hình ảnh hưởng đáng kể đến hiệu suất dự đoán (giảm $\approx 18\%$ ở ảnh bên phải) trong quá trình mô hình dự đoán dãn đúng của dữ liệu đầu vào.

Kết quả cho thấy khi các điểm ảnh quan trọng được làm nổi bật bởi phương pháp bị thay đổi, độ chính xác của mô hình giảm đi rõ rệt (giảm $\approx 18\%$, từ $\approx 99\%$ xuống còn $\approx 80.85\%$) trong quá trình mô hình dự đoán dãn đúng đã chứng minh tầm quan trọng của các điểm ảnh này. Bên cạnh đó, hình ảnh sau khi áp dụng phương pháp không được tự nhiên như ban đầu.

Một kỹ thuật đánh giá khác là sử dụng hình ảnh với khung bao quanh vật thể do người vẽ và tính toán tỷ lệ phần trăm phân bổ điểm ảnh bên trong khung được nghiên cứu của Sundararajan, Taly và Yan 2017 [21] đề cập. Mặc dù đối với hầu hết các vật thể, kết quả được kỳ vọng các điểm ảnh nằm trên vật thể là quan trọng nhất cho quá trình dự đoán nhưng trong một số trường hợp bối cảnh xung quanh vật thể cũng có thể ảnh hưởng đến dự đoán của mô hình. Hình 2.2 cho thấy hình ảnh chú chó đã bị dự đoán sai thành chiếc giày ở bên cạnh do ảnh hưởng từ bối cảnh của môi trường xung quanh.



Hình 2.2: Hình ảnh minh họa cho bài toán nhận diện vật thể. Trong trường hợp này, nhãn của đối tượng bị dự đoán sai bởi ảnh hưởng của bối cảnh xung quanh vật thể chính (chú chó).

Vì những hạn chế của các phương pháp đánh giá thực nghiệm, Sundararajan, Taly và Yan 2017 đề xuất phương pháp tiếp cận dựa trên hệ thống tiên đề (axiomatic approach).

Phương pháp tích hợp độ dốc (Integrated Gradient) là một kỹ thuật giải thích các dự đoán của mạng nơ-ron sâu (deep neural networks). Trình bày một cách tổng quát, phương pháp này phân tích cơ sở mà mô hình học máy đưa ra một quyết định cụ thể bằng cách xác định mức độ quan trọng của từng đặc trưng đầu vào đối với kết quả dự đoán của mô hình. Phương pháp kết hợp các ưu điểm của hai phương pháp khác nhau là

độ dốc (Gradients) và LRP/DeepLIFT được trình bày ở nghiên cứu của Alexander Binder 2016 [2], Avanti Shrikumar 2019 [19]. Phương pháp tích hợp độ dốc được Sundararajan, Taly và Yan 2017 [21] đề xuất dựa trên hai tiên đề là độ nhạy cảm (Sensitivity) và tính bất biến khi thực thi (Implementation Invariance):

- Độ nhạy cảm (Sensitivity): tiên đề yêu cầu rằng nếu một đầu vào và một ngưỡng cơ sở (baseline) khác nhau ở một đặc trưng nào đó và có dự đoán khác nhau, thì đặc trưng khác biệt đó phải được gán một giá trị phân bổ khác không. Điều này đảm bảo rằng các đặc trưng quan trọng được nhận diện chính xác.

- Tính bất biến khi thực thi (Implementation Invariance): tiên đề yêu cầu hai mạng học sâu có đầu ra giống nhau với tất cả các dữ liệu đầu vào, dù có các cách triển khai khác nhau thì các giá trị phân bổ là như nhau.

Phương pháp LRP do Alexander Binder [2] giới thiệu vào năm 2016 là một kỹ thuật giải thích áp dụng cho các mô hình có cấu trúc như mạng nơ-ron, trong đó đầu vào có thể là hình ảnh, video hoặc văn bản. LRP phân rã quyết định phân loại của một mạng nơ-ron sâu xuống các điểm số liên quan đến từng điểm ảnh ở dữ liệu đầu vào. Mục tiêu của phương pháp này là xác định phần dữ liệu hình ảnh quan trọng đối với quyết định phân loại của mô hình. LRP gán điểm số liên quan cho từng điểm ảnh của dữ liệu đầu vào, trong đó các điểm số này phản ánh sự đóng góp của các điểm ảnh đó vào quyết định phân loại cuối cùng. LRP hoạt động bằng cách truyền các điểm số liên quan từ lớp đầu ra về lớp đầu vào thông qua các lớp trung gian. Các điểm số này được tính toán dựa trên các trọng số và hàm kích hoạt của các nơ-ron ở mỗi lớp tương ứng thuộc mô hình, đảm bảo rằng tổng các điểm số liên quan của các lớp trước bằng tổng các điểm số liên quan của lớp sau .

Phương pháp DeepLIFT được Avanti Shrikumar [19] giới thiệu vào năm 2019 là một phương pháp tính toán nhằm xác định tầm quan trọng của các

đầu vào trong việc dự đoán đầu ra cụ thể của mô hình học sâu. Phương pháp này khắc phục các vấn đề của các phương pháp dựa trên độ dốc (gradient), như hiện tượng "bão hoà độ dốc" (gradient vanishing) và "độ dốc bùng nổ" (gradient exploding), bằng cách sử dụng sự khác biệt giữa giá trị kích hoạt so với một giá trị tham chiếu. Phương pháp thực hiện lan truyền ngược các đóng góp của tất cả các nơ-ron trong mạng của mô hình đến mọi đặc trưng của đầu vào. Khác với các phương pháp dựa trên độ dốc, DeepLIFT giải thích sự khác biệt trong đầu ra so với một đầu ra tham chiếu theo sự khác biệt của đầu vào so với đầu vào tham chiếu. Cụ thể, phương pháp này gán điểm số đóng góp cho từng đầu vào dựa trên sự thay đổi của chúng so với một giá trị đầu vào tham chiếu, đảm bảo rằng tổng các điểm số này bằng với sự thay đổi trong đầu ra. Hai phương pháp DeepLIFT và LRP đều phân tích và giải thích các quyết định của mô hình học sâu bằng cách tính toán mức độ quan trọng của từng đặc trưng đầu vào đối với dự đoán của mô hình. Cả hai phương pháp đều sử dụng kỹ thuật lan truyền ngược trong quá trình tính toán mức độ quan trọng của các đặc trưng đầu vào. Trong học máy, độ dốc là một khái niệm quan trọng được sử dụng để tối ưu hóa các mô hình. Độ dốc biểu diễn cho hướng và tốc độ thay đổi của một hàm số, giúp xác định hướng di chuyển với mục đích tìm kiếm giá trị tối ưu (cực đại hoặc cực tiểu) của hàm số. Theo nhóm tác giả [21], phương pháp này vi phạm tiên đề độ nhạy cảm mà nghiên cứu của Sundararajan, Taly và Yan 2017 đưa ra. Cụ thể, hàm dự đoán có nguy cơ xảy ra tình trạng "san bằng" (vanishing gradient) và độ dốc sẽ có giá trị bằng không mặc dù đầu vào của hàm vẫn giữ sự khác biệt so với giá trị cơ sở.

Phương pháp tích hợp độ dốc của Sundararajan, Taly và Yan 2017 [21] tính toán độ dốc tích phân dọc theo một đường thẳng từ điểm cơ sở đến điểm dữ liệu đầu vào. Công thức 2.1 mô tả cách thức hoạt động của phương pháp. Xét tập hợp đầu vào x và tập hợp cơ sở x', phương pháp tích hợp độ dốc tổng hợp các độ dốc dọc theo các đầu vào nằm trên đường thẳng giữa điểm cơ sở và điểm đầu vào theo công thức 2.1:

$$\text{IntegratedGrads}_i(x) := (x_i - x_i') \int_0^1 \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha \qquad (2.1)$$

trong đó, độ phân bổ của đặc trưng thứ i đối với đầu vào x được tính bằng $\text{IntegratedGrads}_i(x)$. Đặc trưng x_i là giá trị của đặc trưng thứ i trong đầu vào x, cần đánh giá mức độ đóng góp vào dự đoán nhãn dữ liệu của mô hình. x_i' đại diện cho giá trị của đặc trưng thứ i trong đầu vào cơ sở x' (baseline), thường là một điểm tham chiếu như một hình ảnh dữ liệu chứa những điểm ảnh có giá trị không. Tham số nội suy α thay đổi từ 0 (điểm cơ sở) đến 1 (điểm đầu vào). Hàm dự đoán F, ví dụ như hàm kích hoạt softmax, được áp dụng trong mô hình học sâu. Đạo hàm riêng $\frac{\partial F}{\partial x_i}$ đo lường độ nhạy cảm của kết quả dự đoán đối với các thay đổi trong đặc trưng thứ i của dữ liệu đầu vào. Các bước sau đây mô tả trình tự hoạt động của công thức 2.1:

Bước 1: Chọn điểm cơ sở là một điểm dữ liệu tham chiếu, thường được chọn là một điểm "trung tính" hoặc một vectơ có giá trị 0.

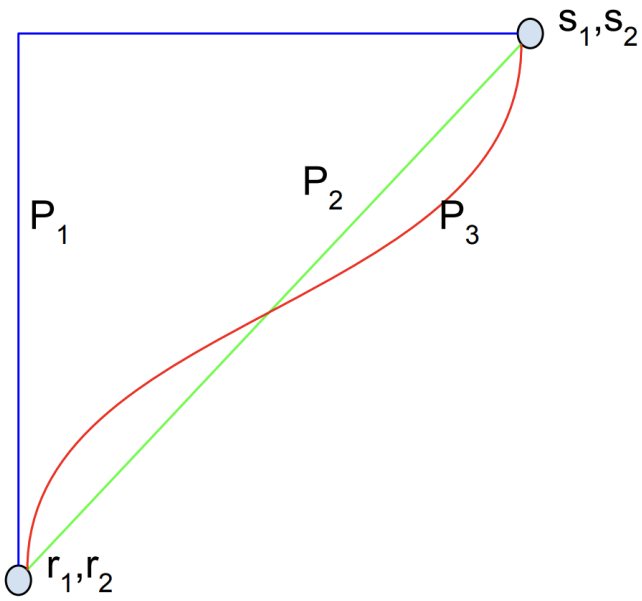
Bước 2: Một đường dẫn (đường thẳng nội suy) được tạo ra để kết nối điểm cơ sở và điểm dữ liệu đầu vào. Đường thẳng này được tham số hóa bởi α, với $\alpha = 0$ tương ứng với điểm cơ sở và $\alpha = 1$ tương ứng với điểm đầu vào.

Bước 3: Tại mỗi điểm trên đường thẳng (tương ứng với mỗi giá trị của α), giá trị độ dốc của hàm dự đoán ở mô hình (F) theo đặc trưng thứ i được tính toán. Độ dốc này cho biết mức độ thay đổi ở kết quả dự đoán của mô hình khi đặc trưng thứ i thay đổi một lượng nhỏ.

Bước 4: Các giá trị độ dốc được tích phân dọc theo đường thẳng từ $\alpha = 0$ đến $\alpha = 1$. Kết quả của tích phân này là một giá trị số thực, cho biết mức độ đóng góp tích lũy của đặc trưng thứ i khi di chuyển từ điểm cơ sở đến điểm dữ liệu đầu vào.

Bước 5: Giá trị tích hợp độ dốc được nhân với hiệu giữa giá trị của đặc trưng thứ i tại điểm đầu vào và giá trị tại điểm cơ sở. Điều này giúp đảm

bảo tính đầy đủ, nghĩa là tổng giá trị phân bổ của các đặc trưng tương đương với sự chênh lệch giữa giá trị dự đoán của mô hình tại điểm đầu vào và điểm cơ sở.



Hình 2.3: Hình ảnh được tham khảo từ bài báo *Axiomatic Attribution for Deep Networks* [21]. Hình ảnh biểu diễn đường thẳng nối giữa điểm cơ sở với điểm dữ liệu đầu vào tương ứng với ba phương pháp phân bổ khác nhau (phương pháp tích hợp độ dốc chính là đường màu xanh lá).

Có nhiều đường dẫn (không phải đường thẳng) nội suy đơn điệu giữa hai điểm và mỗi đường dẫn như vậy sẽ mang lại một phương pháp phân bổ khác nhau.

Thực nghiệm lựa chọn điểm đo (benchmark) theo Sundararajan, Taly và Yan 2017 [21] khuyến khích lựa chọn điểm cơ sở có chứa các giá trị gần với giá trị 0 đối với các bài toán xử lý văn bản áp dụng mô hình mạng nơ-ron. Trong quá trình huấn luyện, các mô hình ngôn ngữ tập trung nhiều hơn vào những từ quan trọng để dự đoán chính xác. Quá trình huấn luyện khiến các từ không quan trọng có xu hướng có chuẩn nhỏ, do đó về mặt lý thuyết thì sự không quan trọng tương ứng với mốc tham chiếu là các giá trị 0. Tính toán giá trị tích hợp độ dốc thông qua xấp xỉ Riemann, sau đó

tính tổng giá trị độ dốc tại các điểm với khoảng đủ nhỏ dọc theo đường thẳng từ đường cơ sở x' đến đầu vào x theo công thức 2.2:

$$\text{IntegratedGrads}_i^{\text{approx}}(x) := (x_i - x_i') \times \sum_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m} \tag{2.2}$$

trong đó x là đầu vào hiện tại của mô hình, và x' là điểm cơ sở (baseline) dùng để tính toán các gradient tích hợp. Giá trị của thành phần thứ i trong đầu vào hiện tại x được ký hiệu là x_i, trong khi giá trị tương ứng trong điểm cơ sở x' là x_i'. Hàm số của mô hình mạng nơ-ron mà chúng ta đang tính gradient được biểu diễn bằng F. Tích phân độ dốc của hàm số F theo thành phần x_i, tại điểm trên đoạn thẳng từ x' đến x, chia thành m đoạn, được ký hiệu là $\frac{\partial F\left(x' + \frac{k}{m} \times (x - x')\right)}{\partial x_i}$. Tham số m là số lượng đoạn chia trên đoạn thẳng từ x' đến x, và $\frac{1}{m}$ là trọng số của mỗi đoạn chia, thể hiện việc xấp xỉ của tích phân thông qua tổng Riemann.

Bài báo của Sundararajan, Taly và Yan 2017 [21] đã áp dụng phương pháp tích hợp độ dốc trên nhiều mô hình học sâu khác nhau. Với mô hình phân loại câu hỏi, phương pháp này giúp phân tích mô hình để xác định loại câu trả lời mà câu hỏi đang tìm kiếm. Phương pháp tích hợp độ dốc gán mức độ quan trọng cho từng từ trong câu hỏi đầu vào dựa trên mức độ đóng góp của từ đó trong quá trình dự đoán loại câu trả lời. Kết quả cho thấy phương pháp này xác định được các từ thường được sử dụng trong quy tắc phân loại câu hỏi và khám phá ra các quy tắc mới. Trong mô hình dịch máy, phương pháp tích hợp độ dốc gán mức độ quan trọng cho từng từ trong văn bản đầu vào, dựa trên mức độ đóng góp của từ đó trong quá trình tạo ra từng từ ở văn bản đầu ra. Ưu điểm của phương pháp tích hợp độ dốc bao gồm dễ dàng triển khai và tính toán hiệu quả, đảm bảo tính toàn diện (completeness) và tính nhạy (sensitivity), và không cần sửa đổi kiến trúc mô hình. Tuy nhiên, khuyết điểm bao gồm việc lựa chọn điểm cơ sở có thể ảnh hưởng đến kết quả và không thể giải thích được các tương tác phức tạp giữa các đặc trưng đầu vào. So sánh với các phương pháp

liên quan như DeepLIFT và LRP, phương pháp tích hợp độ dốc có nhiều ưu điểm như tính đầy đủ và tính nhạy được đảm bảo, đồng thời không yêu cầu sửa đổi kiến trúc mô hình.

2.2 Điểm phân bổ ở cơ chế tự chú ý

Nghiên cứu của Damai Dai 2022 [5] được thực hiện dựa trên bài báo đồng tác giả "Self-Attention Attribution: Interpreting Information Interactions Inside Transformer" trình bày về phương pháp điểm phân bổ tích hợp độ dốc ở tầng tự chú ý trong mô hình Transformer. Nghiên cứu về tầng tự chú ý của Yaru Hao 2021 [11] tập trung vào việc giải thích các tương tác thông tin bên trong mô hình Transformer, đặc biệt là cơ chế tự chú ý. Nghiên cứu đề xuất một phương pháp mới gọi là self-attention attribution (ATTATTR) với mục đích làm rõ cơ chế tự chú ý và được áp dụng vào mô hình ngôn ngữ BERT cơ sở. Cơ chế tự chú ý là thành phần cốt lõi tạo nên sự thành công của các mô hình học máy dựa trên cấu trúc Transformer [22]. Cơ chế này đóng vai trò quan trọng trong việc học các phụ thuộc giữa các từ thuộc văn bản dữ liệu và mã hóa thông tin ngữ cảnh đầu vào. Tuy nhiên, việc giải thích cách các đặc trưng đầu vào tương tác với nhau trong quá trình đưa ra dự đoán vẫn còn là một thách thức. Phương pháp ATTATTR hoạt động bằng cách tính toán điểm phân bổ cho từng kết nối ở quá trình chú ý. Điểm phân bổ này cho biết mức độ quan trọng của kết nối ở quá trình chú ý đó đối với quyết định dự đoán của mô hình.

Điểm phân bổ của một kết nối tự chú ý được tính toán với tích phân độ dốc của hàm mục tiêu (xác suất dự đoán của mô hình) theo trọng số chú ý của kết nối đó trên một đường đi từ ma trận chú ý với giá trị 0 đến ma trận chú ý ban đầu với giá trị tương ứng là 1. Tích phân này được tính toán dựa trên công thức 2.3:

$$\mathrm{Attr}_h(A) = A_h \odot \int_{\alpha=0}^{1} \frac{\partial F(\alpha A)}{\partial A_h} d\alpha \in \mathbb{R}^{n \times n} \tag{2.3}$$

trong đó $\text{Attr}_h(A)$ là ma trận điểm phân bổ cho đầu chú ý thứ h. Nếu đầu chú ý càng có ảnh hưởng trong quá trình dự đoán của mô hình, giá trị điểm phân bổ này sẽ càng lớn. A_h là ma trận trọng số chú ý của đầu chú ý thứ h. Giá trị α là hệ số tỷ lệ (tham số nội suy), là giá trị thực thay đổi từ 0 đến 1. $\odot$ là toán tử thể hiện cho phép nhân tương ứng từng phần tử (element-wise) giữa ma trận A_h và ma trận kết quả của tích phân. Tích phân trong công thức trên được xấp xỉ bằng tổng Riemann, với tham số m là số bước xấp xỉ cho tổng Riemann. Công thức 2.4 tính xấp xỉ Riemann cho công thức tính điểm phân bố của một kết nối chú ý. Theo Yaru Hao 2021 [11], thông thường giá trị $m = 20$ là giá trị hiệu quả trong thực tiễn.

$$\text{Attr}_h(A) = \frac{A_h}{m} \odot \sum_{k=1}^{m} \frac{\partial F(\frac{k}{m}A)}{\partial A_h} \tag{2.4}$$

Phương pháp ATTATTR được áp dụng cho mô hình BERT trên các tập dữ liệu khác nhau như MNLI, RTE, SST-2, và MRPC. ATTATTR được sử dụng để xác định các đầu chú ý quan trọng nhất thuộc mô hình BERT. Kết quả cho thấy một số đầu chú ý đóng góp nhiều hơn vào quyết định của mô hình so với các đầu chú ý khác. Các đầu chú ý có đóng góp quan trọng vào kết quả dự đoán được cắt tỉa nhằm giảm kích thước của mô hình. Kết quả cho thấy quá trình cắt tỉa làm giảm đáng kể hiệu suất của mô hình.

Phương pháp ATTATTR cung cấp một phương pháp mới để giải thích cơ chế tự chú ý trong mô hình Transformer, giúp xác định các đầu chú ý quan trọng và không quan trọng đối với quá trình dự đoán dãn dữ liệu của mô hình, từ đó có thể cắt tỉa mô hình với mục đích làm giảm kích thước mà không ảnh hưởng đáng kể đến hiệu suất dự đoán. Tuy nhiên, việc tính toán giá trị điểm phân bổ có thể tốn kém về mặt tính toán, đặc biệt là đối với các mô hình Transformer lớn. Phương pháp có nhiều điểm tương đồng với các phương pháp giải thích mô hình khác như LRP hay DeepLIFT. Phương pháp ATTATTR cũng có một số ưu điểm so với các phương pháp LRP hay DeepLIFT. Đầu tiên, phương pháp có thiết kế đặc

biệt cho mô hình Transformer, trong khi LRP và DeepLIFT là các phương pháp tổng quát hơn. Điều này cho phép ATTATTR tận dụng các đặc tính của mô hình Transformer để cung cấp các giải thích chính xác và thuyết phục. Thứ hai, phương pháp ATTATTR có thể được sử dụng để xác định và cắt tỉa các đầu chú ý không quan trọng nhằm cải thiện hiệu suất tính toán. Thứ ba, phương pháp ATTATTR có thể được sử dụng cho quá trình trực quan hóa luồng thông tin bên trong mô hình Transformer với mục đích làm rõ nguyên lý hoạt động của mô hình.

2.3 Cơ chế khóa - giá trị

Những nghiên cứu và phân tích thực nghiệm được trình bày ở bài báo của Mor Geva 2020 [10] chỉ ra rằng mô hình ngôn ngữ có cấu trúc nền tảng dựa trên mô hình Transformer [22], bao gồm hai thành phần chính là tầng tự chú ý và mạng nơ-ron truyền thẳng có liên hệ chặt chẽ với cấu trúc khóa - giá trị (key-value).

Một mạng nơ-ron truyền thẳng thuộc mô hình ngôn ngữ gồm hai thành phần chính là tầng tự chú ý và mạng nơ-ron hoạt động xử lý các vectơ dữ liệu đầu vào một cách độc lập. Mỗi vectơ đầu vào này có thể là đại diện cho các vectơ nhúng của các từ đầu vào tương ứng, hay còn gọi là thuộc tính của dữ liệu. Dựa trên nghiên cứu được Mor Geva 2020 [10] trình bày, mạng nơ-ron truyền thẳng $FF(.)$ có cấu trúc gồm một tầng ẩn và tầng đầu ra thuộc tầng con tương ứng trong mô hình có thể được biểu diễn ở biểu thức 2.5:

$$FF(x) = f(x \times K^T) \times V \tag{2.5}$$

trong đó $x \in \mathbb{R}^d$ là vectơ tương ứng với văn bản đầu vào, đây có thể là kết quả xử lý từ tầng con tự chú ý ở trước đó thuộc mô hình. $K, V \in \mathbb{R}^{d_m \times d}$ lần lượt là tham số của tầng ẩn và tầng đầu ra thuộc mạng nơ-ron truyền thẳng. Tham số d thể hiện số chiều của dữ liệu và phụ thuộc vào mô hình cụ thể, ví dụ là giá trị 768 với mô hình BERT cơ sở hay 1024 với mô hình

BERT lớn [7]. Tham số d_m là số lượng nơ-ron (hay còn gọi là số lượng ký ức) thuộc tầng ẩn và thường có giá trị gấp bốn lần giá trị của tham d. f là hàm kích hoạt phi tuyến tính được áp dụng như ReLU, GELU hay softmax.

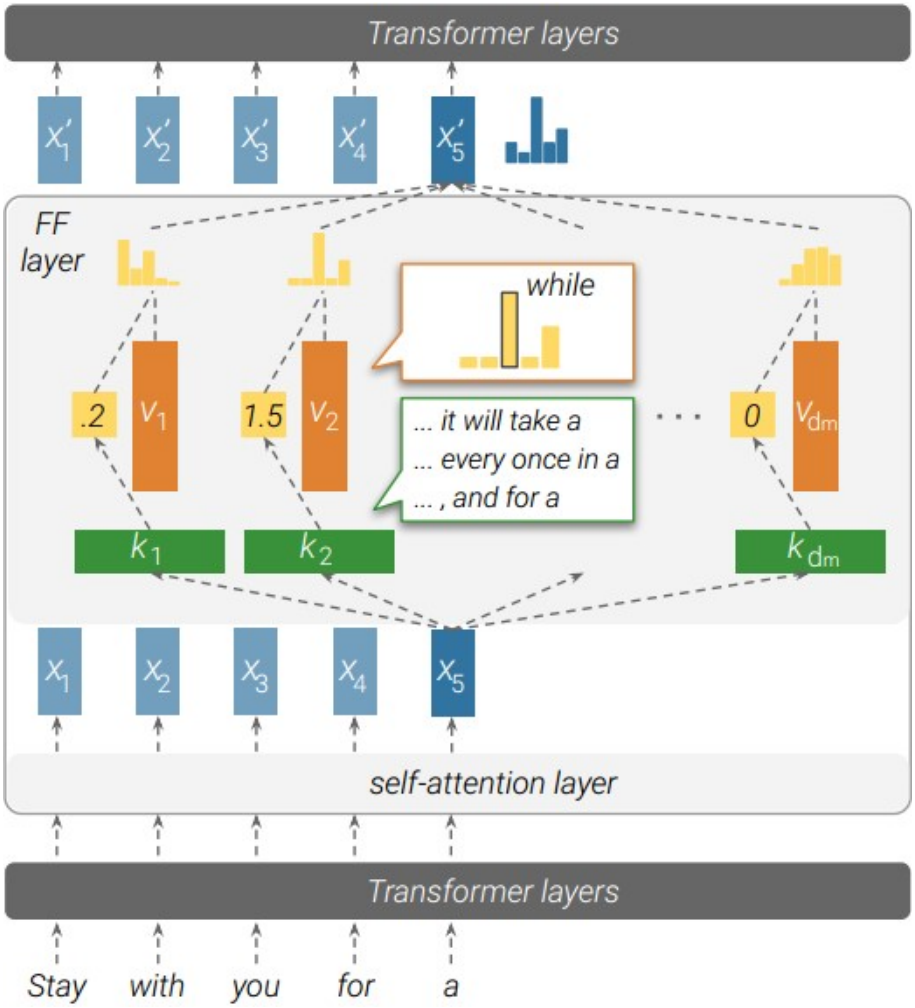
Trình bày về cấu trúc thần kinh ký ức (cấu trúc khóa - giá trị), Mor Geva 2020 [10] nêu ra rằng một cấu trúc thần kinh ký ức chứa đựng d_m cặp khóa - giá trị mà họ gọi đó là các ký ức. Sử dụng cấu trúc ma trận, nhóm tác giả biểu diễn cấu trúc thần kinh ký ức ở biểu thức 2.6:

$$MN(x) = softmax(x \times K^T) \times V \tag{2.6}$$

trong đó $x \in \mathbb{R}^d$ là vectơ dữ liệu đầu vào. Mỗi khóa được thể hiện thông qua một vectơ d chiều $k_i \in \mathbb{R}^d$ và các khóa thuộc cấu trúc mô hình kết hợp với nhau tạo nên một ma trận $K \in \mathbb{R}^{d_m \times d}$. Các phần tử giá trị có cấu trúc tương tự với các khóa tương ứng là $V \in \mathbb{R}^{d_m \times d}$.

Quan sát hai biểu thức 2.5 và 2.6 có thể nhận thấy sự tương đồng giữa mạng nơ-ron truyền thẳng của cấu trúc mô hình ngôn ngữ Transformer và cấu trúc thần kinh ký ức. Sự khác biệt duy nhất giữa hai cấu trúc đó là hàm kích hoạt được áp dụng. Cấu trúc thần kinh ký ức sử dụng hàm kích hoạt $softmax$ mà ở mạng nơ-ron truyền thẳng, hàm kích hoạt được sử dụng để khảo sát và phân tích là ReLU hay GELU với nghiên cứu của Damai Dai 2022 [5]. Bên cạnh đó, một số khái niệm được nghiên cứu của Mor Geva 2020 [10] đưa ra về các tham số d_m, $m = f(x \times K^T)$ thuộc biểu thức . Số chiều của tầng ẩn d_m thể hiện số lượng ký ức của tầng con tương ứng thuộc mạng nơ-ron truyền thẳng. Giá trị của hàm kích hoạt $m = f(x \times K^T)$ là hệ số thực không âm đại diện cho giá trị chưa được chuẩn hóa của mỗi ký ức thuộc tầng ẩn.

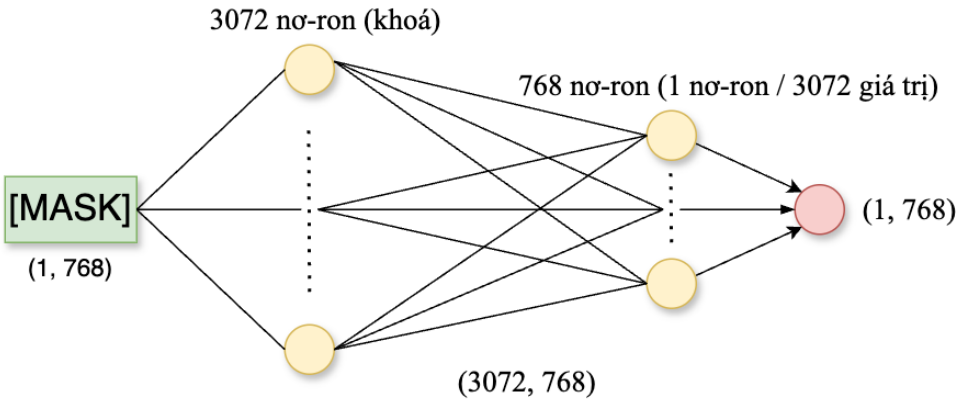
Sự tương quan giữa cấu trúc mạng nơ-ron truyền thẳng ở mô hình Transformer và cấu trúc thần kinh ký ức là cơ sở lý thuyết cho những khảo sát, phân tích thực nghiệm và nghiên cứu áp dụng phương pháp tích hợp độ dốc trong quá trình tính toán điểm phân bổ [5]. Dựa trên sự tương



Hình 2.4: Hình ảnh được tham khảo từ bài báo *Transformer Feed-Forward Layers Are Key-Value Memories* [10]. Hình ảnh thể hiện sự tương đồng của mạng nơ-ron truyền thẳng với cấu trúc thần kinh ký ức. Ở hình minh họa giá trị khóa k_2 đóng vai trò xác định mẫu câu của văn bản đầu vào (x_5) sử dụng giá trị kích hoạt, sau đó được giá trị v_2 phân phối vào kết quả dự đoán của mô hình. Nơ-ron ở tầng đầu ra (các giá trị v) thực hiện phân phối bằng cách nhân tương ứng với các khóa ($0.2 \times v_1, 1.5 \times v_2 ...$), sau đó lấy tổng các kết quả.

quan này, Mor Geva cùng các tác giả khác đã tiến hành thực nghiệm và kết luận rằng thành phần khóa ở mạng nơ-ron truyền thẳng đóng vai trò nhận dạng các mẫu câu cấu tạo nên văn bản đầu vào. Trong khi đó, các giá trị (thể hiện bởi các nơ-ron thuộc tầng đầu ra ở lớp con tương ứng) đóng vai trò phân phối giá trị của các mẫu câu đã được nhận dạng. Hình

2.4 minh họa cho kết quả nghiên cứu này.



Hình 2.5: Hình ảnh minh họa cho cấu trúc khóa - giá trị của mạng nơ-ron truyền thẳng thuộc mô hình BERT cơ sở. Dữ liệu đầu vào là từ "[MASK]" (đây chỉ là dữ liệu tượng trưng, dữ liệu thực tế được cấu tạo phức tạp với nhiều từ ngữ giúp tạo nên ngữ cảnh cụ thể cho văn bản dữ liệu đầu vào) lần lượt được tính toán ở tầng ẩn (khóa) và tầng đầu ra (giá trị). Kết quả sau khi được mạng nơ-ron xử lý (nơ-ron màu đỏ) được bảo toàn chiều dữ liệu so với dữ liệu đầu vào $(1, 768)$.

Hình 2.4 thể hiện trực quan tổng quát cho các cấu trúc của mô hình ngôn ngữ gồm hai thành phần chính là tầng con tự chú ý và mạng nơ-ron truyền thẳng dựa trên nền tảng Transformer. Một trường hợp cụ thể có thể được phân tích nhằm làm rõ trực quan này áp dụng mô hình BERT cơ sở. Sử dụng ngữ cảnh ở biểu thức 2.5 và xét mạng nơ-ron truyền thẳng thuộc một tầng con trong mô hình BERT cơ sở, mỗi nơ-ron ở tầng đầu ra nắm giữ các giá trị tương ứng với mỗi nơ-ron (khóa) ở tầng ẩn trước đó. Cụ thể, tầng ẩn chứa 3072 nơ-ron với chiều dữ liệu chi tiết là $(3072, 768)$ $(K \in \mathbb{R}^{3072\times768})$ và tầng đầu ra chứa 768 nơ-ron với chiều dữ liệu chi tiết là $(3072, 768)$ $(V \in \mathbb{R}^{3072\times768})$. Dựa trên chiều dữ liệu được trình bày, mỗi nơ-ron trong 768 nơ-ron ở tầng đầu ra nắm giữ 3072 giá trị tương ứng với mỗi nơ-ron ở tầng ẩn. Hình 2.5 thể hiện trực quan cho diễn giải này.

Các nghiên cứu và kết quả thực nghiệm của Mor Geva 2020 [10] là nền tảng lý thuyết chính để Damai Dai 2022 [5] lựa chọn tầng ẩn thuộc mạng

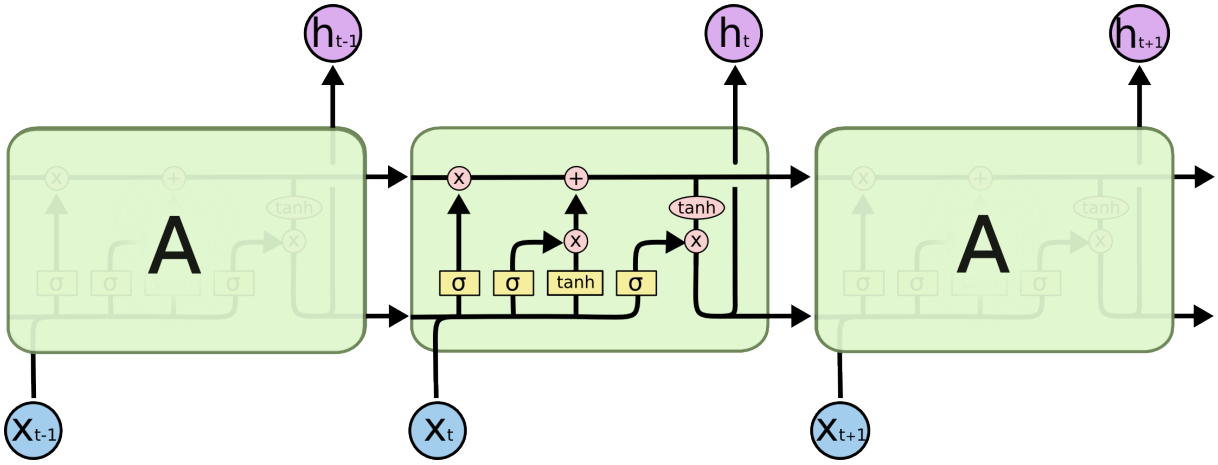
nơ-ron truyền thẳng trong mô hình ngôn ngữ Transformer cho phương pháp tính toán điểm phân bổ tích hợp độ dốc, phương pháp xóa tri thức và các thực nghiệm liên quan.

2.4 Kiến trúc Transformer

Transformer [22] là một trong những mô hình phổ biến, mang lại hiệu quả cao được áp dụng trong các bài toán xử lý ngôn ngữ tự nhiên. Mô hình Transformer ban đầu có cấu trúc gồm bộ mã hóa và bộ giải mã, loại bỏ hoàn toàn cơ chế truy hồi được áp dụng trên các mô hình trước đó như mạng nơ-ron truy hồi (Recurrent Neural Network - RNN) hay mô hình mạng trí nhớ ngắn hạn định hướng dài hạn (Long short term memory - LSTM). Sự thành công của mô hình Transformer phần lớn đến từ cơ chế tự chú ý, bên cạnh đó là mạng nơ-ron truyền thẳng.

Ưu điểm của Transformer đối với các vấn đề về xử lý ngôn ngữ tự nhiên so với các mô hình khác như mạng nơ-ron truy hồi hay mạng trí nhớ ngắn hạn định hướng dài hạn đó là quá trình xác định sự liên quan giữa các từ ngữ thuộc văn bản dữ liệu đầu vào hiệu quả với chi phí tính toán tối ưu. Ở mô hình mạng trí nhớ ngắn hạn định hướng dài hạn, chi phí tính toán dành cho các hàm kích hoạt như sigmoid hay tanh (hàm kích hoạt hyperbolic tangent) khá tốn kém và quá trình tính toán này diễn ra tuần tự (được minh họa ở hình 2.6). Trong khi đó, quá trình toán tính ở mô hình Transformer phần lớn dựa vào hai toán tử là nhân và cộng được hiện thực hóa sử dụng cấu trúc ma trận, cũng như quá trình tính toán này có thể được áp dụng phương pháp song song hóa giúp tối ưu tài nguyên và thời gian tính toán.

Ở đề tài nghiên cứu này, chúng tôi chỉ sử dụng mô hình Transformer có cấu trúc gồm bộ mã hóa. Dựa trên nghiên cứu của Damai Dai 2022 [5] và Yaru Hao 2021 [11], cấu trúc của mô hình có thể được trình bày như sau. Bộ mã hóa của mô hình Transformer bao gồm L tầng con có cấu trúc tương đương nối liền nhau (12 tầng con với mô hình BERT cơ sở). Mỗi



Hình 2.6: Hình ảnh minh họa cho cấu trúc của mô hình mạng trí nhớ ngắn hạn định hướng dài hạn. Hình ảnh được tham khảo từ trang *Khoa học dữ liệu - Khanh's blog*.

tầng con được cấu tạo với hai thành phần chính là tầng tự chú ý và mạng nơ-ron truyền thẳng gồm một tầng ẩn cùng tầng đầu ra. Hai thành phần này được biểu diễn ở biểu thức 2.8 và 2.9:

$$Q_h = XW_h^Q, K_h = XW_h^K, V_h = XW_h^V, \tag{2.7}$$

$$\text{Self-Att}_h(X) = softmax(\frac{Q_h K_h^T}{\sqrt{d_{size}}})V_h, \tag{2.8}$$

$$FFN(H) = gelu(HW_1)W_2 \tag{2.9}$$

trong đó h thể hiện chỉ số của đầu tự chú trong mô hình (giá trị $|h|$ tương ứng là số lượng đầu tự chú ý thuộc mỗi tầng con trong mô hình). $X \in \mathbb{R}^{n \times d}$ biểu diễn ma trận dữ liệu đầu vào với n chỉ số lượng từ trong văn bản đầu vào và d chỉ số chiều của dữ liệu. $W_h^Q, W_h^K \in \mathbb{R}^{d \times d_k}, W_h^V \in \mathbb{R}^{d \times d_v}$ biểu diễn ma trận trọng số của đầu tự chú ý. Tương ứng với các ma trận trọng số W là các ma trận kết quả $Q, K \in \mathbb{R}^{n \times d_k}, V \in \mathbb{R}^{n \times d_v}$. Self-Att$_h$ biểu diễn kết quả tính toán ở đầu tự chú ý tương ứng, H là kết quả tính toán từ tầng tự chú ý tương ứng trước đó. Ở đề tài này, tham số d_{size} có giá trị cụ thể là 64 tương ứng với kết quả của phép chia giữa chiều dữ liệu ($d = 768$) và

số lượng đầu tự chú ý ($|h| = 12$). Các trọng số được sử dụng ở mỗi đầu tự chú ý, ở mỗi tầng con là khác nhau mặc dù hình vi tính toán là y hệt nhau.

Biểu thức 2.8 và 2.9 thể hiện sự tương đồng giữa tầng con tự chú ý và mạng nơ-ron truyền thẳng trong mô hình Transformer mặc dù có sự khác biệt về hàm kích hoạt được sử dụng (hàm kích hoạt softmax ở tầng tự chú ý so với hàm kích hoạt GELU ở mạng nơ-ron truyền thẳng). Ma trận trọng số của tầng ẩn W_1 tương ứng với ma trận khóa K, ma trận trọng số của tầng đầu ra W_2 tương ứng với ma trận giá trị V. Dựa trên quan sát này kết hợp với nghiên cứu của Mor Geva 2020 [10] về kiến trúc khóa - giá trị cùng các nghiên cứu đã thực hiện về cấu trúc tầng tự chú ý ở bài báo đồng tác giả "*Self-Attention Attribution: Interpreting Information Interactions Inside Transformer*" [11], Damai Dai 2022 [5] tiến hành áp dụng phương pháp tích hợp độ dốc dựa trên giá trị kích hoạt ở tầng ẩn thuộc mạng nơ-ron truyền thẳng và loại bỏ tri thức. Mô hình cài đặt cụ thể cho đề tài được tùy chỉnh để phù hợp với quá trình áp dụng phương pháp và quá trình thực nghiệm.

Chương 3

Phương pháp tìm hiểu

3.1 Cơ sở dữ liệu thực nghiệm

Ở đề tài này, chúng tôi sử dụng tập dữ liệu PARAREL đã được chuẩn bị và xử lý bởi Damai Dai 2022 [5]. Tập dữ liệu PARAREL ban đầu là một tập dữ liệu cho bài toán điền vào ô trống được xây dựng bởi các chuyên gia, bao gồm nhiều mẫu câu cho 38 quan hệ dữ liệu khác nhau. Tập dữ liệu PARAREL được tạo ra với mục tiêu chính là kiểm tra tính nhất quán của các mô hình ngôn ngữ tiền huấn luyện (Pretrained Language Models - PLMs). PARAREL được sử dụng để đánh giá khả năng của các mô hình trong việc duy trì sự nhất quán khi trả lời cùng một câu hỏi tồn tại dưới nhiều hình thức khác nhau. Điều này được thực hiện bằng cách sử dụng các mẫu câu khác nhau trong quá trình truy vấn về cùng một thực tế quan hệ và so sánh các câu trả lời mà mô hình đưa ra.

Tập dữ liệu PARAREL bao gồm các kiểu quan hệ dữ liệu khác nhau, được phân loại dựa trên số lượng thực thể trong đó mỗi thực thể có liên kết dựa trên quan hệ dữ liệu. Dưới đây là ba kiểu quan hệ chính trong tập dữ liệu:

- One-to-One (1-1): Đây là các quan hệ mà mỗi thực thể chỉ liên kết với một thực thể khác. Ví dụ: quan hệ giữa một quốc gia và thủ đô của quốc gia đó (mỗi quốc gia thông thường chỉ có một thủ đô và

ngược lại).

- One-to-Many (1-N): Đây là các quan hệ mà một thực thể đầu vào có thể liên kết với nhiều thực thể khác, nhưng mỗi thực thể đầu ra chỉ liên kết với một thực thể đầu vào duy nhất. Ví dụ: quan hệ giữa một tác giả và các tác phẩm của họ (mỗi tác phẩm có thể chỉ có một tác giả, nhưng một tác giả có thể có nhiều tác phẩm).

- Many-to-Many (N-M): Đây là các quan hệ mà nhiều thực thể có thể liên kết với nhiều thực thể khác. Ví dụ: quan hệ giữa một diễn viên và các bộ phim mà họ đã tham gia (mỗi bộ phim có thể có nhiều diễn viên và mỗi diễn viên có thể tham gia nhiều bộ phim).

PARAREL bao gồm nhiều mẫu câu cho 38 quan hệ dữ liệu khác nhau, các quan hệ này được tham khảo từ tập TRex [9]. Để phù hợp với quá trình thực nghiệm, Damai Dai 2022 [5] cùng các tác giả khác đã cung cấp phần đầu cho các mẫu câu và để trống phần đuôi cho việc dự đoán. Bên cạnh đó, họ đã loại bỏ đi 4 quan hệ có số lượng mẫu câu ít hơn bốn nhằm đảm bảo tính đa dạng của tập dữ liệu. Tập dữ liệu sau cùng bao gồm 34 quan hệ khác nhau, trung bình 8.63 mẫu câu cho 1 quan hệ. Các mẫu câu này tạo ra 253,448 câu thể hiện tri thức với 27,738 thực tế quan hệ. Tập dữ liệu được sử dụng trong phần thực nghiệm được lưu trữ ở tập tin "data_all_allbag.json", cấu trúc của tập dữ liệu ở dạng từ điển với khoá là tên quan hệ, giá trị là các mẫu câu và trong mỗi mẫu câu sẽ chứa các phần tử bao gồm: câu truy vấn (prompt), từ trả lời (answer), nhãn quan hệ (rel_label). Cấu trúc tổng quan được chia thành 3 tầng chính:

- Tầng 1: Quan hệ dữ liệu.

- Tầng 2: Mẫu câu của quan hệ tương ứng.

- Tầng 3: Phần tử của từng mẫu câu (prompt, answer, rel_label).

3.2 Tổng quan phương pháp tiếp cận

Phương pháp tích hợp độ dốc (Integrated Gradients) được giới thiệu bởi Sundararajan, Taly và Yan 2017 [21] là một phương pháp nhằm giải quyết thách thức trong việc hiểu cách mô hình học sâu đưa ra dự đoán. Bằng cách tính tích phân độ dốc của đầu ra theo đầu vào từ một điểm cơ sở đến đầu vào thực tế, phương pháp tích hợp độ dốc giúp xác định các đặc trưng quan trọng nhất trong quyết định của mô hình. Trong bối cảnh các mô hình ngôn ngữ tiền huấn luyện quy mô lớn như BERT, phương pháp này đặc biệt hữu ích trong việc xác định các nơ-ron lưu trữ thông tin thực tế hay còn được biết đến là các nơ-ron tri thức. Bằng cách áp dụng phương pháp tích hợp độ dốc lên các nhiệm vụ điền vào chỗ trống, chúng ta có thể giải thích các dự đoán của mô hình dựa vào các nơ-ron cụ thể trong các lớp mạng nơ-ron truyền thẳng của mô hình Transformer. Phân tích này không chỉ giúp hiểu rõ hơn về cách mô hình lưu trữ và biểu đạt thông tin thực tế mà còn mở ra các hướng can thiệp thực tế vào hành vi của mô hình, đảm bảo tính minh bạch và khả năng điều chỉnh kiến thức của mô hình mà không cần huấn luyện lại toàn bộ mô hình.

Phương pháp tích hợp độ dốc dựa trên điểm kích hoạt của các nơ-ron [5] cho phép đo lường sự đóng góp của từng đặc trưng đầu vào đối với đầu ra của mô hình. Điều này giúp xác định chính xác những đặc trưng nào quan trọng nhất trong việc đưa ra dự đoán, tạo ra một cái nhìn rõ ràng và trực quan về cách thức mô hình hoạt động. Kết quả xác định tri thức chính xác còn là tiền đề cho quá trình loại bỏ tri thức hiệu quả, đem lại ý nghĩa về quyền riêng tư và tiết kiệm tài nguyên tái huấn luyện mô hình. Một trong những lợi thế quan trọng của phương pháp tích hợp độ dốc là đáp ứng được hai tiêu chí quan trọng: tiền đề nhạy cảm (sensitivity) và tiền đề tính bất biến khi triển khai (implementation invariance). Điều này cho thấy rằng phương pháp tích hợp độ dốc có thể đảm bảo các đặc trưng quan trọng thực sự được phản ánh ở các kết quả tích phân. Hơn nữa, phương pháp này dễ dàng áp dụng với các mô hình hiện có mà không yêu

cầu thay đổi dữ liệu hay cấu trúc mô hình. Về chi phí tính toán, quá trình tính toán tích phân trong phương pháp tích hợp độ dốc có thể tốn kém về mặt tính toán và thời gian, đặc biệt là đối với các mô hình lớn và tập dữ liệu lớn. Để giải quyết vấn đề này, phương pháp này phải sử dụng xấp xỉ Riemann để xấp xỉ các tích phân.

3.3 Điểm phân bổ tri thức

Mục này trình bày cơ sở lập luận và phương pháp tính toán điểm phân bổ cho các nơ-ron thuộc mạng nơ-ron truyền thẳng, bao gồm hai mục con: (1) Nghiên cứu về vùng nhớ khóa - giá trị ở mô hình Transformer của Mor Geva 2021 [10], (2) Biểu thức tính toán điểm phân bổ với phương pháp tích hợp độ dốc ở nghiên cứu của Damai Dai 2022 [5].

3.3.1 Vùng nhớ khóa - giá trị ở mô hình Transformer

Với mục đích nghiên cứu về tính chất của mạng nơ-ron truyền thẳng trong mô hình ngôn ngữ Transformer, Mor Geva 2021 [10] đã tiến hành các phương thức khảo sát và đánh giá dựa trên cấu trúc khóa - giá trị được trình bày ở biểu thức 2.5 và 2.6. Kết quả thực nghiệm cho thấy rằng các khóa K tương ứng ở biểu thức 2.5 nắm vai trò như thành phần xác định mẫu câu cấu tạo nên các tiền tố (prefixes) từ văn bản dữ liệu đầu vào, trong khi đó các giá trị V thực hiện phân phối giá trị của các mẫu câu được xác định. Quá trình khảo sát các nơ-ron ở tầng ẩn (các khóa) bao gồm hai bước chính (1) Dựa vào giá trị kích hoạt của nơ-ron sử dụng tập dữ liệu huấn luyện, nhóm tác giả chọn ra *top-t* tiền tố tạo ra giá trị kích hoạt cao nhất (2) Nhóm tác giả yêu cầu người tham gia (những người có kiến thức về xử lý ngôn ngữ tự nhiên) tiến hành xác định, mô tả và phân loại các mẫu câu dựa trên *top-t* tiền tố được chọn lọc trước đó. Mỗi khóa và t tiền tố tương ứng được xử lý bởi một người tham gia.

Mô hình được Mor Geva 2021 [10] sử dụng là mô hình tiền huấn luyện được giới thiệu ở nghiên cứu của Baevski và Auli 2019 [1]. Phần lớn cấu trúc của mô hình dựa trên mô hình Transformer nguyên bản [22] nhưng chỉ bao gồm bộ giải mã. Baevski và Auli 2019 [1] sử dụng hàm *sin* trong quá trình tính toán nhúng vị trí ở tầng đầu vào với mô hình bao gồm 16 tầng Transformer, mỗi tầng Transformer được cấu tạo từ hai tầng con bao gồm 16 đầu tự chú ý và mạng nơ-ron truyền thẳng. Mạng nơ-ron truyền thẳng có cấu tạo được biểu diễn với biểu thức $ReLU(W_1X + b_1)W_2 + b_2$ trong đó $W_1 \in \mathbb{R}^{d \times d_m}$, $W_2 \in \mathbb{R}^{d_m \times d}$ với $d = 1024, d_m = 4096$. Mô hình chứa tổng cộng $d_m \times 16 = 65,536$ khóa có thể được khảo sát và nhóm tác giả chỉ chọn ra ngẫu nhiên 10 khóa ở mỗi tầng Transformer (160 khóa được khảo sát). Dữ liệu được sử dụng để huấn luyện mô hình là tập dữ liệu WikiText-103 [14]. WikiText-103 là tập dữ liệu về bài toán dự đoán từ tương ứng với các ô trống thuộc văn bản dữ liệu đầu vào, bao gồm hơn 100 triệu từ được tổng hợp dựa trên các bài viết đã được kiểm định từ Wikipedia với số lượng từ ngữ độc lập lên đến 267,735 từ thuộc bộ từ điển. Ở thời điểm trước đây tập dữ liệu này có thể được truy cập thông qua Amazon S3 hoặc trang lưu trữ mà tác giả cung cấp nhưng tại thời điểm khóa luận này được thực hiện, tập dữ liệu WikiText-103 có thể được truy cập thông qua trang Hugging Face với từ khóa *"Salesforce/wikitext"*.

Mor Geva 2021 [10] giả định rằng ký ức được lưu trữ ở mạng nơ-ron đến từ dữ liệu huấn luyện mô hình. Đối với một khóa (nơ-ron) k_i^l tương ứng với khóa có chỉ số i thuộc tầng ẩn trong mạng nơ-ron ở tầng con l, các giá trị kích hoạt được tính toán với biểu thức $ReLU(x_j^l \times k_i^l)$ với các tiền tố tương ứng $x_1, ..., x_j$ được tạo thành từ mỗi câu đầu vào. Ví dụ với câu *"The game began development"* sẽ tạo thành bốn tiền tố tương ứng là *"The"*, *"The game"*, *"The game began"*, "*The game began development*" và kết quả tính toán thu được là bốn giá trị kích hoạt (hệ số ký ức) tương ứng với mỗi tiền tố. Sau khi tính toán, nhóm tác giả chọn ra t tiền tố có kết quả lớn nhất tương ứng với mỗi khóa. Kết quả này là giá trị kích hoạt của nơ-ron ở tầng ẩn hay còn được chọn làm giá trị cơ sở ở nghiên cứu của

Damai Dai 2022 [5].

1	It requires players to press
1	The video begins at a press
1	The first player would press
1	Ivy, disguised as her former self, interrupts a Wayne Enterprises press
1	The video then cuts back to the press
1	The player is able to press
	Leto switched
1	In the Nintendo DS version, the player can choose to press
1	In-house engineer Nick Robbins said Shields made it clear from the outset that he (Robbins) “was just there to press
1	She decides not to press
1	she decides not to press
1	Originally Watson signaled electronically, but show staff requested that it press
1	At post-game press
1	In the buildup to the game, the press
2	Hard to go back to the game after that news
1	In post-trailer interviews, Bungie staff members told gaming press
	Space Gun was well received by the video game
1	As Bong Load struggled to press
	At Michigan, Clancy started as a quarterback, switched
1	Crush used his size advantage to perform a Gorilla press
1,2	Groening told the press
1	Creative director Gregoire <unk> argued that existing dance games were merely instructing players to press
1,2	Mattingly would be named most outstanding player that year by the press
1	At the post-match press
1,2	The company receives bad press

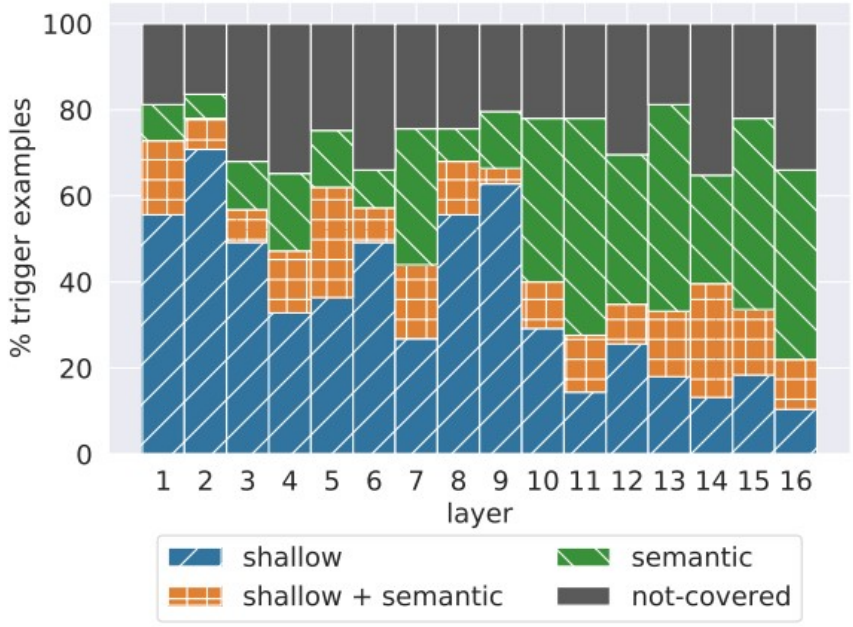
ID	Description	shallow / semantic
1	Ends with the word “press”	shallow
2	Press/news related	semantic

Hình 3.1: Hình ảnh được tham khảo từ bài báo *Transformer Feed-Forward Layers Are Key-Value Memories* [10]. Hình ảnh kết quả chú thích cho khóa k^5_{895}. Mỗi tiền tố được xác định mẫu câu với chỉ số “1” thể hiện mẫu câu cấu tạo nên tiền tố “kết thúc với từ press”, chỉ số “2” thể hiện mẫu câu có liên quan đến từ “Press” hoặc “news”. Một số tiền tố không xác định mẫu câu vì tiền tố đó được cấu tạo không dựa trên mẫu câu “1” hoặc “2”. Cả hai mẫu câu được ghi nhận đều xuất hiện trong ít nhất bốn tiền tố.

Mor Geva 2021 [10] sau đó yêu cầu những người tham gia chú thích cho 25 tiền tố đạt giá trị lớn nhất tương ứng với mỗi khóa. Quá trình chú thích bao gồm ba bước (1) Xác định các mẫu câu xuất hiện ở ít nhất ba tiền tố. Điều này giúp nhận diện sự liên kết giữa dữ liệu và khóa sở hữu tri thức về dữ liệu đó. (2) Mô tả bằng lời văn về các mẫu câu đã xác định (3) Gán nhãn cho mẫu câu. Nhãn dữ liệu bao gồm hai giá trị là *“shallow”* biểu diễn cấu trúc rập khuôn của mẫu câu và *“semantic”* nếu mẫu câu mang một chủ đề, ngữ nghĩa nào đó. Kết quả chú thích hoàn chỉnh cho một khóa được biểu diễn ở hình 3.1.

Quá trình chú thích mang lại kết quả đáng kỳ vọng. Người tham gia có thể xác định được ít nhất một mẫu câu với khóa tương ứng. Phần lớn

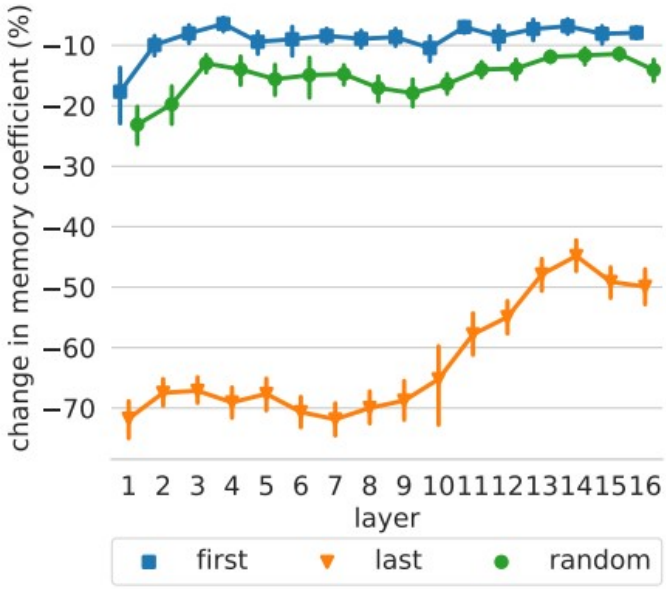
các tiền tố (65%-80%) đều được cấu tạo từ ít nhất một mẫu câu. Hình 3.2 thể hiện trực quan kết quả chú thích các tiền tố tương ứng với các khóa ở mỗi tầng Transformer. Kết quả trực quan ở hình 3.2 cho thấy rằng các tầng thấp (tầng 1 đến tầng 9) chiếm đa số với kết quả nhận dạng các mẫu câu thuộc nhãn "shallow". Trong khi đó, các tầng cao (tầng 10 đến tầng 16) lại có xu hướng nhận dạng các mẫu câu thuộc nhãn "semantic". Kết quả này cho thấy rằng với quá trình xử lý sâu, các khóa có khả năng nhận thức rõ hơn về mặt ngữ nghĩa của dữ liệu. Nhìn chung quá trình sử dụng giá trị kích hoạt của các nơ-ron có thể xác định được tính chất của tầng ẩn thuộc mạng nơ-ron truyền thẳng, cũng như là giá trị cơ sở cho phương pháp tích hợp độ dốc [5].



Hình 3.2: Hình ảnh được tham khảo từ bài báo *Transformer Feed-Forward Layers Are Key-Value Memories* [10]. Hình ảnh trực quan kết quả chú thích các tiền tố ở mỗi tầng Transformer.

Nhằm kiểm định rõ hơn về kết quả này, Mor Geva 2021 [10] còn tiến hành khảo sát với quá trình thực hiện như sau. Nhóm tác giả chọn ra ngẫu nhiên 100 khóa ở mỗi tầng Transformer (1600 khóa được khảo sát) và tiến hành chỉnh sửa *top*-50 tiền tố tương ứng với mỗi khóa. Quá trình chỉnh sửa sẽ loại bỏ một từ ở một trong ba vị trí đầu, đuôi hoặc ngẫu nhiên thuộc

tiền tố tương ứng. Sau khi chỉnh sửa các tiền tố đầu vào, nhóm tác giả tiến hành đo đạc giá trị kích hoạt của các nơ-ron nhằm khảo sát sự ảnh hưởng của quá trình chỉnh sửa. Hình 3.3 trực quan sự thay đổi của giá trị kích hoạt sau khi tiến hành chỉnh sửa các tiền tố đầu vào.



Hình 3.3: Hình ảnh được tham khảo từ bài báo *Transformer Feed-Forward Layers Are Key-Value Memories* [10]. Hình ảnh trực quan sự thay đổi của giá trị kích hoạt sau khi tiến hành loại bỏ một từ ở một trong các vị trí đầu (first), đuôi (last) hoặc ngẫu nhiên (random) thuộc các tiền tố đầu vào tương ứng với mỗi khóa ở các tầng Transformer.

Dựa vào kết quả trực quan ở hình 3.3 có thể thấy rằng giá trị kích hoạt của các nơ-ron ở tầng ẩn bị ảnh hưởng nhiều hơn bởi từ ở vị trí cuối của tiền tố. Điều này nghĩa là mô hình thường dựa vào các từ ở vị trí cuối trong quá trình dự đoán kết quả hơn là các từ ở vị trí đầu. Kết quả còn thể hiện sự phụ thuộc của các từ ở vị trí cuối đối với các tầng trong mô hình. Các tầng cao (tầng 10 đến tầng 16) ít bị ảnh hưởng bởi các từ ở vị trí cuối. Điều này có sự tương đồng với kết quả ở hình 3.2, biểu hiện rằng các tầng cao có xu hướng nhận dạng văn bản đầu vào dựa trên ngữ nghĩa thay vì biểu mẫu rập khuôn.

Nghiên cứu của Mor Geva 2021 [10] chứng minh rằng tri thức ở mạng nơ-ron truyền thẳng thuộc các tầng Transformer tập trung ở tầng ẩn và

các tầng cao có khả năng nhận dạng ngữ nghĩa tốt hơn. Đây là cơ sở lập luận để Damai Dai 2022 [5] thực hiện phương pháp xóa tri thức được đề tài nghiên cứu và trình bày lại.

3.3.2 Điểm phân bổ với phương pháp tích hợp độ dốc

$$\text{Attr}(w_i^{(l)}) = w_i^{(l)} \int_{\alpha=0}^{1} \frac{\partial P_x(\alpha \overline{w_i}^{(l)})}{\partial w_i^{(l)}} d\alpha \tag{3.1}$$

Công thức 3.1 tính toán điểm phân bổ cho một nơ-ron tri thức trong mô hình ngôn ngữ gồm hai thành phần chính là tầng tự chú ý và mạng nơ-ron truyền thẳng dựa trên cấu trúc Transformer. Trong đó, $w_i^{(l)}$ là giá trị kích hoạt của nơ-ron thứ i thuộc tầng ẩn l trong mô hình (đối với mô hình BERT l có giá trị từ 0 đến 11, i có giá trị từ 0 đến 3071). $P_x(\alpha \overline{w_i}^{(l)})$ thể hiện xác suất dự đoán nhãn đầu ra tương ứng với nhãn đúng của mô hình và α là tham số nội suy, có giá trị thay đổi từ 0 (điểm tham chiếu) đến 1 (điểm đầu vào). $\text{Attr}(w_i^{(l)})$ là điểm phân bổ cho một nơ-ron, có giá trị tỉ lệ thuận với lượng tri thức mà nơ-ron nắm giữ. Các bước sau đây mô tả trình tự hoạt động của công thức 3.1:

Bước 1: chọn điểm tham chiếu (baseline). Điểm tham chiếu thường là một điểm trung lập, không mang thông tin liên quan đến nhiệm vụ của mô hình. Trong trường hợp này, điểm tham chiếu được lựa chọn là giá trị 0, tương ứng với việc không có sự kích hoạt của các nơ-ron trong mô hình.

Bước 2: tạo một đường dẫn từ điểm tham chiếu đến điểm đầu vào. Đường dẫn này thường là một đường thẳng trong không gian vectơ biểu diễn các nơ-ron.

Bước 3: tính toán độ dốc của đầu ra mô hình (xác suất trả lời chính xác) đối với từng nơ-ron tại nhiều điểm trên đường dẫn. Giá trị độ dốc cho biết mức độ thay đổi của kết quả dự đoán đầu ra khi có sự thay đổi nhỏ ở các nơ-ron.

Bước 4: tính tổng giá trị tích phân độ dốc đã được tính toán nhằm thu

được điểm phân bổ cho mỗi nơ-ron tương ứng.

Bước 5: quá trình tính toán trực tiếp tích phân độ dốc trực tiếp gặp nhiều khó khăn trong thực tế, tác giả Damai Dai 2022 [5] đã sử dụng xấp xỉ Riemann vào quá trình tổng hợp độ dốc tại các điểm rời rạc trên đường dẫn. Việc tính tích phân bằng phương pháp xấp xỉ Riemann thu được điểm phân bổ cho mỗi nơ-ron, cho biết mức độ đóng góp của từng nơ-ron vào dự đoán của mô hình. Xấp xỉ Riemann được biểu diễn ở công thức 3.2 với m là số bước xấp xỉ tích phân:

$$\tilde{\text{Attr}}(w_i^{(l)}) = \frac{\overline{w_i}^{(l)}}{m} \sum_{k=1}^{m} \frac{\partial P_x\left(\frac{k}{m}\overline{w_i}^{(l)}\right)}{\partial w_i^{(l)}} \tag{3.2}$$

Ở nghiên cứu của Damai Dai 2022 [5] và đề tài nghiên cứu của chúng tôi, giá trị của tham số m được lựa chọn trong quá trình thực nghiệm là 20.

3.4 Chắt lọc nơ-ron tri thức

Với mục đích lọc ra những nơ-ron tri thức thuộc trường hợp "sai-dương" ở kết quả tính toán điểm phân bổ tích hợp độ dốc (những nơ-ron này thể hiện tri thức khác với kỳ vọng ở tập dữ liệu về mặt cú pháp, từ vựng...), Damai Dai 2022 [5] đề xuất một phương pháp chắt lọc nơ-ron tri thức. Để phương pháp hoạt động hiệu quả, tập dữ liệu PARAREL cần đảm bảo sự đa dạng với các quan hệ dữ liệu có số lượng mẫu câu nhiều hơn 3 mẫu câu. Phương pháp gồm ba bước thực hiện chính: (1) đối với mỗi câu truy vấn trong tập dữ liệu, tính toán điểm phân bổ của tất cả nơ-ron thuộc tầng ẩn trong mô hình; (2) đối với mỗi câu truy vấn, tiến hành chắt lọc với ngưỡng t sử dụng giá trị điểm phân bổ của các nơ-ron. Những nơ-ron có điểm phân bổ lớn hơn ngưỡng t được cân nhắc thuộc tập chắt lọc thô; (3) tiến hành chắt lọc sử dụng ngưỡng p đối với các nơ-ron thuộc tập thô ở phạm vi mẫu câu. Những nơ-ron được chia sẻ bởi số lượng câu truy vấn

lớn hơn ngưỡng p được cân nhắc là nơ-ron tri thức và được lựa chọn ở quá trình loại bỏ tri thức trong mô hình.

3.5 Bàn luận

Nghiên cứu này đã sử dụng phương pháp tích hợp độ dốc để xác định các nơ-ron lưu trữ thông tin tri thức trong mạng nơ-ron truyền thẳng của mô hình ngôn ngữ tiền huấn luyện, cụ thể là BERT. Kết quả thực nghiệm cho thấy rằng phương pháp này không chỉ hiệu quả trong việc xác định các nơ-ron quan trọng mà còn cung cấp cái nhìn chuyên sâu về cách mô hình lưu trữ và biểu đạt thông tin tri thức.

Mặc dù phương pháp tích hợp độ dốc đã chứng minh được hiệu quả trong việc xác định các nơ-ron lưu trữ thông tin tri thức, nhưng nghiên cứu này vẫn tồn tại một số hạn chế. Thứ nhất, phương pháp tích hợp độ dốc đòi hỏi một lượng lớn tài nguyên tính toán, đặc biệt là khi áp dụng cho các mô hình lớn và tập dữ liệu lớn. Việc tính toán tích phân độ dốc có thể tốn kém về mặt tính toán và thời gian, gây ra khó khăn trong việc áp dụng phương pháp này cho các mô hình và tập dữ liệu lớn hơn. Thứ hai, mặc dù kết quả cho thấy rằng các nơ-ron trong mạng nơ-ron truyền thẳng của mô hình BERT có khả năng lưu trữ thông tin tri thức, nhưng việc xác định chính xác những thông tin tri thức nào được lưu trữ và cách chúng được biểu đạt vẫn là một thách thức. Các nơ-ron có thể lưu trữ các thông tin phức tạp và đa chiều, do đó việc giải thích và phân tích chi tiết các thông tin này đòi hỏi sự can thiệp của các phương pháp phân tích phức tạp hơn.

Chương 4

Kết quả thực nghiệm

4.1 Môi trường thực nghiệm

4.1.1 Tập dữ liệu thực nghiệm

Để phân tích dữ liệu của tập PARAREL, chúng tôi đã thực hiện các bước phân tích chi tiết cho hai tập tin *data_ all.json* và *data_ all_ allbags.json*. Mục đích của việc chia thành hai tập tin dữ liệu như sau:

- data_all.json: Tập dữ liệu này chứa thông tin chi tiết về từng mẫu câu và quan hệ dữ liệu cụ thể. Điều này giúp cho việc phân tích từng phần tử, thống kê số lượng mẫu câu trong mỗi quan hệ và đánh giá các mẫu câu theo từng quan hệ một cách chi tiết. Tập dữ liệu phục vụ cho mục tiêu nghiên cứu chuyên sâu về từng quan hệ và mẫu câu trong dữ liệu.

- data_all_allbags.json: Tập dữ liệu này chứa thông tin tổng hợp theo các bao (bags) áp dụng kiểu dữ liệu từ điển, với khoá là tên các quan hệ dữ liệu và giá trị chính là các mẫu câu thuộc vào quan hệ tương ứng. Việc phân tích trên cấp độ bao giúp dễ dàng nhận diện các xu hướng lớn, phân bổ dữ liệu theo các nhóm chính và giúp việc đánh giá dữ liệu ở một cấp độ tổng quan hơn. Tập dữ liệu này được sử

dụng ở quá trình áp dụng phương pháp loại bỏ tri thức và các khảo sát liên quan.

Chúng tôi tiến hành khám phá tập dữ liệu PARAREL theo quy trình gồm 4 bước chính: (1) Cài đặt các thư viện cần thiết cho quá trình phân tích, (2) Tải tập dữ liệu thô lên và xử lý đưa về dạng DataFrame, (3) Thực hiện tiền xử lý và khám phá dữ liệu, (4) Thống kê số liệu phân tích trên tập dữ liệu, sau đó tiến hành nhận xét.

Phân tích khám phá cho tập dữ liệu data_all.json

Dữ liệu được đọc từ tập tin *data_all.json* và chuyển đổi thành một DataFrame để thuận tiện cho việc xử lý và phân tích. Cấu trúc dữ liệu sau khi biến đổi bao gồm 3 cột chính: tên quan hệ dữ liệu (rel_label), các câu truy vấn (prompt) và nhãn trả lời tương ứng (answer). Chúng tôi triển khai một hàm để "làm phẳng" dữ liệu, chuyển đổi các cấu trúc lồng nhau thành một cấu trúc ở dạng bảng dữ liệu đơn giản hơn. Sau đó, các bước tiền xử lý bao gồm loại bỏ các giá trị null hoặc không hợp lệ để đảm bảo tính toàn vẹn của dữ liệu. Chúng tôi thống kê và đếm số lượng các mẫu câu (template) theo từng quan hệ dữ liệu, sau đó trình bày dưới dạng bảng dữ liệu và biểu đồ để dễ dàng nhận diện các đặc điểm và xu hướng của dữ liệu.

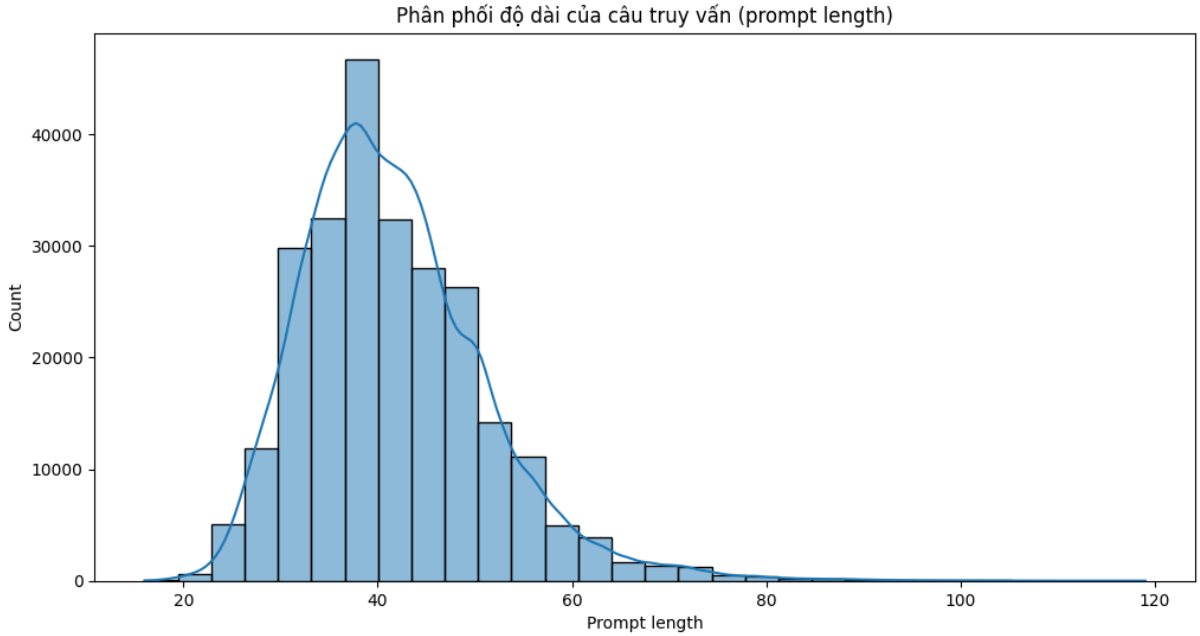
Theo như mô tả của tác giả, họ đã loại bỏ đi 4 quan hệ và giữ lại 34 quan hệ từ tập dữ liệu TRex. Số lượng câu trả lời ít hơn số lượng câu truy vấn bởi vì một câu trả lời có thể dùng để trả lời cho nhiều câu truy vấn (dựa vào bảng thống kê các câu truy vấn trùng lặp). Có tổng cộng 253448 câu câu truy vấn với 3 đặc trưng tương ứng: câu truy vấn (prompt), câu trả lời (answer), nhãn quan hệ (rel_label). Kiểu dữ liệu của từng đặc trưng là kiểu đối tượng (object). Trong bảng 4.1 thể hiện thống kê các thuộc tính trong tập dữ liệu. Trường *prompt* có tổng cộng 253448 mẫu, trong đó có 235728 giá trị khác nhau. Giá trị xuất hiện nhiều nhất trong trường *prompt* là câu truy vấn có nội dung "[MASK] has diplomatic relations with

Germany." với tần suất xuất hiện là 82. Trường *answer* cũng có 253448 mẫu, với 1729 giá trị khác nhau. Giá trị phổ biến nhất trong trường *answer* là "English" với tần suất xuất hiện là 12203 lần. Tương tự, trường *rel_ label* có 253448 mẫu, nhưng chỉ có 34 giá trị khác nhau. Giá trị xuất hiện nhiều nhất trong trường *rel_ label* là "P407 (language of work or name)" với tần suất 15786 lần.

	prompt	**answer**	**rel_label**
count	253448	253448	253448
unique	235728	1729	34
top	[MASK] has diplomatic relations with Germany.	English	P407(language of work or name)
freq	82	12203	15786

Bảng 4.1: Thống kê các thuộc tính trong tập dữ liệu.

Trong tập dữ liệu có 17720 câu truy vấn bị trùng nhau, có thể nhận thấy rằng có những lý do sau: hai câu truy vấn giống nhau nhưng thuộc hai quan hệ dữ liệu khác nhau hoặc hai câu truy vấn giống nhau trong cùng quan hệ nhưng khác nhau về kết quả ở vị trí dự đoán (vị trí từ [MASK]). Ngoài ra, độ dài ngắn nhất của câu trả lời (prompt) là 16 ký tự và lớn nhất là 119 ký tự. Chúng tôi thực hiện thống kê độ dài của các câu truy vấn trong tập dữ liệu và sử dụng biểu đồ trực quan kết quả thu được biểu đồ 4.1:



Hình 4.1: Biểu đồ thể phân phối chiều dài câu truy vấn trong tập dữ liệu PARAREL.

Biểu đồ 4.1 cho thấy các đặc điểm sau:

- Phân phối lệch trái: Phần lớn các câu truy vấn có độ dài từ 30 đến 50 từ, với đỉnh phân phối nằm trong khoảng này. Các câu truy vấn ngắn hơn hoặc dài hơn xuất hiện với tần suất thấp hơn.

- Độ dài trung bình: Chiều dài trung bình của câu truy vấn nằm trong khoảng 40 đến 50 từ. Điều này cho thấy các câu hỏi trong tập dữ liệu thường có độ dài vừa phải, không quá ngắn cũng không quá dài.

- Độ phân tán: Mặc dù đỉnh phân phối tập trung ở khoảng 40-50 từ, nhưng độ dài câu truy vấn vẫn có sự phân tán đáng kể, từ khoảng 20 đến 80 từ. Điều này cho thấy sự đa dạng trong cách diễn đạt và độ phức tạp của các câu hỏi trong tập dữ liệu.

- Câu truy vấn dài xuất hiện rất ít: Các câu truy vấn có độ dài trên 80 từ xuất hiện rất ít, cho thấy tập dữ liệu không chứa nhiều câu hỏi phức tạp hoặc yêu cầu nhiều thông tin.

Tiếp đến, chúng tôi thực hiện tiền xử lý dữ liệu với mục đích loại bỏ các ký tự đặc biệt và các ký tự khoảng trắng. Sau khi tiền xử lý, chúng tôi tiến hành thống kê số lượng các từ xuất hiện trong tập dữ liệu và thực hiện trực quan hoá bằng thư viện WordCloud được hỗ trợ bởi Python. Hình 4.2 biểu diễn kết quả thu được:

Hình 4.2: Biểu đồ thống kê số lượng các từ xuất hiện trong tập dữ liệu.

Biểu đồ 4.2 cho thấy tập dữ liệu PARAREL tập trung chủ yếu vào các chủ đề liên quan đến:

- Con người: Các từ như "person", "citizenship", "born", "died", "life", "native", "tongue", "religion", "twin" và tên các quốc gia, thành phố cho thấy tập dữ liệu chứa nhiều thông tin về tiểu sử, nguồn gốc, quốc tịch, tôn giáo và nơi ở của con người.

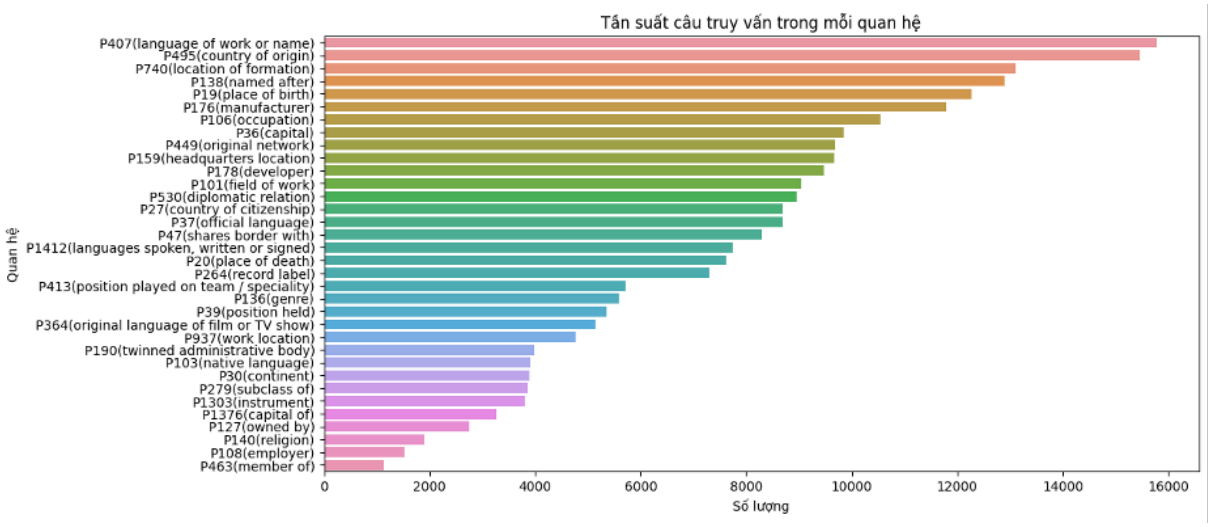
- Nghệ thuật và giải trí: Các từ như "music", "artis", "playing", "released", "label", "show" và "premiered" cho thấy tập dữ liệu có nhiều thông tin về âm nhạc, nghệ sĩ, các sản phẩm âm nhạc và sự kiện liên quan đến nghệ thuật.

- Địa điểm: Các từ như "city", "located", "headquarters", "country", "borders", "continent", "airport" và tên các quốc gia, thành phố cho thấy tập dữ liệu chứa nhiều thông tin về địa điểm, vị trí địa lý của các tổ chức, sự kiện và con người.

- Quan hệ: Các từ như "relations", "with", "belongs", "shares", "part", "formed", "union", "work" và "diplomatic" cho thấy tập dữ liệu chứa nhiều thông tin về các mối quan hệ giữa các cá nhân, tổ chức và quốc gia.

- Ngoài ra, tập dữ liệu cũng chứa các thông tin về các lĩnh vực khác như luật pháp ("law"), thể thao ("football"), doanh nghiệp ("company", "headquarters") và lịch sử ("formed", "originated", "passed").

Tiếp đến, chúng tôi thực hiện thống kê số lượng câu truy vấn và số lượng mẫu câu trong mỗi quan hệ dữ liệu và thu được biểu đồ 4.3:



Hình 4.3: Biểu đồ thống kê số lượng câu truy vấn trong mỗi quan hệ dữ liệu.

Từ biểu đồ 4.3 có thể thấy rằng hai quan hệ có số lượng câu truy vấn lớn nhất là quan hệ "P407 (language of work or name)" và "P495 (country of origin)". Bên cạnh đó, quan hệ có số lượng câu truy vấn ít hơn 2000

câu truy vấn là các quan hệ "P140 (religion)", "P108 (employer)" và "P463 (member of)".

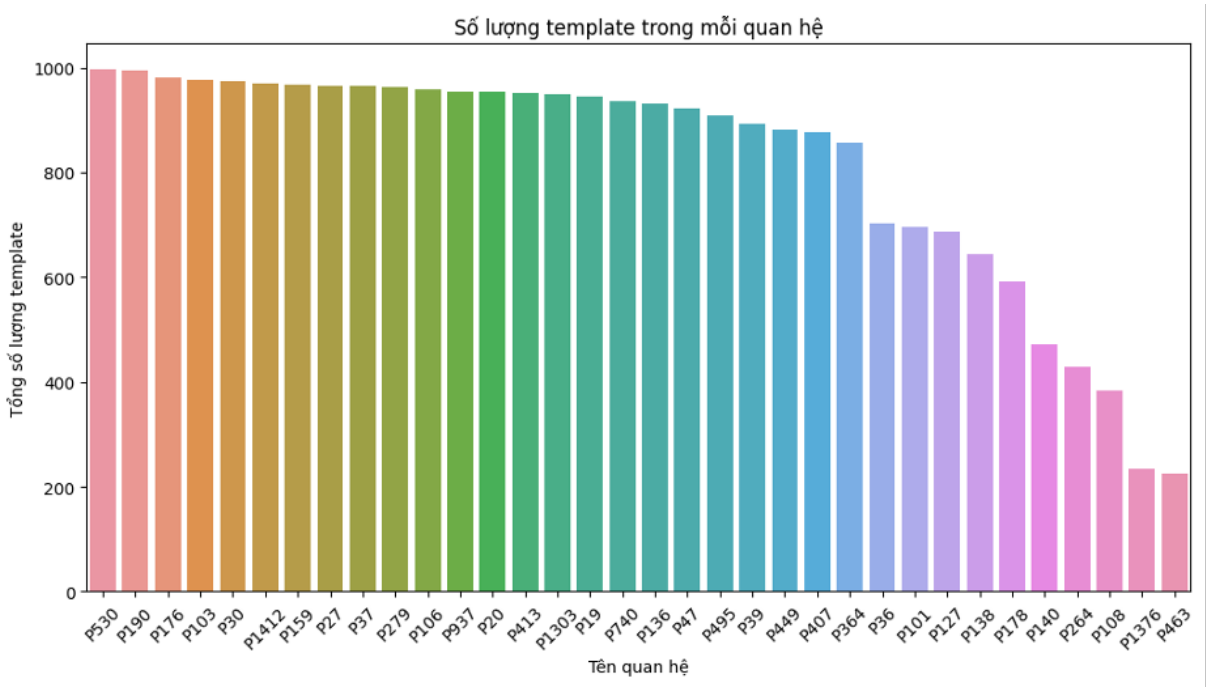
Phân tích khám phá cho tập dữ liệu data_all_allbags.json

Cấu trúc dữ liệu sau khi biến đổi bao gồm bốn cột chính: tên quan hệ dữ liệu (relation), số lượng mẫu câu trong mỗi quan hệ (template_count), tổng số lượng phần tử thuộc mẫu câu của mỗi quan hệ (total_item), danh sách chứa số lượng phần tử của mẫu câu trong quan hệ tương ứng (item_count_in_template). Một số mẫu dữ liệu được trình bày ở bảng 4.2.

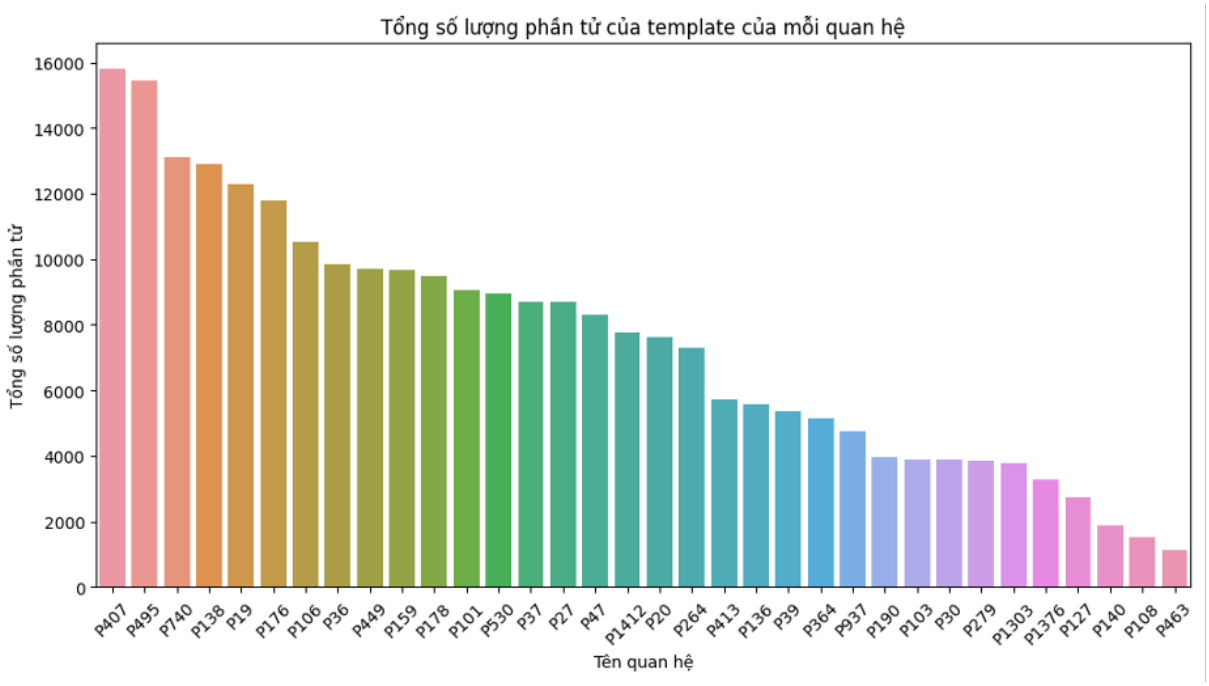
	relation	**template_count**	**total_item**	**item_count_in_template**
28	P463	225	1125	[5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ...]
30	P495	909	15453	[17, 17, 17, 17, 17, 17, 17, 17, 17, ...]
31	P530	996	8964	[9, 9, 9, 9, 9, 9, 9, 9, 9, 9, ...]
16	P20	953	7624	[8, 8, 8, 8, 8, 8, 8, 8, 8, 8, ...]
8	P138	645	12900	[20, 20, 20, 20, 20, 20, 20, 20, 20, ...]

Bảng 4.2: Bảng DataFrame thể hiện 5 mẫu dữ liệu của tập dữ liệu data_all_allbags.json. Các mẫu câu ở các quan hệ có số lượng phần tử tương đối đồng đều (5 phần tử ở quan hệ P463, 17 phần tử ở quan hệ P495, 9 phần tử ở quan hệ P530...).

Sau khi biến đổi dữ liệu ở dạng cấu trúc DataFrame, chúng tôi thực hiện thống kê số lượng mẫu câu trong mỗi quan hệ và thống kê số lượng phần tử trong các mẫu câu của mỗi quan hệ. Hình 4.4 và 4.5 trực quan các kết quả thực hiện.



Hình 4.4: Biểu đồ thống kê số lượng mẫu câu trong mỗi quan hệ



Hình 4.5: Biểu đồ thống kê số lượng mẫu câu trong mỗi quan hệ

Từ hai biểu đồ 4.4 và 4.5 cho thấy rằng số lượng phần tử của quan

hệ "P407 (language of work or name)" và "P495 (country of origin)" đứng đầu về số lượng phần tử, tương ứng với biểu đồ thể hiện số lượng mẫu câu thuộc hai quan hệ dữ liệu này cũng nằm trong ngưỡng giá trị cao nhất. Điều này cho thấy số lượng mẫu câu càng nhiều thì số lượng phần tử tương ứng của quan hệ đó càng lớn. Các quan hệ có số lượng mẫu câu và phần tử thuộc mẫu câu thấp nhất trong tập dữ liệu là những quan hệ "P108 (employer)", "P1376 (capital of)", "P463 (member of)". Ngoài ra, hơn một nửa số quan hệ của tập dữ liệu có số lượng mẫu câu lớn hơn 800 mẫu câu. Điều này cho thấy tập dữ liệu đảm bảo về tính đa dạng.

4.1.2 Môi trường cài đặt

Để có thể đảm bảo tiến độ của đề tài và tối ưu chi phí trong quá trình thực nghiệm, chúng tôi sử dụng kết hợp tài nguyên được cung cấp ở trang Kaggle và Google Colab. Tài nguyên cụ thể được sử dụng ở đề tài như sau:

- Ở trang Kaggle: GPU P100 sử dụng tối đa 8GB RAM với các tác vụ được thực hiện. Kết quả thực thi chiếm dung lượng không đáng kể ở ổ cứng lưu trữ (Disk).
- Ở Google Colab: sử dụng kết hợp GPU L4 và GPU A100. RAM và ổ cứng được cung cấp dư dả với yêu cầu của các tác vụ.

4.1.3 Ngôn ngữ và thư viện lập trình

Chúng tôi sử dụng Python làm ngôn ngữ lập trình ở đề tài này. Mô hình được cài đặt sử dụng thư viện Pytorch. Quá trình thực thi được cài đặt ở các tập tin notebook (các tập tin có phần mở rộng là ipynb) và mã nguồn được thực thi thông qua tập lệnh bash (các tập tin có phần mở rộng là sh).

Quá trình thực nghiệm phụ thuộc vào thư viện transformers phiên bản 4.20.0 xây dựng bởi Hugging Face và không được đảm bảo với các phiên

bản khác (một số phiên bản mới hơn không thể sử dụng vì có sự thay đổi ở mã nguồn của thư viện).

4.1.4 Hàm lỗi và độ đo chính xác

Độ đo được sử dụng ở quá trình chất lọc tri thức là giá trị ngưỡng t và ngưỡng p được trình bày ở mục 3.4. Các độ đo khác được sử dụng là độ chính xác của mô hình trong quá trình dự đoán (so sánh kết quả dự đoán với nhãn đúng) và độ đo Perplexity (PPL) được tính toán dựa trên hàm lỗi Cross Entropy.

4.1.5 Các tham số thực nghiệm

Chúng tôi thực hiện thống kê và đánh giá dựa trên ý tưởng được trình bày ở mục 4.4 - *Statistics of Knowledge Neurons* và 4.5 - *Knowledge Neurons Affect Knowledge Expression* được trình bày ở bài báo *Knowledge Neurons in Pretrained Transformers* [5]. Chúng tôi thực hiện bằng cách chạy chương trình nhiều lần với các tham số khảo sát khác nhau và thực hiện so sánh các kết quả, mỗi lần chạy đều được thực hiện trên toàn bộ tập dữ liệu được cung cấp. Các tham số được lựa chọn khảo sát bao gồm:

- Giá trị tăng cường điểm kích hoạt của các nơ-ron tri thức. Giá trị này được khởi tạo là 2 (ở nghiên cứu của Damai Dai 2022 [5]), chúng tôi tiếp tục khảo sát với các giá trị 4, 6, 8, 10 và 12. Mục tiêu của phương pháp tích hợp độ dốc là đem lại sự tương đồng tỷ lệ thuận giữa tầm quan trọng của nơ-ron và giá trị phân bổ. Do đó, chúng tôi lựa chọn tham số này với mục đích khảo sát tính đúng đắn của kết quả điểm phân bổ được tính toán.

- Số lượng nơ-ron tri thức được chọn trong quá trình loại bỏ tri thức đối với quan hệ dữ liệu được chỉ định. Chúng tôi khảo sát với số lượng nơ-ron tri thức được chọn lần lượt là 30, 40 và 50 nơ-ron. Chúng

tôi lựa chọn tham số này với mục đích khảo sát sự tương đồng giữa lượng tri thức được loại bỏ và hiệu suất của mô hình. Liệu rằng với càng nhiều tri thức bị loại bỏ, mô hình sẽ bị ảnh hưởng đáng kể hay không.

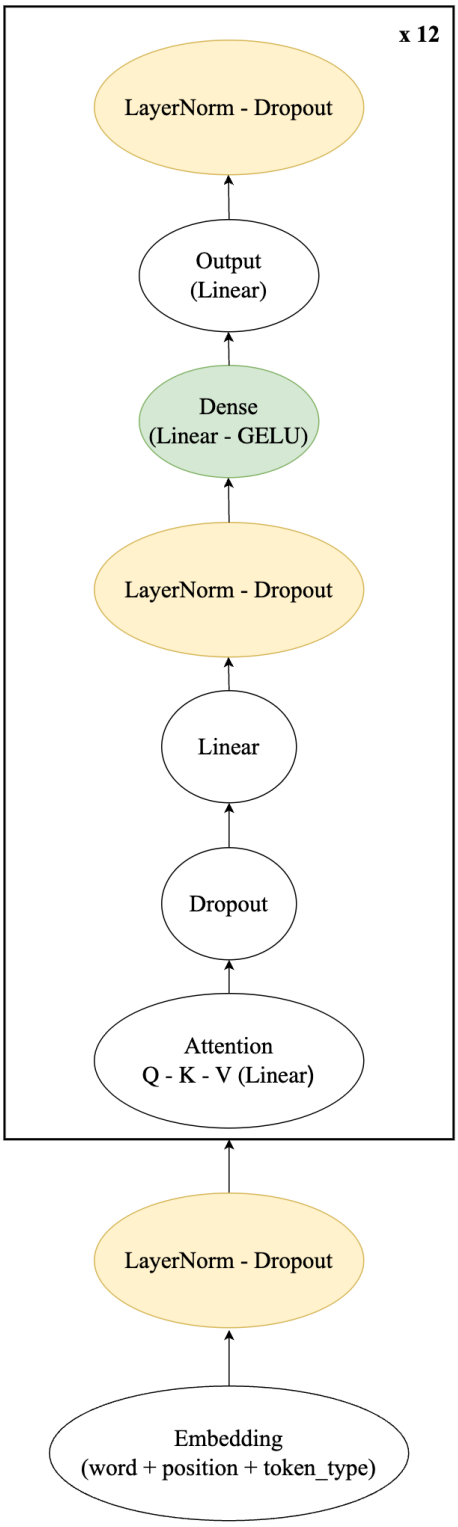
4.2 Lựa chọn mô hình

Có rất nhiều mô hình ngôn ngữ dựa trên cấu trúc Transformer tiền huấn luyện nổi bật như Generative Pre-trained Transformer (OpenAI's GPT-1) [17], RoBERTa [13]. Mô hình ngôn ngữ được lựa chọn cho đề tài cần có cấu trúc phù hợp với phương pháp đồng thời có kích thước phù hợp cho quá trình thực nghiệm. Mô hình có cấu trúc phù hợp được tạo thành từ hai thành phần cơ bản của Transformer bao gồm tầng tự chú ý (hoạt động dựa trên cơ chế truy vấn - khóa - giá trị hay Q - K - V), mạng nơ-ron truyền thẳng gồm hai tầng với một tầng ẩn và tầng đầu ra tương ứng. Đối với tiêu chí về kích thước của mô hình, mô hình được lựa chọn cần có số lượng tham số đủ nhỏ để phù hợp với tiến độ nghiên cứu cũng như tài nguyên (phần cứng, phần mềm) thực nghiệm. Mô hình có cấu trúc càng lớn và phức tạp sẽ yêu cầu càng nhiều tài nguyên cho quá trình tính toán điểm phân bổ, quá trình lọc nơ-ron tri thức và xóa tri thức. Dựa trên những tiêu chí đã đề ra, mô hình được sử dụng trong đề tài là mô hình Transformer tiền huấn luyện BERT cơ sở được đề xuất ở nghiên cứu của Jacob Devlin 2019 [7].

Điểm nổi bật của mô hình BERT là tính chất hai chiều (bidirectional) so với các mô hình ngôn ngữ khác là đơn chiều (unidirectional). Ở mô hình GPT-1 [17] được phát triển bởi OpenAI, ngữ cảnh dự đoán chỉ có thể đến từ các từ trước đó (các từ ở phía bên trái thuộc văn bản dữ liệu đầu vào) thì đối với BERT, mô hình có thể dựa vào ngữ cảnh toàn cục trong văn bản (các từ bên trái và bên phải) để đạt được kết quả dự đoán chính xác hơn. Dữ liệu trong quá trình tính toán của mô hình BERT được gọi là chuỗi (sequence) đầu vào, với mỗi chuỗi được cấu tạo từ một cặp câu

chứa các từ ngữ liên tiếp. Trước khi được đưa vào mô hình, các chuỗi đầu vào này cần được thêm các ký tự đặc biệt *"[CLS]"* ở đầu mỗi chuỗi, ký tự *"[SEP]"* ở vị trí phân cách hai câu và ở cuối mỗi chuỗi.

Để mô hình có thể đạt được tính chất hai chiều, nhóm tác giả ở Google AI Language đã áp dụng hai tác vụ không giám sát là masked language model (MLM) và next sentence prediction (NSP) cho quá trình huấn luyện mô hình. Tổng quát, masked language model mô tả phương pháp che khuất dữ liệu đối với dữ liệu huấn luyện. Với mỗi chuỗi đầu vào, bộ sản sinh dữ liệu huấn luyện lựa chọn ngẫu nhiên 15% số lượng vị trí các từ trong chuỗi nhằm thực hiện thay thế với một từ khác. Từ được chọn sẽ được thay thế với (1) từ *"[MASK]"* trong 80% số lần thực hiện (2) một từ ngẫu nhiên khác trong 10% số lần thực hiện (3) chính từ được chọn trong 10% số lần thực hiện còn lại. Next sentence prediction mô tả quá trình lựa chọn một cặp câu tạo nên một chuỗi huấn luyện. Một chuỗi huấn luyện được tạo thành bởi hai câu nhưng thay vì chọn ngẫu nhiên một cặp câu bao gồm hai câu A và B, trong 50% số lần thực hiện thì câu B được lựa chọn là câu kế tiếp sẽ xuất hiện tương ứng với câu A (với nhãn IsNext) và 50% còn lại là sự lựa chọn ngẫu nhiên. Tập dữ liệu huấn luyện được sử dụng là tập dữ liệu BooksCorpus là kết quả nghiên cứu của Yukun Zhu 2015 [23] bao gồm 800 triệu từ và English Wikipedia bao gồm 2,500 triệu từ đã được loại bỏ các thành phần danh sách, bảng và tiêu đề.

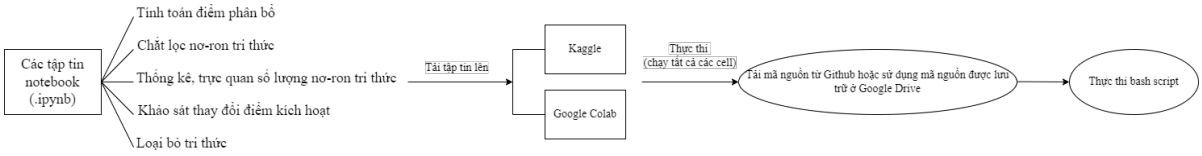


Hình 4.6: Hình ảnh trực quan mô hình được sử dụng ở đề tài. Trong quá trình tính toán điểm phân bổ tích hợp tích phân (quá trình khảo sát giá trị kích hoạt) diễn ra ở tầng *Dense* (tầng ẩn thuộc mạng nơ-ron), các tầng *Dropout* không được sử dụng. Quá trình xóa tri thức diễn ra ở tầng *Output* (tầng đầu ra thuộc mạng nơ-ron).

Mô hình BERT (bert-base-cased) được sử dụng ở đề tài có tổng cộng 110 triệu tham số được cấu trúc bởi 12 tầng Transformer, trong đó mỗi tầng Transformer chứa 12 đầu tự chú ý với chiều dữ liệu là 768 ($d = 768$, $d_m = 3072$). Số lượng từ ngữ thuộc từ điển phục vụ cho quá trình mã hóa và giải mã là 28996 từ cùng hàm kích hoạt GELU được sử dụng. Mô hình được chỉnh sửa một số thành phần để phù hợp với quá trình thực nghiệm nhưng phần lớn vẫn đảm bảo cấu trúc ban đầu của Jacob Devlin 2019 [7]. Hình 4.6 thể hiện trực quan mô hình được sử dụng trong quá trình thực nghiệm của đề tài. Ở thời điểm thực hiện đề tài, các tham số tiền huấn luyện và cấu hình của mô hình có thể được truy cập thông qua đường dẫn Amazon S3 hoặc truy cập mô hình đã được triển khai ở trang Hugging Face sử dụng từ khóa *"google-bert/bert-base-cased"*.

4.3 Áp dụng phương pháp

Dựa trên nghiên cứu của Damai Dai 2022 [5], mỗi tập tin mã nguồn sẽ nắm giữ một vai trò trong phương pháp. Chúng tôi xây dựng các tập tin notebook (các tập tin với phần mở rộng là ipynb) tương ứng nhằm thực thi những tập tin mã nguồn này. Hình 4.7 trực quan tổng quát cách thức tiến hành thực nghiệm của chúng tôi ở đề tài này.



Hình 4.7: Hình ảnh trực quan cách thức tiến hành thực nghiệm. Các tập tin notebook được xây dựng tương ứng với 5 tác vụ của quá trình thực nghiệm. Mã nguồn của chương trình được lưu trữ bởi GitHub hoặc Google Drive, được thực thi sử dụng bash script thông qua các tập tin notebook.

Các tác vụ chính của quá trình khảo sát được trình bày theo các mục nhỏ, bao gồm (1) Tính toán điểm phân bổ của các nơ-ron với mỗi câu truy

vấn đầu vào, (2) Chắt lọc nơ-ron tri thức, (3) Thực hiện thống kê số lượng nơ-ron tri thức, (4) Khảo sát thay đổi giá trị phân bổ, (5) Loại bỏ tri thức khỏi mô hình.

4.3.1 Tính toán điểm phân bổ của các nơ-ron với mỗi câu truy vấn đầu vào

Quá trình tính toán điểm phân bổ áp dụng phương pháp độ dốc được thực hiện bởi tập tin *1_ analyze_ mlm.py*. Tập tin nhận đầu vào là tập dữ liệu PARAREL đã qua xử lý *data_ all_ allbags.json* và kết quả đầu ra là một tập tin *.rlt.jsonl* chứa thông tin các nơ-ron (chỉ số l, i được trình bày ở biểu thức 3.1) cùng điểm phân bổ của quan hệ dữ liệu được chỉ định thực thi (mỗi lần chạy chỉ có thể tính toán với một quan hệ dữ liệu cụ thể).

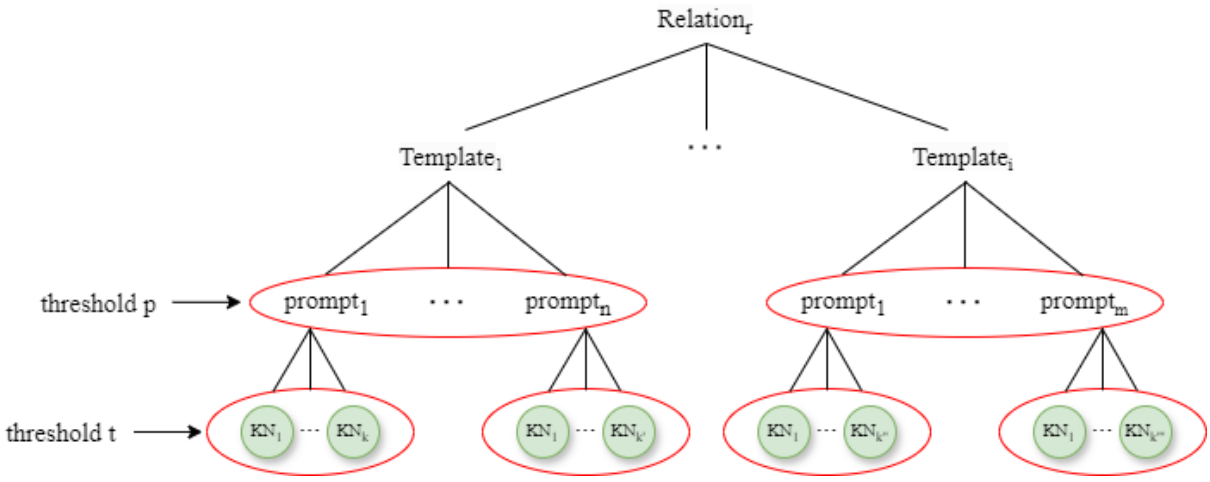
Với mỗi câu truy vấn đầu vào thuộc quan hệ dữ liệu được chỉ định, biểu thức 3.2 được cài đặt sử dụng các vòng lặp. Giá trị kích hoạt của các nơ-ron ở tầng ẩn tương ứng với mỗi tầng con trong mô hình BERT kết hợp tính toán áp dụng độ dốc để tạo ra điểm phân bổ của phương pháp.

4.3.2 Chắt lọc nơ-ron tri thức

Dựa trên kết quả ở bước "Tính toán điểm phân bổ của các nơ-ron với mỗi câu truy vấn đầu vào", tập tin *2_ get_ kn.py* tiến hành chắt lọc nơ-ron dựa trên hai tham số t và p được trình bày ở mục 3.4. Tập tin nhận đầu vào là một tập tin *.rlt.jsonl* tương ứng với mỗi quan hệ dữ liệu để thực hiện tính toán và kết quả đầu ra là bốn tập tin *.json* lưu trữ vị trí l, i của các nơ-ron tri thức. Bốn tập tin đầu ra của mỗi quan hệ dữ liệu được chia thành hai nhóm chính bao gồm kết quả dựa trên điểm kích hoạt cơ sở (được tính toán với giá trị kích hoạt ban đầu của các nơ-ron); kết quả dựa trên điểm phân bổ của phương pháp tích hợp độ dốc [5].

Phương pháp chắt lọc được áp dụng cho cả điểm phân bổ được tích hợp độ dốc và điểm phân bổ cơ sở. Ở quá trình thực nghiệm, giá trị ngưỡng

t được lựa chọn cho phương pháp tích hợp độ dốc là 0.2, giá trị ngưỡng p được khởi tạo là 0.7. Toàn bộ quá trình chắt lọc cho một quan hệ dữ liệu được thực hiện tối đa 6 lần và quá trình dừng lại khi số lượng nơ-ron tri thức trung bình của mỗi mẫu câu thu được $\in [2, 5]$ nơ-ron. Để số lượng nơ-ron thu được có thể đạt khoảng giá trị $\in [2, 5]$ trong quá trình chắt lọc, giá trị ngưỡng p sẽ giảm 0.05 (giảm nhẹ điều kiện chắt lọc) nếu số lượng nơ-ron thu được ít hơn 2 và sẽ tăng 0.05 (khắt khe ở điều kiện chắt lọc) nếu số lượng nơ-ron thu được nhiều hơn 5. Hình 4.8 thể hiện trực quan phương pháp chắt lọc được trình bày.



Hình 4.8: Hình ảnh trực quan phương pháp chắt lọc nơ-ron tri thức sử dụng cấu trúc cây. Một quan hệ dữ liệu r được tạo thành từ $i > 3$ mẫu câu, mỗi mẫu câu có n, m câu truy vấn và mỗi câu truy vấn có k, k', k'', k''' số lượng nơ-ron tương ứng. Giá trị ngưỡng t được áp dụng lần lượt với các tập nơ-ron, giá trị ngưỡng p tiếp tục được áp dụng với tập thô thu được. Các nơ-ron tri thức thu được tương ứng với mỗi mẫu câu thuộc quan hệ dữ liệu.

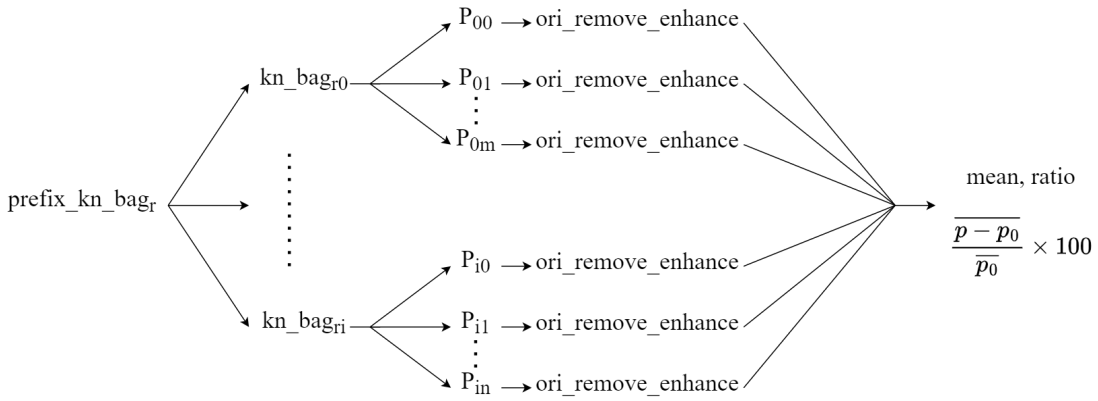
4.3.3 Thực hiện thống kê số lượng nơ-ron tri thức

Dựa trên các tập tin *.json* ở bước "Chắt lọc nơ-ron tri thức", tập tin *2_ analyze_ kn.py* thực hiện thống kê và trực quan số lượng nơ-ron tri thức của điểm phân bổ cơ sở và điểm phân bổ tích hợp độ dốc. Tập tin nhận

đầu vào là các tập tin *.json* tương ứng, tạo ra kết quả là hai tập tin *.pdf* trực quan sự phân bổ của các nơ-ron ở điểm cơ sở và điểm phân bổ của phương pháp cùng các thông tin thống kê được in ra màn hình.

4.3.4 Khảo sát thay đổi giá trị kích hoạt

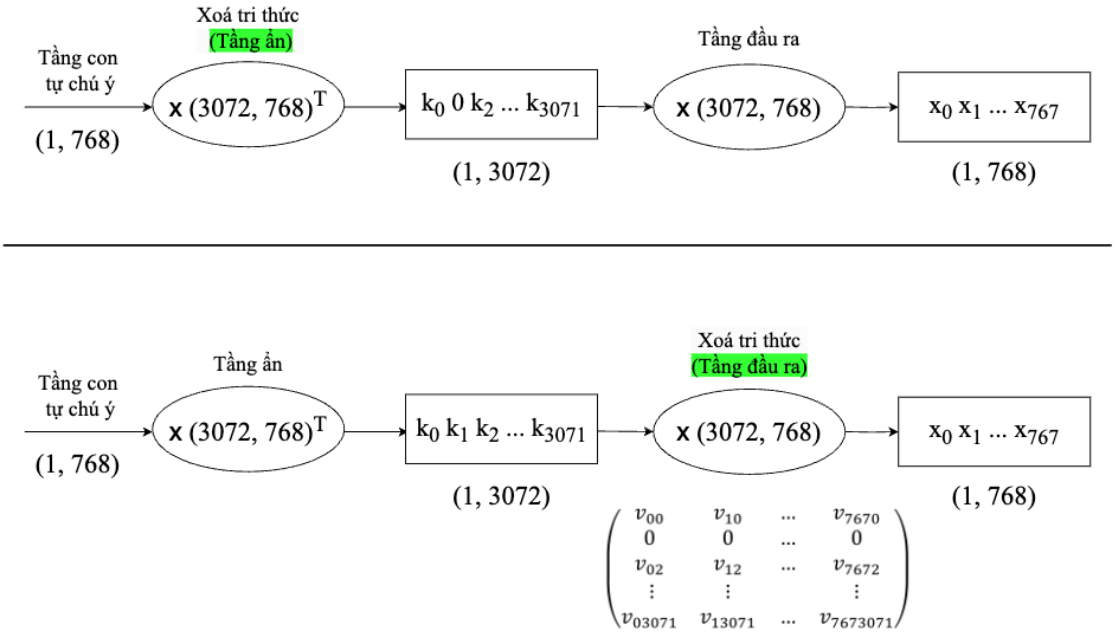
Dựa trên các tập tin *.json* ở bước "Chắt lọc nơ-ron tri thức", tập tin *3_ modify_ activation.py* thực hiện thay đổi giá trị điểm kích hoạt của các nơ-ron tri thức với kết quả là tập tin *modify_ activation_ rlt.json* tương ứng dựa trên điểm cơ sở hoặc điểm phân bổ của phương pháp. Tập tin kết quả này thể hiện sự chênh lệch về xác suất dự đoán là nhãn đúng sau và trước khi thay đổi điểm kích hoạt. Hình 4.9 trực quan cho quá trình thực hiện khảo sát.



Hình 4.9: Hình ảnh trực quan quá trình thay đổi điểm kích hoạt với hai phương thức chính là gán giá trị 0 (remove) và tăng cường bằng cách nhân với một hệ số dương (enhance). Mỗi quan hệ dữ liệu r có các túi nơ-ron tri thức i tương ứng với các mẫu câu trong tập dữ liệu. Với mỗi câu truy vấn m, n thuộc mẫu câu i, giá trị điểm kích hoạt được thay đổi nhằm khảo sát sự chênh lệch về xác suất dự đoán là nhãn đúng sau và trước khi thay đổi. Các kết quả được lấy trung bình trên nhiều câu truy vấn đầu vào.

4.3.5 Loại bỏ tri thức khỏi mô hình

Tập tin *7_ erase_ knowledge.py* tiến hành loại bỏ tri thức về quan hệ dữ liệu được chỉ định trong mô hình đồng thời đánh giá độ đo Perplexity, độ chính xác của mô hình trước và sau quá trình loại bỏ tri thức. Tập tin nhận đầu vào là tập dữ liệu *data_ all_ allbags.json* sử dụng cho quá trình khảo sát hiệu suất của mô hình, tập tin *kn_ bag.json* chứa các nơ-ron tri thức cần xóa của quan hệ dữ liệu được chỉ định.



Hình 4.10: Hình ảnh minh họa quá trình loại bỏ tri thức của nơ-ron thứ hai (có chỉ số $i = 1$) diễn ra ở tầng ẩn (phần phía trên) và tầng đầu ra (phần phía dưới) thuộc cùng tầng con l bất kỳ. Kết quả của hai cách thức loại bỏ tri thức này về lý thuyết là tương đương nhau nhưng cài đặt hàm kích hoạt GELU ảnh hưởng đến kết quả này.

Quá trình được tiến hành như sau: chương trình sử dụng bộ đếm nhằm tìm ra $n \in \{20, 30, 40, 50\}$ nơ-ron tri thức phổ biến nhất (nơ-ron phổ biến là những nơ-ron được bộ đếm phát hiện nhiều lần). Sau khi thu được tập nơ-ron tri thức phổ biến, chương trình tiến hành đánh giá mô hình và loại bỏ tri thức. Quá trình loại bỏ tri thức được thực hiện bằng cách gán giá trị 0 cho các trọng số của mô hình ở vị trí tương ứng với các nơ-ron tri

thức (dựa trên chỉ số l, i). Quá trình được thực hiện đối với tầng đầu ra của các mạng nơ-ron thay vì đối với tầng ẩn trên lý thuyết. Lý do cho phương pháp này dựa trên cài đặt của mô hình vì ở tầng ẩn được cài đặt hàm kích hoạt GELU, yêu cầu tính toán dựa trên toàn bộ ma trận. Mặc dù được thực hiện khác với lý thuyết nhưng hai vị trí loại bỏ tri thức này là tương đồng, được chúng tôi làm rõ ở hình 4.10.

4.4 Thời gian thực thi

Thời gian thực thi cho mỗi tác vụ được trình bày ở mục 4.3 cụ thể như sau:

- Tính toán điểm phân bổ của các nơ-ron với mỗi câu truy vấn đầu vào: với mục đích tối ưu chi phí và thử nghiệm, chúng tôi tiến hành tính toán điểm phân bổ sử dụng bốn loại GPU khác nhau. Với GPU T4 thời gian tính toán ≈ 3.1 giây/câu truy vấn; GPU P100 là ≈ 1.9515 giây/câu truy vấn; GPU L4 là ≈ 1.68 giây/câu truy vấn; GPU A100 là ≈ 0.9176 giây/câu truy vấn. Chi tiết thời gian tính toán của 32 quan hệ dữ liệu được chúng tôi tổng hợp ở bảng 4.3.

- Chắt lọc nơ-ron tri thức: sử dụng GPU A100, thời gian thực hiện tác vụ này ≈ 16 phút.

- Thực hiện thống kê số lượng nơ-ron tri thức: với GPU P100, tác vụ này yêu cầu ≈ 10 phút.

- Khảo sát thay đổi giá trị phân bổ: với GPU P100 và thực hiện với loại điểm phân bổ cụ thể (cơ sở hoặc phương pháp tích hợp độ dốc), tác vụ này yêu cầu $\approx 11,711$ giây (3.25 giờ).

- Loại bỏ tri thức khỏi mô hình: với GPU P100, tác vụ này yêu cầu ≈ 2.4 giờ.

Loại GPU	Quan hệ dữ liệu	Số lượng câu truy vấn	Thời gian thực thi (giây)
T4	P30	3900	11569.9727
	P364	5136	15464.7142
	P108	1532	4805.6624
P100	P127	2748	5422.8923
	P101	9048	17642.6500
	P103	3908	7616.6720
	P1376	3276	6381.4401
	P1303	3796	7522.4580
	P138	12900	25127.7399
	P140	1892	3687.2965
	P19	12272	23951.8625
	P20	7624	14910.1457
	P27	8694	16945.9757
	P740	13104	25504.4368
	P407	15786	30617.8699
	P47	8298	16158.9662
	P106	10538	20581.9380
	P159	9670	18890.7267
	P176	11784	22961.1849
	P178	9472	18512.2530
	P190	3980	7796.7133
	P449	9691	18873.0350
	P1412	7752	15206.7876
	P136	5586	10927.1860
	P937	4770	9319.4436
	P495	15453	30126.1403
	P36	9842	19167.6565
L4	P463	1125	1886.5109
	P39	5352	9008.7768
A100	P279	3856	3551.0416
	P37	8694	8022.6338
	P413	5712	5194.1283

Bảng 4.3: Thời gian tính toán điểm phân bổ của 32 quan hệ dữ liệu.

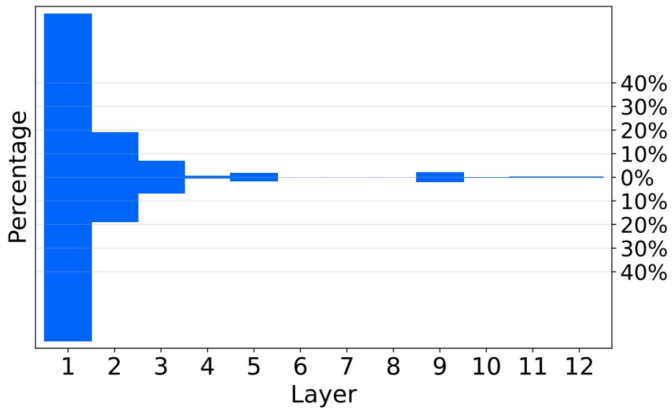
4.5 Kết quả thực nghiệm

Một số giá trị quan trọng liên quan đến quá trình thực nghiệm của các kết quả được trình bày: điểm tham chiếu khi tính toán điểm phân bổ tích hợp độ dốc là giá trị 0 và bước xấp xỉ $m = 20$; giá trị ngưỡng $t = 0.2$.

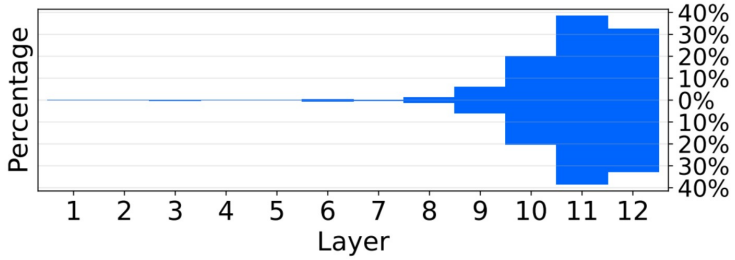
Kết quả thống kê số lượng nơ-ron tri thức được trình bày ở bảng 4.4, tỉ lệ phân bổ của các nơ-ron dựa trên hai loại điểm phân bổ được trình bày ở hình 4.11 và 4.12 tương ứng.

	Phương pháp tích hợp độ dốc	**Phương pháp cơ sở**
Số lượng nơ-ron tri thức trung bình của mỗi mẫu câu	4.1338	3.9632
Số lượng nơ-ron tri thức giao nhau giữa hai mẫu câu bất kỳ thuộc cùng một quan hệ	1.2279	2.8466
Số lượng nơ-ron tri thức giao nhau giữa hai mẫu câu bất kỳ thuộc hai quan hệ bất kỳ	0.0932	1.9235

Bảng 4.4: Bảng thống kê số lượng nơ-ron tri thức của phương pháp tích hợp độ dốc và phương pháp cơ sở.

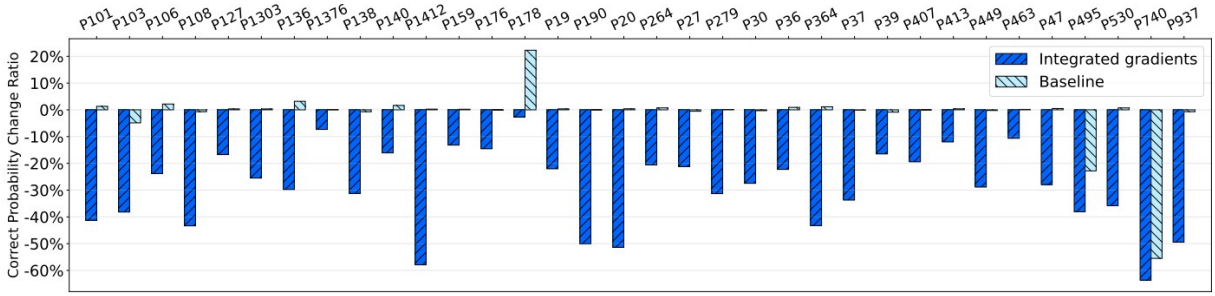


Hình 4.11: Tỉ lệ phân bổ các nơ-ron dựa trên điểm phân bổ cơ sở theo các tầng con trong mô hình.

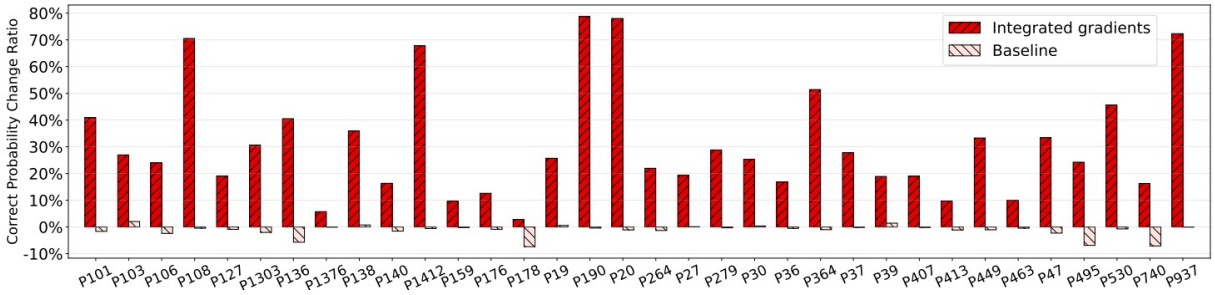


Hình 4.12: Tỉ lệ phân bổ các nơ-ron dựa trên điểm phân bổ tính toán với phương pháp tích hợp độ dốc theo các tầng con trong mô hình.

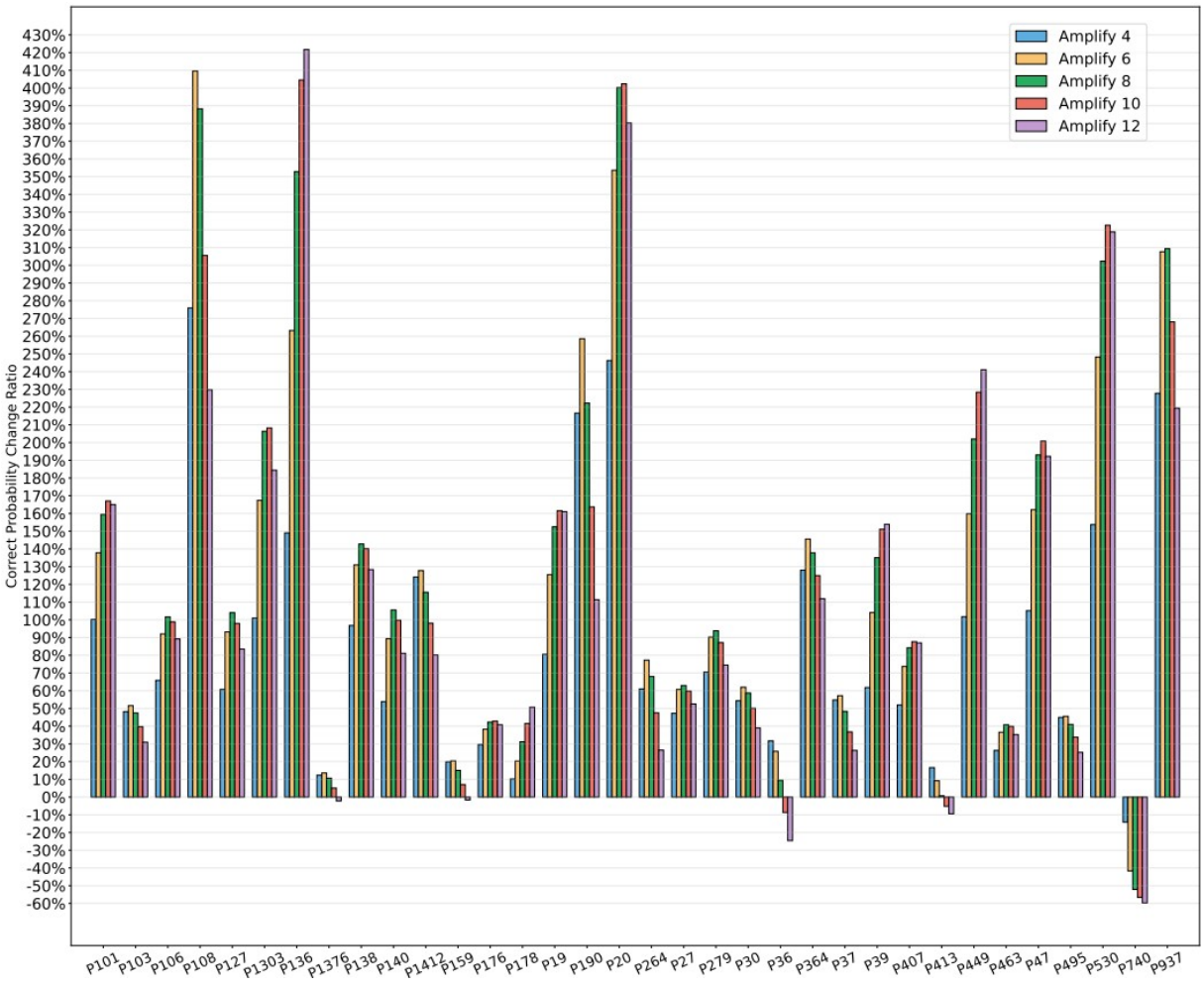
Kết quả khảo sát thay đổi giá trị kích hoạt được trình bày tương ứng với hình 4.13 - gán giá trị 0; hình 4.14 - tăng cường giá trị kích hoạt bằng cách nhân với hệ số 2; hình 4.15 - tăng cường giá trị kích hoạt bằng cách nhân với hệ số 4, 6, 8, 10, 12.



Hình 4.13: Tỉ lệ chênh lệch dự đoán của mô hình là nhãn đúng sau và trước khi gán giá trị kích hoạt bằng 0 dựa trên hai loại điểm phân bổ theo từng quan hệ dữ liệu.



Hình 4.14: Tỉ lệ chênh lệch dự đoán của mô hình là nhãn đúng sau và trước khi tăng cường giá trị kích hoạt bằng cách nhân với hệ số 2 dựa trên hai loại điểm phân bổ theo từng quan hệ dữ liệu.



Hình 4.15: Tỉ lệ chênh lệch dự đoán của mô hình là nhãn đúng sau và trước khi tăng cường giá trị kích hoạt bằng cách nhân lần lượt với các hệ số 4, 6, 8, 10, 12 dựa trên điểm phân bổ từ phương pháp tích hợp độ dốc theo từng quan hệ dữ liệu.

Bảng 4.5 thống kê hiệu suất của mô hình với số lượng nơ-ron tri thức được chọn trong quá trình loại bỏ tri thức đối với quan hệ dữ liệu được chỉ định.

Quan hệ xóa	Số lượng nơ-ron	Trên quan hệ xóa				Trên các quan hệ khác			
		ACC		PPL		ACC		PPL	
		Trước	Sau	Trước	Sau	Trước	Sau	Trước	Sau
P937 (work_location)	15 (baseline)	0.3338	0.3333 (-0.1%)	58.0	64.6 (+11.3%)	0.2272	0.2285 (+0.6%)	138.0	139.0 (+1.3%)
	15		0.1667 (-50.1%)		129.9 (+123.8%)		0.2233 (-1.7%)		153.3 (+11.1%)
	30		0.1019 (-69.5%)		173.5 (+199.0%)		0.2204 (-3.0%)		160.5 (+16.3%)
	40		0.0943 (-71.7%)		184.2 (+217.3%)		0.222 (-2.3%)		160.2 (+16.1%)
	50		0.0891 (-73.3%)		187.2 (+222.5%)		0.2258 (-0.6%)		158.1 (+14.6%)
P19 (place_of_birth)	15 (baseline)	0.0318	0.0348 (+9.5%)	1,450	1,341 (-7.5%)	0.2393	0.2392 (-0.04%)	120.3	115.7 (-3.8%)
	15		0.0368 (+15.9%)		2,873 (+98.1%)		0.2447 (+2.3%)		120.7 (+0.3%)
	30		0.0363 (+14.1%)		3,105 (+114.1%)		0.2454 (+2.5%)		121.6 (+1.0%)
	40		0.0257 (-19.2%)		4,025 (+177.6%)		0.2429 (+1.5%)		128.5 (+6.8%)
	50		0.0249 (-21.8%)		4,384 (+202.3%)		0.2424 (+1.3%)		130.6 (+8.5%)
P27 (country_of_citizenship)	10 (baseline)	0.3573	0.3559 (-0.3%)	28.0	27.7 (-1.2%)	0.2247	0.222 (-1.2%)	143.6	147.9 (+3.0%)
	10		0.3264 (-8.6%)		34.8 (+24.3%)		0.2226 (-0.9%)		152.3 (+6.1%)
	30		0.2869 (-19.7%)		39.0 (+39.3%)		0.2239 (-0.3%)		151.5 (+5.5%)
	40		0.2826 (-20.9%)		40.2 (+43.6%)		0.2243 (-0.2%)		152.4 (+6.1%)
	50		0.2687 (-24.8%)		43.6 (+55.7%)		0.2261 (+0.6%)		148.3 (+3.3%)
P106 (occupation)	14 (baseline)	0.0226	0.0224 (-0.8%)	2,279	1,956 (-14.2%)	0.2382	0.2372 (-0.4%)	120.1	118.9 (-1.0%)
	14		0.0176 (-22.1%)		5,048 (+121.5%)		0.2352 (-1.3%)		125.1 (+4.2%)
	27 (max)		0.0144 (-36.1%)		5,536 (+142.9%)		0.2348 (-1.4%)		127.1 (+5.8%)
P178 (developer)	9 (baseline)	0.2272	0.2875 (+26.5%)	204.5	53.3 (-74.0%)	0.2293	0.2298 (+0.2%)	133.6	142.5 (+6.6%)
	9		0.2138 (-5.9%)		391.9 (+91.6%)		0.2293 (0.0%)		136.5 (+2.2%)
	17 (max)		0.208 (-8.5%)		618.8 (+202.5%)		0.2268 (-1.1%)		142.2 (+6.4%)

Bảng 4.5: Bảng thống kê độ chính xác và độ đo perplexity của một số quan hệ dữ liệu trước và sau quá trình loại bỏ tri thức.

4.6 Đánh giá và phân tích kết quả thực nghiệm

Thống kê số lượng nơ-ron tri thức: Kết quả ở xác định nơ-ron ở bảng 4.4 thể hiện sự hiệu quả của phương pháp. Số lượng nơ-ron tri thức giao nhau giữa hai mẫu câu bất kỳ thuộc cùng một quan hệ được xác định bởi phương pháp tích hợp độ dốc ít hơn so với phương pháp cơ sở (1.2279 so với 2.8466), số lượng nơ-ron tri thức giao nhau giữa hai mẫu câu bất kỳ thuộc hai quan hệ bất kỳ thể hiện kết quả vượt trội hơn (0.0932 so với 1.9235). Số lượng nơ-ron giao nhau ít thể hiện rằng các nơ-ron thuộc về mẫu câu cụ thể và trong quá trình loại bỏ tri thức, chỉ những tri thức được chỉ định bị loại bỏ mà không ảnh hưởng đến lượng tri thức khác.

Dựa vào tỉ lệ phân bổ nơ-ron ở hình 4.11 có thể nhận thấy điểm phân bổ cơ sở xác định những nơ-ron tri thức ở tầng thấp (tầng 1 đến tầng 3) và ở những tầng cao (tầng 9 đến tầng 12), số lượng nơ-ron được xác định là rất ít. Trái ngược với kết quả dựa trên điểm phân bổ cơ sở, tỉ lệ

phân bổ các nơ-ron ở hình 4.12 dựa trên phương pháp tích hợp độ dốc xác định những nơ-ron ở tầng cao (tầng 9 đến tầng 12) là tri thức. Như những nghiên cứu được trình bày ở mục 3.3.1, những nơ-ron ở tầng thấp mang khả năng nhận diện dữ liệu có tính rập khuôn và không cân nhắc về mặt ngữ nghĩa (các mẫu dữ liệu mang nhãn "shallow"), trong khi đó những nơ-ron ở tầng cao mang khả năng nhận diện dữ liệu sử dụng yếu tố ngữ nghĩa (các mẫu dữ liệu mang nhãn "semantic"). Những nơ-ron ở tầng cao có ảnh hưởng đáng kể hơn đến quá trình dự đoán của mô hình vì ngôn ngữ tự nhiên phức tạp bởi ngữ nghĩa, hàm ý được chứa đựng và không đơn thuần là cấu trúc rập khuôn.

Khảo sát thay đổi giá trị kích hoạt: Tỉ lệ chênh lệch dự đoán của mô hình sau quá trình gán giá trị 0 cho điểm kích hoạt được trực quan ở hình 4.13. Khi thực hiện gán giá trị 0, kết quả dự đoán của mô hình dựa trên điểm phân bổ cơ sở có xu hướng giảm nhưng sự chênh lệch rất ít (từ 0% đến 5%). Một số ít quan hệ dữ liệu như P495, P740 giảm đáng kể xác suất dự đoán nhãn đúng, trong đó quan hệ P178 có tỉ lệ tăng bất thường (>20%). Trái ngược với kết quả của điểm phân bổ cơ sở, phương pháp tích hợp độ dốc thể hiện rõ ràng sự chênh lệch. Đa số các quan hệ dữ liệu có tỉ lệ dự đoán giảm đáng kể nhưng không đồng đều, trong đó giảm nhiều nhất là quan hệ P740 (>60%), giảm ít nhất là quan hệ P178 (<5%) và không có quan hệ nào tăng bất thường. Bên cạnh kết quả của quá trình gán giá trị 0, tỉ lệ chênh lệch dự đoán khi tăng cường điểm kích hoạt bằng cách nhân với các hệ số tương ứng là 2, 4, 6, 8, 10, 12 được trực quan ở hình 4.14 và 4.15. Trong quá trình nhân với hệ số 2, điểm phân bổ cơ sở có xu hướng giảm tỉ lệ dự đoán nhãn đúng nổi bật là các quan hệ P136, P178, P495 và P740 (>5%). Trong khi đó phương pháp tích hợp độ dốc tăng đáng kể tỉ lệ chênh lệch dự đoán nhưng không đồng đều ở các quan hệ, trong đó tăng nhiều nhất là quan hệ P190 ($\approx$ 80%) và tăng ít nhất là quan hệ P178 (<5%). Khi so sánh các hệ số nhân 4, 6, 8, 10 và 12, tỉ lệ chênh lệch dự đoán xảy ra ba trường hợp là tăng (tỉ lệ chênh lệch đồng biến với hệ số nhân như ở quan hệ P136, P449), giảm (tỉ lệ chênh

lệch nghịch biến với hệ số nhân như ở quan hệ P36, P413), tăng-giảm (tỉ lệ chênh lệch tăng đến một hệ số và sau đó giảm dần như ở quan hệ P101, P103). Phần lớn các quan hệ dữ liệu rơi vào trường hợp tăng-giảm, tỉ lệ chênh lệch âm phần lớn ở các quan hệ thuộc trường hợp giảm (P36, P413, P740).

Kết quả khảo sát thể hiện được kỳ vọng của phương pháp với mục tiêu tạo ra sự tương đồng tỷ lệ thuận giữa tầm quan trọng của nơ-ron trong quá trình dự đoán của mô hình và giá trị điểm phân bổ. Khi loại bỏ điểm kích hoạt của các nơ-ron tri thức, kết quả dự đoán nhãn đúng giảm đi rất nhiều và khi tăng hệ số nhân thì kết quả dự đoán tăng theo đáng kể. Đối với trường hợp tăng hệ số nhân thì sự thay đổi là không tuyến tính và phụ thuộc vào dữ liệu cụ thể. Đối với điểm phân bổ cơ sở, điểm số cao không đồng nghĩa rằng nơ-ron có ảnh hưởng đến kết quả dự đoán của mô hình.

Loại bỏ tri thức khỏi mô hình: Khảo sát hiệu suất của mô hình trước và sau quá trình loại bỏ tri thức được trình bày ở bảng 4.5. Hiệu suất của mô hình thay đổi phụ thuộc vào tri thức cụ thể được loại bỏ. Phần lớn tri thức giảm đi đáng kể sau quá trình loại bỏ biểu hiện thông qua tỉ lệ giảm độ chính xác (ACC) của mô hình và tỉ lệ tăng của độ đo perplexity (PPL). Độ chính xác của mô hình đối với quan hệ được loại bỏ giảm đi đáng kể (giảm từ 0.1% đến 73.3%) trong khi độ chính xác đối với các quan hệ còn lại không bị ảnh hưởng quá nhiều (giảm từ 0.0% đến 3.0%). Độ đo perplexity có xu hướng tăng mạnh ở quan hệ được loại bỏ (tăng từ 11.3% đến 222.5%), trong khi ở các quan hệ khác tăng nhẹ (tăng từ 0.3% đến 16.3%). Kết quả dựa trên phương pháp cơ sở phần lớn là các kết quả bất thường vì độ giảm perplexity sau quá trình loại bỏ tri thức. Kết quả cho thấy sự tương đồng giữa lượng tri thức được loại bỏ và hiệu suất của mô hình. Số lượng nơ-ron tri thức được loại bỏ càng nhiều đồng nghĩa với sự giảm hiệu suất của mô hình, cụ thể là tỉ lệ giảm độ chính xác và tỉ lệ tăng của độ đo perplexity đối với quan hệ xóa. Phương pháp thể hiện được sự hiệu quả khi đảm bảo kết quả dự đoán của mô hình trên các quan hệ khác thông qua tỉ lệ tăng không đáng kể ở độ đo perplexity.

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận

Đề tài này tập trung vào việc nghiên cứu và phát triển phương pháp xác định và xóa các nơ-ron tri thức trong mô hình Transformer, một bước quan trọng trong lĩnh vực Machine Unlearning. Mục tiêu của chúng tôi là cải thiện khả năng loại bỏ hoàn toàn dữ liệu cá nhân khỏi các mô hình học máy mà không cần phải huấn luyện lại mô hình từ đầu, đồng thời đảm bảo sự bảo mật và hiệu suất của hệ thống.

Kết quả thực nghiệm cho thấy phương pháp xác định và xóa nơ-ron tri thức dựa trên cơ chế tự chú ý và mạng nơ-ron truyền thẳng đã đạt được những kết quả đáng khích lệ trong việc duy trì hiệu suất của mô hình trên bài toán điền từ vào ô trống. Phương pháp này không chỉ giúp tiết kiệm tài nguyên bằng cách loại bỏ dữ liệu mà không cần huấn luyện lại mô hình, mà còn tăng cường tính bảo mật, giảm nguy cơ bị tấn công đối kháng và cải thiện niềm tin của người dùng vào các hệ thống trí tuệ nhân tạo. Việc áp dụng các phương pháp Machine Unlearning có thể đóng góp đáng kể vào việc phát triển các hệ thống AI có đạo đức, đồng thời bảo vệ quyền riêng tư của người dùng. Bằng cách cung cấp một giải pháp hiệu quả cho vấn đề này, nghiên cứu của chúng tôi không chỉ giúp duy trì hiệu suất của mô hình học máy mà còn đảm bảo sự bảo mật và riêng tư cho người dùng.

Những kết quả này mở ra hướng nghiên cứu tiếp theo trong việc tối ưu hóa và áp dụng các phương pháp Machine Unlearning vào các mô hình học máy khác, nhằm đáp ứng tốt hơn các yêu cầu về bảo mật và quyền riêng tư trong tương lai.

5.2 Hướng phát triển đề tài

Mặc dù nghiên cứu hiện tại đã đạt được những kết quả đáng khích lệ trong việc xác định và xóa các nơ-ron tri thức trong mô hình Transformer, song vẫn còn nhiều kiến thức cần được khám phá và phát triển để cải thiện hơn nữa khả năng và hiệu quả của phương pháp Machine Unlearning. Một số hướng phát triển có thể được xem xét bao gồm:

Làm rõ các kết quả bất thường: Một trong những hướng phát triển quan trọng là giải thích và làm rõ các kết quả bất thường của khảo sát thay đổi giá trị phân bổ và quá trình loại bỏ tri thức. Điều này bao gồm việc tìm hiểu yếu tố gây ra những bất thường và cải thiện độ tin cậy của phương pháp loại bỏ tri thức. Các bước nghiên cứu bao gồm:

- Phân tích các trường hợp bất thường trong quá trình thay đổi giá trị phân bổ để hiểu rõ nguyên nhân và tác động.
- Áp dụng các thực nghiệm để xác nhận các giải thích và cải thiện độ tin cậy của phương pháp.

Loại bỏ tri thức trên nhiều quan hệ dữ liệu: Một hướng nghiên cứu khác là mở rộng khả năng loại bỏ tri thức trên nhiều quan hệ dữ liệu khác nhau. Hiện tại, phương pháp nghiên cứu trong đề tài này tập trung loại bỏ một quan hệ dữ liệu nên cần phát triển các phương pháp có khả năng xử lý đa dạng các loại quan hệ dữ liệu. Các bước cụ thể bao gồm:

- Mở rộng tập dữ liệu thực nghiệm để bao gồm nhiều loại quan hệ khác nhau, từ đó đánh giá tính tổng quát của phương pháp.

- Đánh giá hiệu quả của các phương pháp trên các tập dữ liệu lớn và đa dạng.

Tối ưu hóa quá trình tính toán: Tối ưu hóa quá trình tính toán là một hướng phát triển quan trọng nhằm giảm thiểu tài nguyên và thời gian cần thiết cho quá trình loại bỏ tri thức. Điều này bao gồm việc cải tiến các thuật toán hiện có và phát triển các kỹ thuật mới để tăng cường hiệu quả tính toán. Các bước cụ thể bao gồm:

- Nghiên cứu và áp dụng các kỹ thuật tối ưu hóa nhằm giảm thiểu số lượng phép tính trong quá trình loại bỏ tri thức.
- Áp dụng các công nghệ tính toán song song để xử lý một cách hiệu quả.

Tài liệu tham khảo

Tiếng Anh

[1] Baevski, Alexei and Auli, Michael. "Adaptive Input Representations for Neural Language Modeling". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=ByxZX20qFQ.

[2] Binder, Alexander et al. *Layer-wise Relevance Propagation for Neural Networks with Local Renormalization Layers*. 2016. arXiv: 1604.00825 [cs.CV]. URL: https://arxiv.org/abs/1604.00825.

[3] Cao, Yinzhi and Yang, Junfeng. "Towards Making Systems Forget with Machine Unlearning". In: *2015 IEEE Symposium on Security and Privacy*. 2015, pp. 463–480. DOI: 10.1109/SP.2015.35.

[4] Chang, Yi et al. *Example-based Explanations with Adversarial Attacks for Respiratory Sound Analysis*. https://doi.org/10.48550/arXiv.2203.16141.

[5] Dai, Damai et al. "Knowledge Neurons in Pretrained Transformers". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*. 2022, pp. 8493–8502.

[6] Deng, Jia et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[7] Devlin, Jacob et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. https://doi.org/10.48550/arXiv.1810.04805.

[8] Elazar, Yanai et al. *Measuring and Improving Consistency in Pretrained Language Models*. https://doi.org/10.48550/arXiv.2102.01017.

[9] Elsahar, Hady et al. "T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Ed. by Calzolari, Nicoletta et al. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. URL: https://aclanthology.org/L18-1544.

[10] Geva, Mor et al. "Transformer Feed-Forward Layers Are Key-Value Memories". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2021.

[11] Hao, Yaru et al. "Self-Attention Attribution: Interpreting Information Interactions Inside Transformer". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.14 (2021), pp. 12963–12971. DOI: 10.1609/aaai.v35i14.17533. URL: https://ojs.aaai.org/index.php/AAAI/article/view/17533.

[12] Hu, Zizhao et al. *Evaluating NLP Systems On a Novel Cloze Task: Judging the Plausibility of Possible Fillers in Instructional Texts*. https://doi.org/10.48550/arXiv.2112.01867.

[13] Liu, Yinhan et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. https://doi.org/10.48550/arXiv.1907.11692.

[14] Merity, Stephen et al. "Pointer Sentinel Mixture Models". In: *International Conference on Learning Representations*. 2017. URL: https://openreview.net/forum?id=Byj72udxe.

[15] Miller, George et al. “Introduction to WordNet: An On-line Lexical Database*”. In: 3 (Jan. 1991). DOI: 10.1093/ijl/3.4.235.

[16] Nguyen, Thanh Tam et al. “A Survey of Machine Unlearning”. In: *arXiv preprint arXiv:2209.02299* (2022).

[17] Radford, Alec et al. *Improving Language Understanding by Generative Pre-Training*. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[18] Samek, Wojciech et al. “Evaluating the Visualization of What a Deep Neural Network Has Learned”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28.11 (2017), pp. 2660–2673. DOI: 10.1109/TNNLS.2016.2599820.

[19] Shrikumar, Avanti, Greenside, Peyton, and Kundaje, Anshul. *Learning Important Features Through Propagating Activation Differences*. 2019. arXiv: 1704.02685 [cs.CV]. URL: https://arxiv.org/abs/1704.02685.

[20] Siau, Keng and Wang, Weiyu. “Artificial Intelligence (AI) Ethics: Ethics of AI and Ethical AI”. In: *Journal of Database Management (JDM)* 31 (2020), pp. 74–87. ISSN: 1063-8016. DOI: 10.4018/JDM.2020040105. URL: https://www.researchgate.net/publication/340115931_Artificial_Intelligence_AI_Ethics_Ethics_of_AI_and_Ethical_AI.

[21] Sundararajan, Mukund, Taly, Ankur, and Yan, Qiqi. “Axiomatic Attribution for Deep Networks”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Precup, Doina and Teh, Yee Whye. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3319–3328. URL: https://proceedings.mlr.press/v70/sundararajan17a.html.

[22] Vaswani, Ashish et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by Guyon, I. et al.

Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4Paper.pdf.

[23] Zhu, Yukun et al. *Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books*. 2015. arXiv: 1506.06724 [cs.CV]. URL: https://arxiv.org/abs/1506.06724.