ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

BÁO CÁO
ĐỒ ÁN TỐT NGHIỆP (CO4337)

Khám phá thực thể dựa trên ý định cho nhiệm vụ xây dựng đồ thị tri thức trong lĩnh vực giáo dục

---

Ngành: Khoa học máy tính

HỘI ĐỒNG:	Hội đồng 3 Khoa học máy tính
GIẢNG VIÊN HƯỚNG DẪN:	PGS. TS. Bùi Hoài Thắng ThS. Bùi Công Tuấn
CHỦ TỊCH HỘI ĐỒNG:	TS. Nguyễn An Khương
THƯ KÝ HỘI ĐỒNG:	ThS. Trần Ngọc Bảo Duy
ỦY VIÊN HỘI ĐỒNG:	TS. Trần Tuấn Anh
SINH VIÊN THỰC HIỆN 1:	Nguyễn Minh Khỏe (2011438)
SINH VIÊN THỰC HIỆN 2:	Võ Ngọc Duy Nghiêm (2220036)

Tp Hồ Chí Minh, Tháng 9/2024

Ngày:

PGS.TS Bùi Hoài Thắng (Giảng viên hướng dẫn 1)
Phó giáo sư - Tiến sĩ
Khoa Khoa học và Kỹ thuật Máy tính

Ngày:

ThS. Bùi Công Tuấn (Giảng viên hướng dẫn 2)
Nghiên cứu sinh - Thạc sĩ
Khoa Khoa học và Kỹ thuật Máy tính

Lời tri ân

Đầu tiên nhóm chúng tôi xin gửi lời cảm ơn chân thành đến thầy Bùi Hoài Thắng và thầy Bùi Công Tuấn, giáo viên hướng dẫn của nhóm. Các thầy đã hướng dẫn và giúp đỡ nhóm rất tận tình trong quá trình làm Đồ án tốt nghiệp. Chúng tôi đã nhận được những lời nhận xét, góp ý thẳng thắn và những lời khuyên bổ ích trong các buổi báo cáo định kỳ. Sự hướng dẫn và định hướng của các thầy là yếu tố vô cùng quan trọng để nhóm có thể thực hiện đề tài này.

Nhóm cũng xin cảm ơn đến các thầy cô đang công tác và giảng dạy tại Khoa Khoa học và Kỹ thuật Máy Tính - Trường Đại học Bách khoa thành phố Hồ Chí Minh. Các thầy cô đã tận tình dạy bảo, truyền đạt kiến thức, kỹ năng cần thiết để nhóm có thể thực hiện được đề tài.

Ngoài ra, nhóm chúng tôi xin gửi lời cảm ơn đến các tác giả, nhà nghiên cứu mà chúng tôi có dịp tham khảo công trình nghiên cứu, bài báo khoa học cũng như phần hiện thực (gồm các đoạn mã, mô hình tiền huấn luyện, v.v...) đã được trích dẫn trong đề tài này.

Một lần nữa, nhóm xin chân thành cảm ơn tất cả những người đã đóng góp và ủng hộ chúng tôi trong quá trình nghiên cứu và hoàn thiện đồ án này.

Xin chân thành cảm ơn!

Lời cam đoan

Chúng tôi xin cam đoan tất cả nội dung trình bày trong đồ án chuyên ngành này, ngoại trừ những phần đã được trích dẫn từ các nguồn tham khảo, đều là thành quả nghiên cứu của cả nhóm dưới sự hướng dẫn của Phó giáo sư - Tiến sĩ Bùi Hoài Thắng và Thạc sĩ Bùi Công Tuấn.

Tất cả các tài liệu được tham khảo, trích dẫn để hoàn thành đồ án này đều được liệt kê đầy đủ trong danh mục "Tài liệu tham khảo".

Chúng tôi xin hứa sẽ hoàn toàn chịu trách nhiệm và mọi hình thức kỷ luật trước bất kì vi phạm nào gây ra bởi nội dung đồ án này, nếu có.

Tóm tắt

Với sự phát triển nhanh chóng của trí tuệ nhân tạo ngày nay, nhu cầu sử dụng các trợ lý ảo, chatbot thay cho con người trong việc hỗ trợ sinh viên ở các trường đại học đang xuất hiện. Đồ thị tri thức (Knowledge Graph) đóng vai trò quan trọng trong việc tạo ra một cấu trúc dữ liệu có thể tìm kiếm và truy xuất thông tin chính xác. Đồng thời, nhiệm vụ phát hiện ý định (Intent Detection) và điền thông tin vào các trường (Slot Filling) cũng là một phần quan trọng giúp hệ thống hiểu rõ yêu cầu của sinh viên.

Trong đề tài này, nhóm đã áp dụng các phương pháp rút trích và gom cụm thực thể để khám phá các kiểu thực thể trong miền giáo dục để nhận dạng các thực thể trong dữ liệu giáo dục, sau đó sử dụng các luật kết hợp (association rules) để xây dựng lược đồ cho đồ thị tri thức. Ngoài ra, nhóm thử nghiệm phương pháp đọc hiểu máy (MRC) để giải quyết bài toán phát hiện ý định và điền thông tin vào các trường. Đồ thị tri thức được xây dựng cũng hỗ trợ cho nhiệm vụ này.

Đóng góp chính của nghiên cứu bao gồm việc đề xuất một hướng tiếp cận đầu tiên cho bài toán xây dựng đồ thị tri thức theo hướng bottom-up, trong đó kiểu thực thể phải được khám phá từ dữ liệu thay vì được xác định trước. Đồng thời, nhóm cũng áp dụng phương pháp đọc hiểu máy tiên tiến vào bài toán phát hiện ý định và điền thông tin vào các trường.

Mục lục

Danh sách hình vẽ

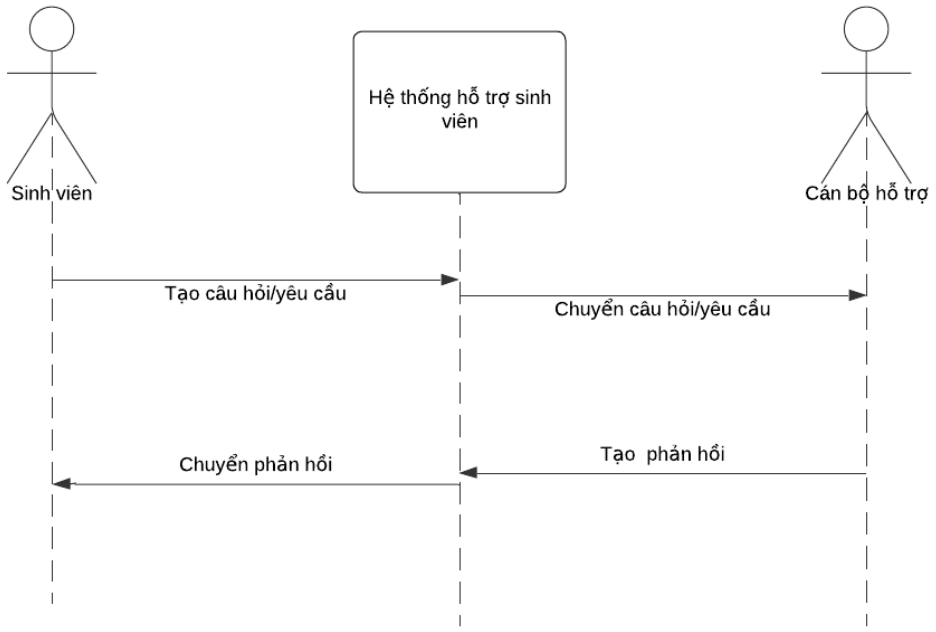
Danh sách bảng

1 Giới thiệu

1.1 Đặt vấn đề

Trong các trường đại học hiện nay, các dịch vụ hỗ trợ sinh viên đóng vai trò quan trọng trong việc giải quyết các thắc mắc và xử lý các yêu cầu của sinh viên liên quan đến các vấn đề hành chính, học thuật và cá nhân. Những dịch vụ này hoạt động như một cầu nối giữa sinh viên và nhà trường, giúp giải quyết các vấn đề như đăng ký môn học, hỗ trợ tài chính, điều chỉnh khóa học và các yêu cầu khác.

Hệ thống hoạt động như một kênh trò chuyện giữa sinh viên và người trả lời. Đầu tiên, sinh viên có nhu cầu được hỗ trợ, giải đáp thắc mắc sẽ tạo câu hỏi/yêu cầu và gửi đến hệ thống. Hệ thống sẽ chuyển câu hỏi/yêu cầu của sinh viên đến người có chuyên môn thích hợp phụ trách trả lời câu hỏi. Người chuyên viên này sau khi tiếp nhận câu hỏi/yêu cầu sẽ đưa ra phản hồi chính xác nhằm giải đáp vấn đề sinh viên đang gặp phải và gửi lại cho hệ thống. Cuối cùng hệ thống sẽ chuyển lại phản hồi đó cho sinh viên. Quy trình được mô tả trực quan ở sơ đồ trong hình 1.



Hình 1: Sơ đồ luồng xử lý của hệ thống hỗ trợ sinh viên

Cụ thể, ở hệ thống BKSI - kênh hỗ trợ sinh viên của Trường Đại học Bách

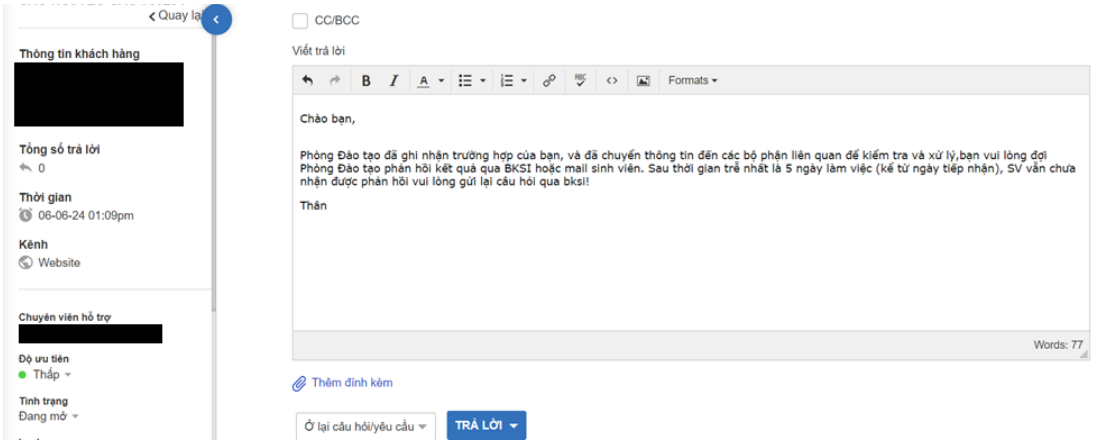
Khoa - Đại học Quốc gia Thành phố Hồ Chí Minh, sinh viên đặt câu hỏi bằng cách chọn loại câu hỏi/yêu cầu, nhập chủ đề và nội dung câu hỏi, sau đó gửi câu hỏi/yêu cầu đi (hình 2).

Hình 2: Giao diện BKSI từ phía sinh viên

Các yêu cầu của sinh viên được tiếp nhận và xử lý thông qua những cán bộ phụ trách trong trường học. Giao diện của cán bộ trả lời cho sinh viên được mô tả như hình 3 dưới đây. Một câu hỏi của sinh viên bao gồm chủ đề (dòng chữ lớn) và nội dung. Khi sinh viên gửi một yêu cầu mới, hệ thống sẽ tạo ra một ticket. Đây là một luồng thông tin giúp sinh viên và người trả lời có thể gửi tin cho nhau như đoạn hội thoại. Việc trả lời câu hỏi diễn ra trong ticket. Nếu sinh viên tiếp tục có thắc mắc với vấn đề liên quan hoặc chưa hài lòng với câu trả lời, có thể tiếp tục gửi yêu cầu trong ticket này.

Các cán bộ phải đọc kỹ và hiểu nội dung câu hỏi được gửi đến từ sinh viên, đồng thời cũng cần phải có một lượng hiểu biết lớn về các quy chế, quy định và một số kiến thức nghiệp vụ khác của Nhà trường để trả lời các câu hỏi một cách chính xác. Hình 4 là hình ảnh giao diện của cán bộ hỗ trợ sinh viên. Người cán bộ đang trong quá trình trả lời câu hỏi.

Hình 3: Hệ thống BKSI (Giao diện của cán bộ tiếp nhận câu hỏi)



Hình 4: Cán bộ phụ trách nhập nội dung trả lời câu hỏi cho sinh viên

Việc nhớ được một khối lượng kiến thức về quy chế, quy định cùng với toàn bộ nghiệp vụ của nhà trường là rất khó khăn, thậm chí bất khả thi. Những người phụ trách trả lời câu hỏi có thể sẽ phải đọc lại các tài liệu (trong trường hợp người cán bộ muốn trả lời chi tiết cho sinh viên, chứ không chỉ đưa đường dẫn quy chế quy định cho sinh viên tự tìm hiểu), tổng hợp và kiểm tra kỹ càng trước khi đưa ra câu trả lời của mình. Vì thế, công việc này có thể tiêu tốn nhiều thời gian và công sức của chuyên viên phụ trách, cũng như có thể xảy ra nhầm lẫn khó tránh khỏi mà đưa ra những câu trả lời sai, trong khi các kênh hỗ trợ này yêu cầu sự chính xác và nhanh chóng.

Ngoài ra, các câu hỏi dài của sinh viên cũng có khả năng gây ra nhiều trở ngại

cho người phụ trách khi trả lời các câu hỏi đó. Các câu hỏi dài này của sinh viên có thể chứa rất nhiều thông tin lan man hoặc thậm chí không rõ ý định, gây lãng phí thời gian không cần thiết cho người đọc (hình 5). Vì vậy, việc tìm ra ý định và những thông tin liên quan đến ý định từ yêu cầu của sinh viên sẽ tiết kiệm được rất nhiều thời gian của chuyên viên trả lời. Khi nắm được các thông tin này, người cán bộ hoàn toàn có thể biết được toàn bộ nội dung câu hỏi của sinh viên, và có thể phản hồi một cách nhanh chóng.

Hình 5: Câu hỏi dài của sinh viên

Số lượng yêu cầu và thắc mắc được sinh viên gửi đến là quá lớn, các trường đại học phải đối mặt với thách thức trong việc quản lý và phản hồi các thắc mắc này một cách hiệu quả và kịp thời. Vì thế hiện nay các trường đại học đang có mong muốn chuyển dần các dịch vụ hỗ trợ sinh viên do con người phụ trách sang cho máy tính thông minh đảm nhiệm, chẳng hạn như trợ lý ảo, chatbot. Nền tảng của hệ thống thông minh là đồ thị tri thức dùng để quản lý các kiến thức về nghiệp vụ của trường học cũng như việc phát hiện ý định và thông tin liên quan trong câu hỏi, yêu cầu của sinh viên.

1.2 Động lực

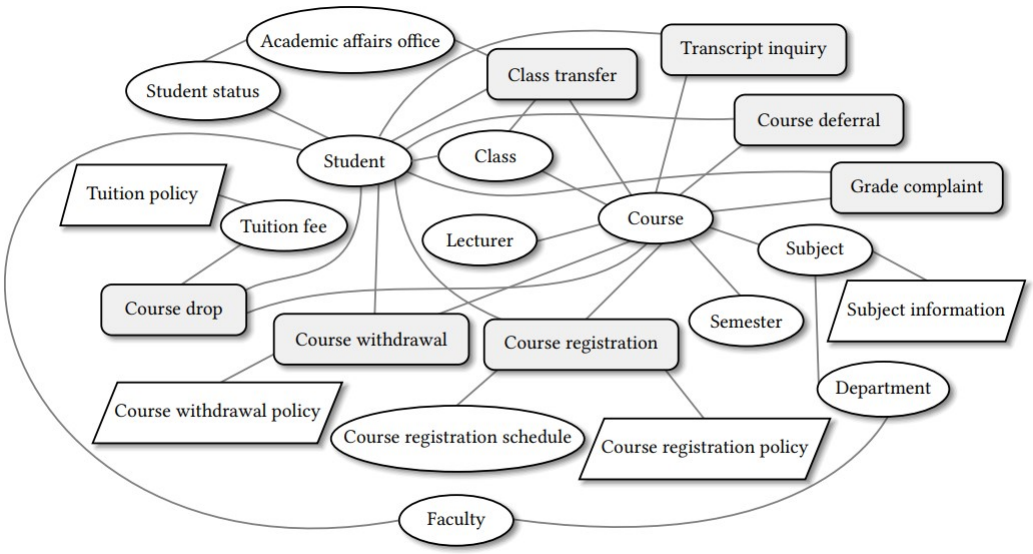
Trợ lý ảo đang ngày càng phổ biến trong thế giới hiện đại nên việc nâng cao hiệu quả hoạt động của chúng trở nên thiết yếu. Yếu tố then chốt quyết định sự thành công của các trợ lý ảo này chính là khả năng nắm bắt và hiểu đúng yêu cầu của người dùng, từ đó đưa ra hành động hoặc phản hồi thỏa đáng, đáp ứng đúng nhu cầu mà họ mong đợi. Trong lĩnh vực xử lý ngôn ngữ, nội dung yêu cầu của người dùng có thể tóm tắt dưới dạng một khung ngữ nghĩa (semantic frame) bao gồm thông tin về ý định và các thông tin khác có trong câu [49]. Trong hệ thống hỗ trợ sinh viên, việc nhận diện ý định và các thông tin liên quan cũng vô cùng quan trọng. Ngoài việc nó có vai trò cốt lõi khi hiện thực một hệ thống tự động,

trước hết nó cũng giúp ích cho các cán bộ tiếp nhận câu hỏi. Thay vì phải đọc những đoạn nội dung dài trong câu hỏi, người cán bộ chỉ cần tiếp nhận thông tin ngắn gọn trong khung ngữ nghĩa, từ đó có thể hiểu được toàn bộ nội dung yêu cầu để cho phản hồi thích hợp. Vì vậy, việc thực hiện nhiệm vụ Phát hiện ý định và Điền trường thông tin trong đề tài có ý nghĩa thực tiễn.

Một vấn đề khác của hệ thống hỗ trợ sinh viên là một nền tảng để không chỉ lưu trữ kiến thức và còn phải dễ dàng tìm kiếm và truy xuất. Công nghệ hiện nay có thể thỏa mãn được điều đó chính là đồ thị tri thức. Đồ thị tri thức vốn bắt nguồn từ công nghệ Semantic Web (Web ngữ nghĩa) dùng để biểu diễn tất cả những hiểu biết trong một lĩnh vực hay một vấn đề cụ thể nào đó dưới dạng thực thể và các mối quan hệ giữa chúng. Trong lĩnh vực giáo dục, đồ thị tri thức có thể tích hợp thông tin từ các cơ sở dữ liệu khác nhau của trường đại học, chẳng hạn như hồ sơ sinh viên, các khóa học được cung cấp, quy chế quy định,... vào một định dạng chung và có cấu trúc. Việc các chatbot, hệ thống hỏi đáp sử dụng đồ thị tri thức đang rất phổ biến hiện nay, vì nó có thể truy xuất các thông tin liên quan, cung cấp phản hồi chính xác hơn, giảm thiểu tình trạng "hallucination" (hiện tượng một mô hình sinh ngôn ngữ có khả năng tạo ra những thông tin không có cơ sở thực tế) ở các mô hình ngôn ngữ lớn (LLM).

Đồ thị tri thức dùng cho hệ thống hỗ trợ sinh viên không chỉ chứa những thông tin kiến thức nghiệp vụ hay quy chế quy định khác của nhà trường mà còn nên chứa thông tin về ý định của sinh viên, nhằm giúp hệ thống dễ dàng suy luận phản hồi thông qua các thông tin liên quan đến ý định. Lấy ví dụ, một bạn sinh viên muốn đăng ký luận văn tốt nghiệp vào học kì 231. Ta có thông tin của bạn sinh viên và cả thời điểm bạn đăng ký luận văn. Với thông tin của sinh viên này lưu trên đồ thị tri thức, ta biết được điểm anh văn của bạn, cùng với thông tin quy chế quy định của nhà trường trên đồ thị, hệ thống có thể kết luận bạn đạt chuẩn anh văn để được đăng ký luận văn hay không và đưa ra phản hồi tương ứng. Với ví dụ trên, ta thấy thông tin khung ngữ nghĩa (semantic frame) gồm ý định và các thông tin liên qua (còn gọi là các trường - slot) nên được tích hợp vào trong đồ thị tri thức giúp hệ thông hoạt động hiệu quả. Cùng ý tưởng này, bài báo "Cross-Data Knowledge Graph Construction for LLM-enabled Educational Question-Answering System: A Case Study at HCMUT" [4] cũng đã xây dựng đồ thị tri thức gồm ý định được xem như là thực thể, và mối quan hệ giữa ý định và các quy chế quy định. Tiếp nối ý tưởng này, nhóm thực hiện đề tài cũng thực hiện xây dựng đồ thị tri thức chứa thông tin ý định, nhưng xem chúng như mối quan hệ giữa các thực thể. Do cùng lĩnh vực và phạm vi nghiên cứu, nhóm có sử dụng lại một số kết quả của bài báo. Hình 6 dưới đây mô tả lược đồ đồ thị tri thức của nghiên cứu trên. Trong đó, các khối hình chữ nhật cạnh tròn chỉ các thực thể ý

định; các khối hình bình hành chỉ các thực thể là quy chế quy định; và các khối oval là các thực thể còn lại trong trong trường.



Hình 6: Đồ thị tri thức bao gồm ý định trong lĩnh vực giáo dục [4]

Rất nhiều nghiên cứu cho đến nay tiếp cận nhiệm vụ xây dựng đồ thị tri thức theo hướng top-down (rút trích tri thức theo một lược đồ cho trước). Hướng tiệp cận này cần nhiều nỗ lực từ những chuyên gia hiểu biết sâu về lĩnh vực mà ta xây dựng đồ thị để thiết kế ra một lược đồ chính xác, vững chắc, đầy đủ làm khung cho đồ thị. Nhưng trong những tình huống thực tế, việc có sẵn những chuyên gia này là điều không chắc chắn. Dựa trên vấn đề này, nhiều nghiên cứu hiện nay đang theo hướng tiếp cận bottom-up (rút trích thông tin trước rồi tổng hợp lại thành lược đồ). Tuy đánh đổi một phần chất lượng, nhưng quá trình sinh đồ thị theo hướng này có thể thực hiện tự động và không cần hoặc chỉ cần ít sự tham gia của con người, khiến nó trở thành một hướng đi hứa hẹn trong tương lai. Các nghiên cứu xây dựng đồ thị tri thức theo hướng bottom-up cho tới hiện nay vẫn chỉ giới hạn trong việc khám phá ra các kiểu quan hệ, các kiểu thực thể vẫn phải được cho từ trước từ một người chuyên gia nào đó. Trong đề tài, nhóm sẽ thử nghiệm hướng tiếp cận bottom-up và khám phá kiểu thực thể để xây dựng đồ thị tri thức.

1.3 Mục tiêu của đề tài

Dựa trên vấn đề đã nêu, nhóm nghiên cứu hướng đến hai mục tiêu chính, đó là:

- Đề xuất framework để xây dựng lược đồ cho đồ thị tri thức trong lĩnh vực giáo dục tập trung vào ý định theo hướng bottom-up.
- Đề xuất phương pháp cho nhiệm vụ "Nhận diện ý định và điền trường thông tin".

1.4 Phạm vi đề tài

- Tuy một đồ thị tri thức đầy đủ phải chứa cả các mối quan hệ giữa các thực thể và các thuộc tính có liên quan, trong phạm vi thực hiện đề tài, này nhóm chỉ trích các khung ngữ nghĩa, được mô hình thành các bộ quan hệ trong đồ thị tri thức bao gồm ý định và các thực thể tham gia vào nó. Các mối quan hệ khác, và những thực thể không liên quan đến ý định, cũng như các quy chế quy định sẽ nằm ngoài phạm vi nghiên cứu của nhóm.
- Đồ thị tri thức nhóm xây dựng chỉ là đồ thị thô được tạo nên nhờ rút trích thông tin từ tập dữ liệu, sẽ chưa qua các giai đoạn tinh chỉnh để cho ra đồ thị tri thức hoàn thiện.
- Dữ liệu nhóm sử dụng trong đề tài này đều lấy hệ thống hỏi đáp BKSI và các nguồn dữ liệu công khai khác của trường Đại học Bách Khoa - Đại học quốc gia thành phố Hồ Chí Minh, không bao gồm các nguồn dữ liệu giáo dục khác.

Các vấn đề ngoài phạm vi đề tài sẽ tiến hành trong các công trình nghiên cứu sau này.

1.5 Bố cục bài viết

Phần còn lại của bản báo cáo này có cấu trúc như sau:

Chương 2: Cơ sở lý thuyết
Chương này trình bày những điểm lý thuyết mà nhóm sử dụng cho đề tài. Nội dung chương này bao gồm những kiến thức liên quan đến Đồ thị tri thức, Nhận diện thực thể, Phân cụm ngữ nghĩa, Nhận diện ý định và Điền thông tin trường, Đọc hiểu máy và nhiều vấn đề khác thuộc lĩnh vực Học máy và Xử lý ngôn ngữ tự nhiên. Các kiến trúc mô hình được sử dụng cũng được trình bày trong phần này.

Chương 3: Các công trình liên quan
Chương này trình bày tổng quan các nghiên cứu trước đây cho các nhiệm vụ trong đề tài bao gồm Xây dựng đồ thị tri thức, Nhận diện thực thể đặt tên, Trích xuất thông tin mở, Phát hiện ý định và Điền trường thông tin, Đọc hiểu máy. Nhóm

tập trung chú ý đến những nghiên cứu trong lĩnh vực giáo dục và ngôn ngữ Việt. Nhóm sẽ đề xuất phương pháp và mô hình sử dụng dựa trên các nghiên cứu này.

Chương 4: Phương pháp
Chương này trình bày chi tiết về framework mà nhóm xây dựng cũng như phương pháp mà nhóm sử dụng bao gồm tổng thể quy trình và chi tiết các bước xử lý.

Chương 5: Thí nghiệm
Chương này trình bày chi tiết về dữ liệu mà nhóm sử dụng, cách thiết lập các thí nghiệm và kết quả đánh giá các thí nghiệm này. Nhóm cũng phân tích và đánh giá các kết quả thí nghiệm.

Chương 6: Tổng kết
Trong chương này, nhóm sẽ trình bày khái quát những kết quả mà nhóm đạt được so với mục tiêu ban đầu gồm những gì đã hoàn thành và những gì cần cải thiện thêm. Bảng phân chia khối lượng công việc của các thành viên trong nhóm cũng sẽ được công bố trong chương.

2 Cơ sở lý thuyết

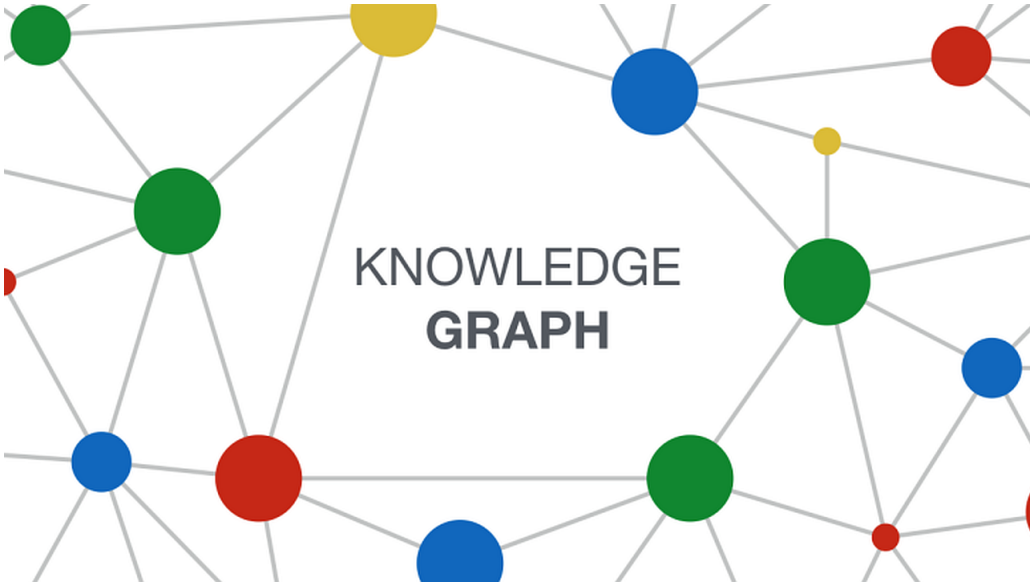
Trước khi đi vào phần phương pháp mà nhóm thực hiện, nhóm sẽ trình bày tóm gọn những nội dung lý thuyết đã tìm hiểu để sử dụng trong đề tài. Trước tiên là phần nội dung quan trọng "Đồ thị tri thức", trình bày khái niệm, cấu trúc một đồ thị tri thức, các bước xây dựng đồ thị tri thức. Vì nhóm sử dụng kĩ thuật Nhận diện thực thể đặt tên, gom cụm trong quá trình trích xuất tri thức nên phần nội dung lý thuyết của những chủ đề này cũng sẽ được trình bày. Tiếp đến là nội dung lý thuyết cho bài toán "Phân loại ý định và Điền trường thông tin" và lý thuyết "Đọc hiểu máy" - phương pháp mà nhóm áp dụng trong bài toán này. Kế đến là một số nội dung trình bày các kiến trúc mô hình học máy mà nhóm sử dụng. Cuối cùng nhóm điểm qua lý thuyết "ma trận nhầm lẫn", để hiểu qua những độ đo dùng để đánh giá phương pháp trong đề tài.

2.1 Đồ thị tri thức

Khái niệm "đồ thị tri thức" lần đầu tiên được giới thiệu vào năm 1972 bởi nhà ngôn ngữ học người Úc Edgar W. Schneider trong một cuộc thảo luận về cách xây dựng hệ thống hướng dẫn mô-đun cho các khóa học [42], nhưng phải đến sau năm 2012, khi Google ra mắt đồ thị tri thức của mình của mình mang tên "Google Knowledge Graph" để nâng cấp công cụ tìm kiếm của họ, khái niệm này mới được sử dụng rộng rãi. Từ đó càng nhiều đồ thị tri thức được phát triển và được ứng dụng ngày càng tích cực trong nhiều ngành công nghiệp khác nhau.

Tuy đồ thị tri thức được nghiên cứu nhiều là thế, nhưng người ta vẫn khá mơ hồ khi đưa ra một định nghĩa chính xác cho khái niệm này. Nhiều nỗ lực đưa ra để mô tả một đồ thị tri thức là gì. Trong đa số các nghiên cứu, các nhà khoa học cho rằng một đồ thị tri thức là một dạng cấu trúc được thiết kế để tích lũy và truyền tải tri thức về thế giới thực, trong đó các nút biểu thị các thực thể quan tâm và các cạnh biểu thị mối quan hệ giữa các thực thể này. Về hình thức, một đồ thị tri thức là một đồ thị có hướng (G), được định nghĩa là G = (V, E) [1], trong đó V đại diện cho các đỉnh hoặc nút tương ứng với các thực thể trong thế giới thực, và E đại diện cho các cạnh hoặc liên kết biểu thị mối quan hệ giữa các thực thể này. Thông thường, các thực thể và mối quan hệ của chúng được biểu diễn dưới dạng bộ ba (chủ ngữ, vị ngữ, và tân ngữ)[1] và được hình dung dưới dạng đồ thị như hình 8. Trong hình, mỗi màu của hình oval tương ứng với một kiểu thực thể, mỗi hình oval là thực thể. Mũi tên có hướng biểu diễn mối quan hệ giữa các thực thể đó.

Một số đồ thị tri thức cho phép quan hệ nhiều ngôi (n-ary relation) thay vì chỉ quan hệ hai ngôi (binary relation) [56]. Lúc này, bộ ba mô tả quan hệ có dạng



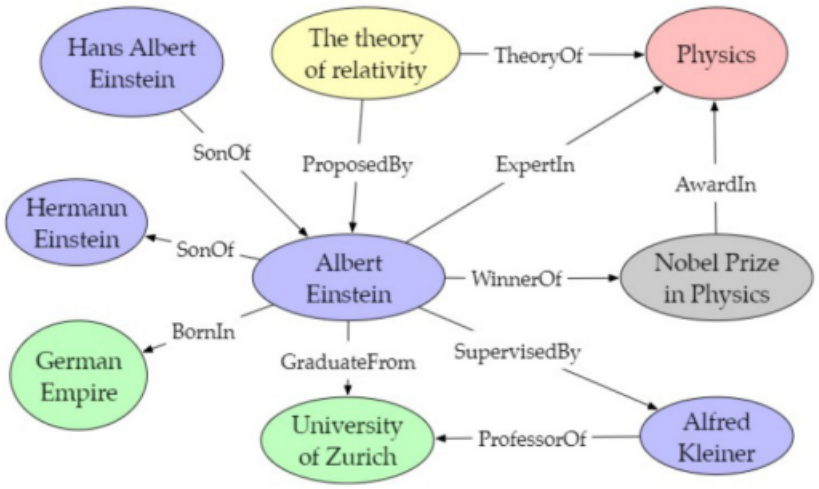
Hình 7: Đồ thị tri thức

$(r, e_1, e_2, ...)$ với các e là thực thể tham gia vào mối quan hệ r.

Đồ thị tri thức được sử dụng cho nhiều mục đích khác nhau, chẳng hạn như tìm kiếm và truy vấn (như Google và Bing), hỗ trợ trợ lý ảo hay chatbot (như ChatGPT), đóng vai trò là cơ sở dữ liệu ngữ nghĩa (như Wikidata), và phân tích dữ liệu lớn (ví dụ như Walmart). Đồ thị tri thức cũng được ứng dụng nhiều trong các lĩnh vực chuyên môn và trong các ngành công nghiệp, chẳng hạn như y học và chăm sóc sức khỏe, tài chính và ngân hàng, IoT, an ninh mạng, truyền thông và giải trí, thương mại điện tử và bán lẻ, giáo dục và một số lĩnh vực học thuật, khoa học, công nghiệp khác.

Tùy vào tính chất của tri thức lưu trong đồ thị, đồ thị tri thức có thể phân chia thành nhiều loại. Tuy nhiên, trong nhiều thảo luận học thuật, đồ thị tri thức thường được phân thành hai loại chính: đồ thị tri thức chung và đồ thị tri thức chuyên biệt. Đồ thị tri thức chung bao phủ nhiều lĩnh vực và thường bao gồm nội dung bách khoa toàn thư, ví dụ như Wikidata [46], YAGO [43], và DBpedia [3]. Ngược lại, đồ thị tri thức chuyên biệt tập trung vào một lĩnh vực cụ thể hoặc ngành công nghiệp, thường được thiết kế cho các vấn đề hoặc lĩnh vực cụ thể.

Lingfeng Zhong và các đồng nghiệp [56] đã tổng hợp từ nhiều nguồn khác nhau và đề xuất một quy trình chung để xây dựng đồ thị tri thức như hình 9.



Hình 8: Một ví dụ minh họa trực quan đồ thị tri thức [19]

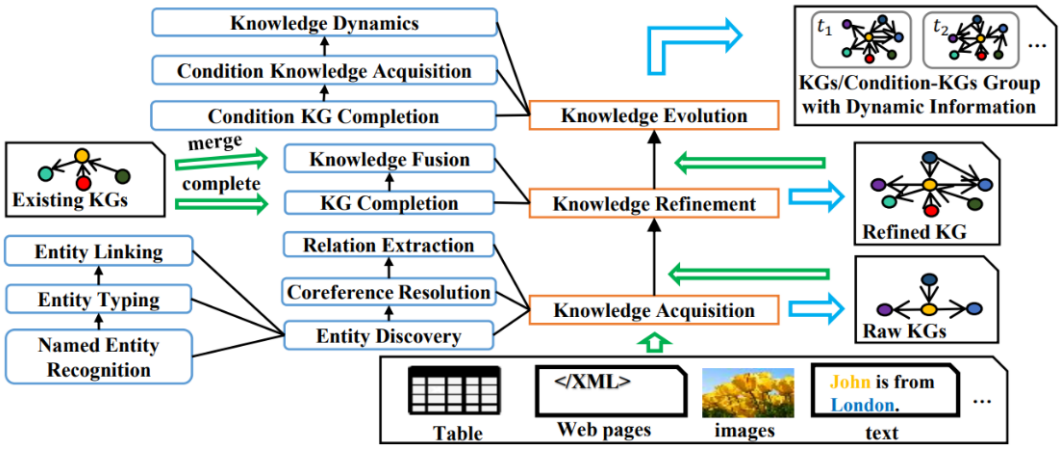
Dữ liệu để xây dựng đồ thị tri thức có thể đến từ nhiều nguồn với nhiều định dạng như trang web, bảng tính, hình ảnh, cơ sở dữ liệu, văn bản,... và có tính chất khác nhau như có cấu trúc, bán cấu trúc và phi cấu trúc đi qua ba giai đoạn chính là Knowledge Acquisition, Knowledge Refinement, Knowledge Evolution. Mỗi giai đoạn sẽ cho ra kết quả là đồ thị tri thức ở dạng khác nhau.

Knowledge Acquisition:
Bước này nhận đầu vào là dữ liệu thực hiện rút trích thông tin. Đối với dữ liệu phi cấu trúc, việc rút trích cần các kĩ thuật Xử lý ngôn ngữ tự nhiên phức tạp, thường phải trải qua các bước sau:

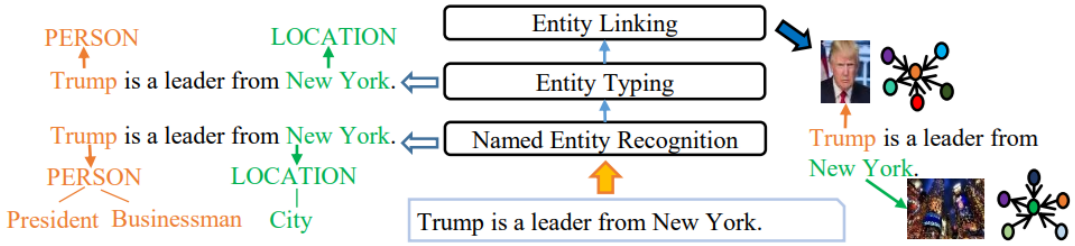
- *Entity Discovery*: xác định các khái niệm (concepts) trong dữ liệu có thể tạo thành node trong đồ thị, gồm ba bước nhỏ hơn là Named Entity Recognition - xác định các thực thể có tên trong dữ liệu và phân loại vào kiểu thực thể sơ bộ như người, tổ chức,...; Entity Typing - xác định kiểu thực thể chi tiết của các thực thể vừa nhận diện; Entity Linking - liên kết thực thể khám phá được với một node trên đồ thị tri thức (nếu không có sẽ tạo ra node mới). Xem hình 10.

- *Coreference Resolution*: là quá trình xác định các đề cập (mention - tức là một cụm từ chỉ một thực thể nào đó) có cùng tham chiếu đến các thực thể giống nhau hay không. (xem hình 11)

- *Relation Extraction*: là việc trích các mối quan hệ giữa các thực thể có trong dữ liệu (xem hình 12). Thông thường, nhiệm vụ này sẽ phân loại luôn cho quan hệ vừa trích được vào kiểu quan hệ được định sẵn. Nhưng nếu không



Hình 9: Quy trình chung để xây dựng đồ thị tri thức [56]



Hình 10: Quy trình khám phá thực thể [56]

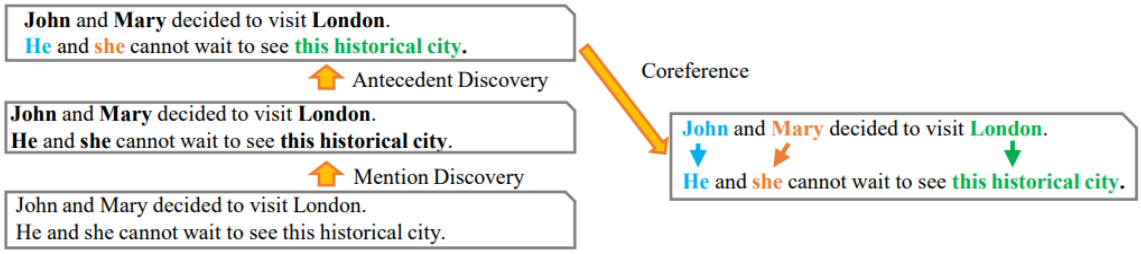
có sẵn lược đồ, đây sẽ là dạng bài toán Open Relation Extraction (rút trích quan hệ mở).

Sau khi kết thúc, giai đoạn này sẽ cho ra Raw KG tức đồ thị tri thức được tạo ra từ các bộ ba, vẫn còn sai sót và chưa hoàn chỉnh.

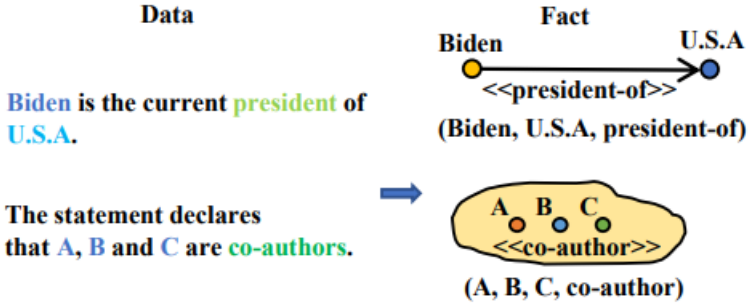
Hai giai đoạn sau không nằm trong phạm vi đề tài của nhóm, nên nhóm chỉ viết tổng quan những thông tin đã đọc được, mục đích để tham khảo thêm.

Knowledge Refinement:
Đây là bước sẽ hoàn thiện Raw KG được tạo ra từ giai đoạn trước bằng cách tham khảo đến một KG (đồ thị tri thức) đã tồn tại cho bài toán này hoặc vấn đề tương tự.

- *Knowledge Graph Completion*: dẫn xuất thêm bộ ba mới bằng những bộ ba đã đầy đủ để hoàn thiện những bộ ba chưa đầy đủ.
- *Knowledge Fusion*: vì yêu cầu thực tiễn luôn thay đổi, kiến thức thế giới bên



Hình 11: Quy trình giải quyết đồng tham chiếu [56]



Hình 12: Trích xuất quan hệ [56]

ngoài luôn luôn cập nhật nên trong hầu hết các trường hợp, đồ thị tri thức ngoại được thêm vào để làm giàu cho đồ thị tri thức đang tồn tại. Đây là hình thức sáp nhập các đồ thị lại với nhau.

Kết quả của giai đoạn này là một đồ thị tri thức đã được tinh chỉnh.

Knowledge Evolution:
Đây là giai đoạn nâng cao hình thức của đồ thị tri thức hơn nữa gồm các bước *Condition KG Completion*, *Condition Knowledge Acquisition* và *Knowledge Dynamics*. Kết quả của giai đoạn này là một đồ thị tri thức đã được nâng cao hơn.

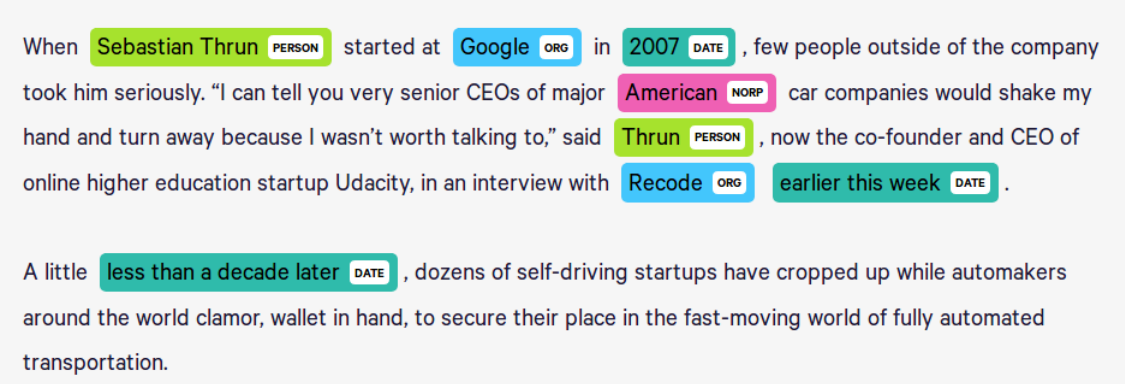
Quá trình phát triển đồ thị tri thức thường được chia thành hai phương pháp: từ trên xuống (top-down) và từ dưới lên (bottom-up). Với phương pháp từ trên xuống, trước tiên người ta xây dựng ontology(hay lược đồ tri thức) trước, sau đó trích xuất tri thức dựa trên ontology này [24]. Ngược lại, phương pháp từ dưới lên bắt đầu bằng việc trích xuất tri thức trực tiếp từ dữ liệu, rồi dựa vào đó để xây dựng ontology của đồ thị tri thức [24].

Nhiều cách thức, công cụ đã được phát triển để lưu trữ đồ thị tri thức. Thời gian đầu, các đồ thị được lưu trữ trong các hệ quản trị cơ sở dữ liệu quan hệ

(RDBMS) vì tính đáng tin cậy của nó và các toán tử CRUD giúp truy xuất thuận tiện. Tuy nhiên cách thức này lại rất tốn kém, đặc biệt là đối với các đồ thị tri thức thưa. Các cơ sở dữ liệu phi quan hệ như key/value cũng được sử dụng để lưu đồ thị tri thức (Trinity, Probase, CouchDB,...). Những chiều hướng hứa hẹn cho công nghệ này là cơ sở dữ liệu tài liệu (document) như NoSQL hay cơ sở dữ liệu đồ thị (Neo4j). Một số ngôn ngữ được phát triển để lưu trữ các đồ thị tri thức, chẳng hạn RDF hay OWL. Triple store - một cơ sở dữ liệu dùng để lưu bộ ba của đồ thị được phát triển dựa trên những ngôn ngữ này.

2.2 Nhận diện thực thể có tên

Nhận diện thực thể có tên (Named Entity Recognition hay NER) là một trong những nhiệm vụ quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên. Mục tiêu của NER là xác định và phân loại các thực thể có tên như người, tổ chức, địa điểm, ngày tháng, số lượng và các thực thể quan trọng khác từ văn bản không có cấu trúc. Khả năng nhận diện chính xác các thực thể này có vai trò quan trọng trong các ứng dụng của xử lý ngôn ngữ tự nhiên như phân tích văn bản, trích xuất thông tin và hỏi đáp tự động. NER đã trải qua nhiều cải tiến, từ các phương pháp dựa trên luật đến các mô hình học sâu hiện đại.



Hình 13: Nhận diện các thực thể có tên từ văn bản

Trong giai đoạn đầu, NER chủ yếu dựa trên các phương pháp dựa trên luật và quy tắc thủ công. Các hệ thống này sử dụng các tập luật dựa trên ngữ cảnh của từ và các dấu hiệu như chữ viết hoa hay các tiền tố. Tuy nhiên, nhược điểm lớn của các phương pháp này là khó mở rộng và yêu cầu sự can thiệp thủ công lớn. Các phương pháp dựa trên luật, dù có hiệu quả trong một số miền dữ liệu nhất định, lại thiếu tính tổng quát khi áp dụng trên các miền dữ liệu khác.

Gần đây, với sự phát triển của các mô hình mạng nơ-ron, LSTM (Long Short-Term Memory) hay BiLSTM (Bidirectional LSTM) đã được sử dụng rộng rãi trong việc xử lý các chuỗi văn bản dài và phụ thuộc ngữ cảnh. Khi kết hợp với các lớp CRF, mô hình BiLSTM-CRF đã chứng tỏ hiệu quả cao trong việc nhận diện thực thể nhờ khả năng nắm bắt mối quan hệ giữa các từ trong câu [36].

Ngoài ra, các mô hình dựa trên transformer như BERT [12] cũng đã được áp dụng thành công vào NER. BERT không chỉ mang lại sự hiểu biết ngữ cảnh mạnh mẽ cho các từ trong chuỗi văn bản mà còn dễ dàng được tinh chỉnh (fine-tune) để áp dụng cho nhiều nhiệm vụ khác nhau, bao gồm cả NER. Một bước tiến quan trọng nữa là việc sử dụng các mô hình đa ngôn ngữ như mBERT [37] hay XLM-R [7], cho phép áp dụng NER trên nhiều ngôn ngữ khác nhau mà không cần quá nhiều dữ liệu huấn luyện.

Mặc dù các phương pháp học sâu hiện đại đã mang lại nhiều thành tựu, NER vẫn đối mặt với nhiều thách thức, đặc biệt là trong các ngôn ngữ ít tài nguyên và các miền dữ liệu chuyên biệt. Trong các ngôn ngữ như tiếng Việt, việc xây dựng các bộ dữ liệu đủ lớn để huấn luyện các mô hình học sâu vẫn là một thách thức lớn. Ngoài ra, vấn đề phân giải thực thể đa nghĩa và nhận diện các thực thể mới cũng đòi hỏi nhiều nghiên cứu hơn để cải thiện độ chính xác của các hệ thống NER. Vấn đề các thực thể lồng nhau cùng là một khía cạnh được quan tâm nhiều trong các nghiên cứu về NER.

Độ đo thường dùng nhất cho bài toán NER là Precision, Recall, F1. Trong đó, với một kiểu thực thể K:

- TP - số True Positive, là số thực thể được trích và gán nhãn chính xác vào kiểu thực thể K
- FP - số False Positive, là số thực thể được trích nhưng bị gán sai nhãn thực thể K
- FN - số False Negative, là số thực thể được trích có kiểu thực thể K nhưng bị gán sai sang nhãn thực thể khác

Các phương pháp tính micro average, macro average, weighted average giúp tính giá trình precision, recall, f1 trung bình cho toàn bộ bài toán.

Theo đó, một số hướng nghiên cứu có thể kể đến là sử dụng các phương pháp học không giám sát hoặc bán giám sát, trong đó các mô hình có thể học từ dữ liệu không gán nhãn hoặc dữ liệu ít gán nhãn. Các phương pháp này có tiềm năng giải quyết những hạn chế về dữ liệu trong các ngôn ngữ ít tài nguyên.

2.3 Gom cụm

2.3.1 Giới thiệu

Gom cụm là một kỹ thuật quan trọng trong lĩnh vực học máy không giám sát, nhằm nhóm các đối tượng tương tự lại với nhau dựa trên một số tiêu chí hoặc đặc trưng nhất định. Mục tiêu của gom cụm là tìm ra các nhóm (hoặc cụm) mà trong đó các đối tượng trong cùng một cụm có tính tương đồng cao hơn so với các đối tượng nằm ngoài cụm đó. Đặc biệt, kỹ thuật này hữu ích trong việc phân tích các tập dữ liệu lớn mà không có nhãn, giúp khám phá các mẫu ẩn trong tập dữ liệu.

Đối với các ứng dụng thực tế, gom cụm không chỉ đơn thuần là nhóm dữ liệu mà còn là một công cụ quan trọng để khám phá tri thức, đặc biệt trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, phân tích hình ảnh và dữ liệu phi cấu trúc. Các thuật toán gom cụm hiện nay đã được tích hợp chặt chẽ với các phương pháp biểu diễn dữ liệu như embedding, giúp tăng cường khả năng tìm ra các nhóm có ý nghĩa về ngữ nghĩa thay vì chỉ dựa vào các đặc trưng bề mặt.

Trong quá trình nghiên cứu, nhóm nhận thấy có nhiều phương pháp gom cụm khác nhau, mỗi phương pháp lại có những điểm mạnh và điểm yếu tùy thuộc vào loại dữ liệu và bài toán cụ thể. K-Means [32] là một thuật toán gom cụm nổi tiếng nhờ tính đơn giản và khả năng mở rộng cho các tập dữ liệu lớn. Tuy nhiên, thuật toán này yêu cầu người dùng phải xác định trước số cụm (k), điều này có thể không khả thi trong một số trường hợp khi chưa biết rõ về cấu trúc dữ liệu. Bên cạnh K-Means, DBSCAN [11] và HDBSCAN [34] là những phương pháp mà nhóm đặc biệt chú ý. DBSCAN không yêu cầu số lượng cụm cố định và có khả năng phát hiện các cụm với hình dạng khác nhau, cũng như xử lý tốt các điểm nhiễu. HDBSCAN là một phiên bản mở rộng của DBSCAN, sử dụng cấu trúc phân cấp để xác định cụm một cách linh hoạt và tự động, không cần người dùng chỉ định giá trị ngưỡng cố định cho độ dày đặc của cụm (epsilon). Qua nghiên cứu, nhóm đánh giá HDBSCAN là một lựa chọn phù hợp cho các bài toán dữ liệu không đồng đều, và nó đã chứng minh được hiệu quả trong việc phân tích các tập dữ liệu có tính phức tạp cao.

2.3.2 HDBSCAN

HDBSCAN [34] là một thuật toán phân cụm mạnh mẽ dựa trên mật độ, mở rộng từ DBSCAN bằng cách thêm yếu tố phân cấp. HDBSCAN có khả năng tự động phát hiện số lượng cụm phù hợp, xử lý dữ liệu có hình dạng phức tạp và nhận biết các điểm nhiễu.

Nguyên lý hoạt động của thuật toán này như sau:

- Thuật toán bắt đầu bằng việc tính core distance (khoảng cách cốt lõi) cho từng điểm dữ liệu. Core distance của một điểm a là khoảng cách của a đến điểm lân cận thứ min samples gần nhất, trong đó min samples là tham số của HDBSCAN quy định số lượng lân cận tối thiểu để xác định mật độ của điểm đó.

- Sau khi tính toán core distance, HDBSCAN sử dụng mutual reachability distance (khoảng cách tiếp cận lận nhau) để đo khoảng cách giữa hai điểm. Mutual reachability distance không chỉ phụ thuộc vào khoảng cách trực tiếp giữa hai điểm mà còn xét đến mật độ lân cận của mỗi điểm, thông qua core distance của cả hai. Công thức tính là:

$$\text{mutual\_reachability\_distance}(a,b) = \max\left\{\begin{array}{l}\text{core\_distance}(a),\\ \text{core\_distance}(b),\\ \text{distance}(a,b)\end{array}\right\}$$

- Tiếp đến thuật toán sử dụng reachability distance để xây dựng một đồ thị liên kết các điểm dữ liệu. Mỗi điểm dữ liệu là một nút, và các cạnh nối các nút là mutual reachability distance giữa các điểm.

- Từ đồ thị này, thuật toán thực hiện quá trình phân cấp, sắp xếp các điểm thành các cụm dựa trên mật độ. Điều này được thực hiện bằng cách loại bỏ các cạnh có khoảng cách lớn nhất và tìm kiếm các cụm con mật độ cao. Kết quả là một cây phân cấp (hierarchical tree) thể hiện mối quan hệ mật độ giữa các điểm.

- Thuật toán sử dụng một cơ chế dựa trên sự ổn định của cụm để chọn ra các cụm tốt nhất từ cây phân cấp. Cụm càng ổn định khi nó duy trì được sự tồn tại qua nhiều cấp độ mật độ khác nhau. Những cụm ổn định này sẽ được giữ lại, còn các điểm không thuộc cụm ổn định sẽ bị coi là nhiễu.

2.3.3 Silhouette Score

Silhouette Score [40] là một chỉ số đánh giá chất lượng của các cụm trong phân cụm. Nó đo lường mức độ tương đồng của mỗi điểm dữ liệu với cụm mà nó thuộc về so với cụm gần nhất khác. Silhouette Score cung cấp cái nhìn trực quan về sự phân tách giữa các cụm và mức độ gắn kết của các điểm trong cùng một cụm.

Silhouette Score cho mỗi điểm dữ liệu i được tính dựa trên hai giá trị chính:

- a(i): khoảng cách trung bình từ điểm i đến tất cả các điểm khác trong cùng một cụm. Điều này đo lường sự tương đồng giữa điểm i và các điểm trong cụm của nó.

- b(i): Khoảng cách trung bình từ điểm i đến tất cả các điểm trong cụm gần nhất khác. Điều này đo lường sự khác biệt giữa điểm i và các điểm trong cụm khác.

Silhouette Score cho điểm i được tính theo công thức:

$$s(i) = \frac{\max(a(i), b(i))}{b(i) - a(i)}$$

- Nếu *s(i)* càng gần 1 thi điểm i nằm gần các điểm trong cùng cụm hơn so với các điểm trong cụm khác.
- Nếu *s(i)* càng gần 0 thì i nằm ở biên giới giữa hai cụm, có thể là điểm không rõ ràng giữa hai cụm.
- Nếu *s(i)* càng gần -1 thì điểm i nằm gần các điểm trong một cụm khác hơn là trong cụm của nó.

Silhouette Score tổng thể cho một phân cụm được tính bằng trung bình của tất cả các Silhouette Score cho các điểm dữ liệu:

$$\text{Silhouette Score} = \frac{1}{N} \sum_{i=1}^{N} s(i)$$

2.4 Phân loại ý định và Điền trường thông tin

Phân loại ý định, từ gốc là Intent Classification, là một nhiệm vụ trong lĩnh vực Hiểu ngôn ngữ tự nhiên (Natural Language Understanding) có mục tiêu là phát hiện mục đích, mong muốn của chủ thể câu nói hay câu viết mà người đó tạo ra [49]. Nhiệm vụ này còn có tên gọi khác là Intent Detection (tiếng Việt là Phát hiện ý định). Trong một số bài báo cũ, từ goal (mục đích) được dùng thay thế cho intent (ý định).

Điền trường thông tin, tên gốc là Slot Filling là nhiệm vụ điền thông tin vào các trường (slots). Các slot là một loại từ hay span (một chuỗi kí tự con trong văn bản nhất định) có chứa thông tin ngữ nghĩa giúp bổ sung ý nghĩa cho ý định, làm cho ý định hoàn thiện [49]. Vì các thông tin này thường là thực thể nên nhiệm vụ còn được gọi là Điền thông tin thực thể.

Một ví dụ cho nhiệm vụ Phân loại ý định và Điền trường thông tin được cho ở bảng 1 dưới đây. Với truy vấn "find recent comedies by james cameron" mà người nói, người viết đưa ra, ta có thể biết được mục đích của người đó là tìm kiếm

một bộ phim với thể loại hài kịch và mới được ra mắt được đạo diễn bởi James Cameron. Ý định trong trường hợp này được xác định là "find_movie" với các trường là kiểu thực thể (hay loại thông tin) và giá trị tương ứng của chúng chính là các khoảng văn bản có trong câu truy vấn *date: recent (thời gian), comedies: genre (thể loại phim), dir: james cameron (đạo diễn).*

query	find	recent	comedies	by	james	cameron
slots	O	B-date	B-genre	O	B-dir	I-dir
intent	find_movie					

Bảng 1: Một ví dụ về bài toán Phân loại ý định và Điền trường thông tin [49]

Bài toán Phân loại ý định và Điền trường thông tin có thể mở rộng sang nhiệm vụ xác định miền hay lĩnh vực mà phát ngôn của một người thuộc vào. Trong ví dụ trên, miền của truy vấn có thể là "movie"(điện ảnh, phim ảnh). Tuy nhiên trong đa số các trường hợp, miền của bài toán thường được xác định từ trước. Danh sách các ý định và các trường cũng được định nghĩa từ miền này.

Phân loại ý định và Điền trường thông tin là một nhiệm vụ quan trọng trong lĩnh vực Hiểu ngôn ngữ tự nhiên (Natural Language Understanding), một nhánh nhỏ của Xử lý ngôn ngữ tự nhiên (Natural Language Processing) tập trung vào việc giúp máy tính hiểu và giải thích ý nghĩa của văn bản hoặc lời nói của con người. Dạng nhiệm vụ này đặc biệt tìm thấy nhiều trong Hiểu ngôn ngữ nói (Spoken Language Understading), tập trung làm việc với dữ liệu là giọng nói, tuy nhiên nó vẫn có thể làm việc với dữ liệu văn bản. Các thông tin ý định và trường được trích xuất được sử dụng có thể được dùng trong những pha xử lý tiếp theo như thu thập thông tin, hệ thống hỏi đáp, quản lý hội thoại, vân vân. Phân loại ý định và Điền trường thông tin được ứng dụng vào các hệ thống trợ lý ảo, chatbot, điều khiển robot, Internet of things, và còn nữa.

Phân loại ý định, đúng như tên gọi, thường được xem là bài toán phân loại văn bản (text classification) với dữ liệu đầu vào là câu truy vấn và kết quả là nhãn ý định trong danh sách ý định được định nghĩa trước. Đây là bài toán phân loại nhiều nhãn. Trong khi đó, Điền trường thông tin thường được giải quyết như bài toán gán nhãn chuỗi (sequence labeling), mục đích là xác định xem từ hoặc tiểu từ (subword) có phải là thông tin cho trường hoặc một phần thông tin trường hay không và gán nhãn kiểu cho nó. Cụ thể trong ví dụ ở bảng 1, câu truy vấn được phân loại là "find_movie" trong một danh sách ý định cho trước trong lĩnh vực điện ảnh. Từng từ trong câu truy vấn được phân loại thành các nhãn khác nhau. Các nhãn bắt đầu bằng "B-", "I-" chỉ ra rằng từ đó chính là thông tin cần điền cho trường, trường cần điền chính là những cụm "date", "genre", "dir" nằm trong

nhãn, đại diện cho loại thông tin liên quan đến ý định.

Hai nhiệm vụ Phân loại ý định và Điền trường thông tin trước đây thường được xử lý tách biệt hoặc chỉ ghép lại thành một "đường ống xử lý" (pipeline). Các phương pháp dùng cho từng nhiệm vụ có cùng chiều hướng phát triển giống như các nhiệm vụ phân loại văn bản và gán nhãn chuỗi khác, tùy từng thời điểm mà sẽ có phương pháp thịnh hành khác nhau. Những nhóm phương pháp chính có thể kể đến như là phương pháp dựa vào luật (rule-based), phương pháp dựa trên thống kê (statistical) hay được biết đến như những phương pháp học máy cổ điển (machine learning), phương pháp học sâu (deep learning) và gần đây nhất là các mô hình Transformer được huấn luyện trước. Một số nghiên cứu giả thấy được nhãn trường (slot label) và các lớp ý định (intent class) nên và có ảnh hưởng lẫn nhau theo cách mà giải quyết hai vấn đề cùng lúc sẽ cho kết quả tốt hơn cho cả hai nhiệm vụ. Kết quả là càng có nhiều nghiên cứu và những mô hình "joint" để giải quyết hai tác vụ đồng thời (xem hai tác vụ này như một và huấn luyện một mô hình để cho ra kết quả) [49]. Tuy nhiên, dù có là pipeline hay joint thì hai nhiệm vụ này vẫn thường được giải quyết theo phương pháp học có giám sát (supervised learning) nên cần một lượng dữ liệu được gán nhãn lớn và chất lượng để đạt được hiệu quả cao.

Tập dữ liệu sẵn có cho bài toán thường được giới hạn trong một miền nhất định. Điều này cũng thường xảy ra trong thực tế, khi ta chỉ quan tâm giải quyết bài toán trong một lĩnh vực cụ thể nào đó. Tuy nhiên vẫn có một số ít tập dữ liệu như SNIPS cố gắng tổng quát hóa bài toán và chứa dữ liệu từ nhiều miền khác nhau. ATIS [17] và SNIPS [9] là hai tập dữ liệu được sử dụng nhiều nhất trong hai nhiệm vụ Phân loại ý định và Điền trường thông tin.

Độ đo phổ biến nhất dùng để đánh giá tính hiệu quả của các mô hình cho bài toán Phân loại ý định là "Intent accuracy" (độ chính xác ý định). Nó được tính một cách đơn giản là tỉ lệ của số dự đoán đúng trên tổng số câu dữ liệu được cho dự đoán. Một số bài báo khác sử dụng độ đo precision, recall, f1 và một số loại kiểm định khác cho nhiệm vụ này. Đối với nhiệm vụ Điền trường thông tin, các độ đo dựa trên ma trận nhầm lẫn vẫn thường được sử dụng. Giống như các bài toán gán nhãn chuỗi khác, các độ đo precision, recall, f1 được tính cho từng nhãn, sau đó tính điểm trung bình bằng phương pháp micro-averaged hoặc macro-averaged. Cho một lớp K, các đại lượng True Positive, False Positive, False Negative được xác định như sau:

- TP - số True Positive, là số đoạn văn bản (span) của lớp K được dự đoán chính xác

- FP - số False Positive, là số đoạn văn bản của lớp khác nhưng bị đoán sai thành lớp K
- FN - số False Negative, là số đoạn văn bản của lớp K bị đoán sai thành lớp khác

Ngoài ra slot accuracy cũng được sử dụng trong một vài bài báo [53].

Một số trường hợp xét đến tính chính xác ngữ nghĩa (semantic accuracy). Một câu được tính là được phân tích đúng nếu cả ý định được phân loại của câu đó và các trường trong câu đều có nhãn đúng [49].

Một số vấn đề khác khi làm việc với Phân loại ý định và Điền trường thông tin có thể gặp phải đó là trường hợp đa lĩnh vực, đa ngôn ngữ, đa ý định trong câu, vấn đề nhập nhằng, thiếu hụt dữ liệu cho các miền hay ngôn ngữ tài nguyên thấp (low-resource), độ nhạy với dữ liệu chưa thấy, câu truy vấn quá dài hoặc quá ngắn.

2.4.1 Phân loại văn bản

Phân loại văn bản (text classification) là một bài toán cơ bản trong Xử lý ngôn ngữ tự nhiên, nơi nhiệm vụ chính là gán một nhãn hoặc phân loại văn bản vào một nhóm hoặc lớp cụ thể. Đây là quy trình tự động giúp sắp xếp và tổ chức văn bản theo các danh mục đã định trước, như phân loại tin tức, phân tích cảm xúc, phát hiện thư rác hoặc các ứng dụng khác liên quan đến phân loại văn bản. Các ứng dụng của phân loại văn bản vô cùng rộng rãi, bao gồm từ tìm kiếm thông tin trong hệ thống cơ sở dữ liệu đến phân tích nội dung trên mạng xã hội.

Các dạng bài toán phân loại văn bản bao gồm phân loại nhị phân, trường hợp văn bản chỉ được gán vào một trong hai nhãn (ví dụ như "spam" và "không spam" hoặc "tích cực" và "tiêu cực"); phân loại đa lớp, khi văn bản thuộc về một trong nhiều nhãn (ví dụ như các danh mục tin tức như "thể thao", "giải trí", "kinh tế"); và phân loại đa nhãn, khi một văn bản có thể được gán nhiều nhãn đồng thời (ví dụ một bài báo có thể thuộc cả danh mục "chính trị" và "kinh tế").

Về mặt phương pháp, phân loại văn bản ban đầu được thực hiện thông qua các phương pháp thống kê truyền thống như Bag-of-Words (BoW) và TF-IDF, sử dụng các đặc trưng đơn giản từ văn bản để huấn luyện mô hình. Các thuật toán phân loại cổ điển như Naive Bayes, Support Vector Machine (SVM), và Logistic Regression đã được áp dụng rộng rãi trong thời gian đầu. Tuy nhiên, với sự phát triển của học sâu, các mô hình học sâu như Convolutional Neural Networks (CNN), họ mô hình Recurrent Neural Networks (RNN), và đặc biệt các mô hình

tiền huấn luyện Transformer gần đây như BERT đã cải thiện đáng kể độ chính xác. Những mô hình này có khả năng nắm bắt ngữ nghĩa và mối quan hệ ngữ cảnh giữa các từ trong câu, giúp phân loại văn bản trở nên hiệu quả hơn.

Dù đã đạt được những tiến bộ đáng kể, phân loại văn bản vẫn gặp phải một số thách thức như vấn đề dữ liệu không cân bằng, tài nguyên thấp, sự phức tạp khi xử lý ngôn ngữ và còn nhiều vấn đề khác.

2.4.2 Gán nhãn chuỗi

Gán nhãn chuỗi (sequence labeling) là một bài toán quan trọng trong lĩnh vực Xử lý ngôn ngữ tự nhiên, mà nhiệm vụ của nó là gán nhãn cho mỗi phần tử trong một chuỗi dữ liệu. Thực chất, nó cũng là một dạng bài toán phân loại nhưng dùng để phân loại cho phần tử trong chuỗi. Thông thường, chuỗi dữ liệu là các từ trong một câu và các nhãn có thể là từ loại, kiểu thực thể có tên, hoặc các thành phần cú pháp. Vì thế bài toán này cũng thường được gọi là "Phân loại từ" (token classification). Mục tiêu của gán nhãn chuỗi là hiểu được vai trò của từng từ trong văn bản hoặc câu, giúp máy tính có thể phân tích và xử lý ngôn ngữ một cách có cấu trúc.

Các bài toán phổ biến trong gán nhãn chuỗi bao gồm nhận diện thực thể có tên (NER), phân tích từ loại (POS tagging), và phân chia cụm từ (chunking). Ví dụ, trong bài toán NER, hệ thống có nhiệm vụ gán nhãn các từ thuộc về các kiểu thực thể như tên người (PER), tên địa điểm (LOC), tổ chức (ORG), vân vân. Tương tự, trong POS tagging, hệ thống gán các nhãn từ loại cho từng từ như danh từ (N), động từ (V), tính từ (ADJ) và một số từ loại khác. Gán nhãn chuỗi có vai trò quan trọng trong nhiều ứng dụng NLP như dịch máy, tóm tắt văn bản, và hệ thống hỏi đáp.

Phương pháp để giải quyết bài toán gán nhãn chuỗi đã trải qua nhiều giai đoạn phát triển. Trước đây, các mô hình thống kê truyền thống như Hidden Markov Models (HMM) và Conditional Random Fields (CRF) thường được sử dụng. Những mô hình này dựa trên xác suất và dự đoán nhãn dựa trên mối quan hệ giữa các từ trong câu. Với sự phát triển của học sâu, các mô hình như Recurrent Neural Networks (RNN) và đặc biệt là Long Short-Term Memory (LSTM) đã giúp cải thiện đáng kể khả năng gán nhãn chuỗi nhờ khả năng nắm bắt thông tin ngữ cảnh xa. Gần đây, các mô hình Transformers với cơ chế attention như BERT, GPT, T5 đã được ứng dụng rộng rãi trong gán nhãn chuỗi, nhờ khả năng hiểu ngữ cảnh sâu rộng và hiệu quả vượt trội trên nhiều tập dữ liệu.

Gán nhãn chuỗi vẫn còn nhiều khó khăn cần giải quyết. Những vấn đề này

là những vấn đề chung trong nhiều nhiệm vụ Xử lý ngôn ngữ tự nhiên khác như vấn đề cân bằng dữ liệu, thiếu hụt dữ liệu, dự đoán cho nhãn dữ liệu chưa thấy (zero-shot), và còn nhiều vấn đề khác.

2.5 Đọc hiểu máy

Đọc hiểu máy (Machine Reading Comprehension hay MRC) là một nhiệm vụ đầy thách thức và là chủ đề nóng trong trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Nhiệm vụ này tập trung vào việc phát triển các hệ thống máy tính có khả năng đọc hiểu và trả lời các câu hỏi dựa trên các văn bản.

Nhiệm vụ Đọc hiểu máy, như tên gọi, dùng để đánh giá xem một chiếc máy (machine) có khả năng hiểu được ngôn ngữ tự nhiên đến đâu. Để đo lường độ hiểu ngôn ngữ tự nhiên của một máy tính, một văn bản (bài văn, trích đoạn, câu văn) và một tập các câu hỏi liên quan đến nó sẽ được đưa cho các máy và so sánh câu trả lời của nó qua một tập câu trả lời cho trước. Trong tập dữ liệu đọc hiểu máy, mỗi điểm dữ liệu chứa một văn bản cho trước gọi là ngữ cảnh (context) c, một câu hỏi liên quan đến ngữ cảnh q và một câu trả lời cho nó a. Các hệ thống MRC sẽ học một hàm số f, sao cho nó nó có thể trích xuất hoặc sinh ra câu trả lời giống với a khi nó nhận được ngữ cảnh c và câu hỏi q:

$$a = f(c, q)$$

Một mẫu dữ liệu được lấy từ tập SQuAD cho nhiệm vụ đọc hiểu máy [**?**] được trình bày trong Bảng 2.

Context	Alyssa got to the beach after a long trip. She's from Charlotte. She traveled from Atlanta. She's now in Miami. She went to Miami to visit some friends. But she wanted some time to herself at the beach, so she went there first. After going swimming and laying out, she went to her friend Ellen's house. Ellen greeted Alyssa and they both had some lemonade to drink. Alyssa called...
Question	Why did Alyssa go to Miami?
Answer	To visit some friends

Bảng 2: Mẫu dữ liệu từ tập SQuAD

Dù có một vài nghiên cứu sử dụng thuật ngữ Đọc hiểu máy (MRC) và Hỏi đáp (QA) thay phiên nhau. Chúng thực chất có sự khác nhau. Bài toán đọc hiểu là pha sau của bài toán Information Retrieval của các hệ thống QA, hay nói cái

cách đọc hiểu máy là một phần của QA. Đầu vào của hệ thống QA chỉ có câu hỏi trong khi hệ thống MRC thì cần thêm ngữ cảnh (ngữ cảnh này được cung cấp bởi bộ trích tài liệu). Sau khi có được tài liệu liên quan (từ bộ truy vấn thông tin – retriever), bộ Reader (MRC) có nhiệm vụ tìm câu trả lời (nếu có) trong các tài liệu này.

Ta có thể phân loại hệ thống Đọc hiểu máy qua đầu vào (input) gồm câu hỏi (question) và ngữ cảnh (context); và đầu ra (output) của nó:

- *Câu hỏi:* chia làm các dạng câu hỏi như factoid (câu hỏi về một fact - thông tin thực tế nào đó), non-factoid (câu hỏi cần sự suy diễn, hay nêu ra nhận xét, cảm nghĩ), yes/no (câu hỏi đúng sai)
- *Ngữ cảnh:* chia làm hai dạng 'single passage' (chỉ gồm một đoạn trích) và 'multiple passages' (gồm nhiều đoạn trích kết hợp)
- *Đầu ra:* generative (câu trả lời có được bằng cách sinh từ ngữ) và extractive (câu trả lời được rút trích chính xác trong ngữ cảnh)

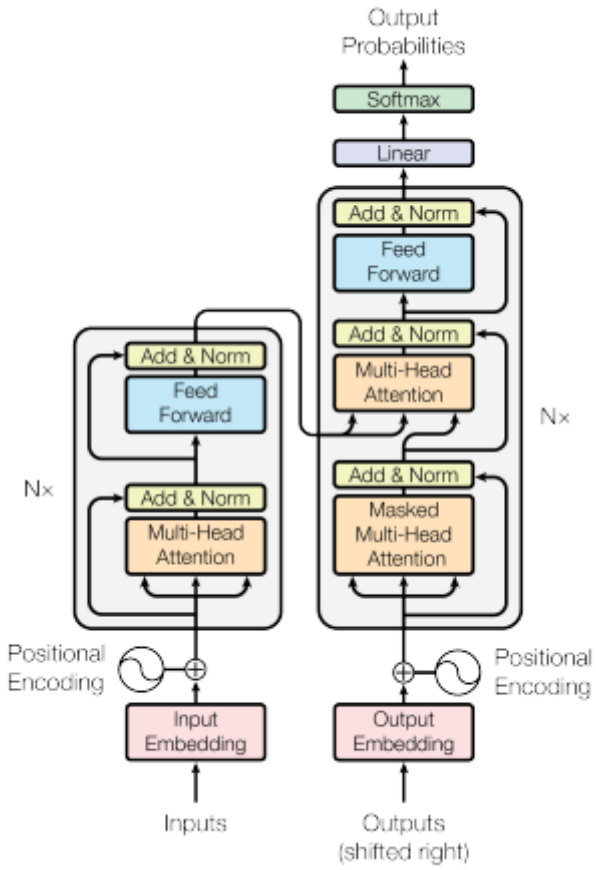
Một số cách tiếp cận để giải quyết bài toán MRC:

- *Rule-based:* những chuyên gia ngôn ngữ học xây dựng thủ công các luật. Phương pháp này có vấn đề là tính không bao quát đầy đủ của các luật. Ngoài ta, nó còn chuyên biệt về một lĩnh vực, khi một lĩnh vực mới được nghiên cứu, một tập luật mới phải được tạo ra.
- *Các phương pháp machine learning cổ điển:* Cần một tập những đặc trưng (features) được định nghĩa sẵn qua quá trình "feature engineering", sau đó huấn luyện mô hình để chuyển các đặc trưng input này cho ra output.
- *Các phương pháp deep learning:* học các đặc trưng một cách tự động.

Trong đó, các nghiên cứu về Đọc hiểu máy từ trước đến nay phần nhiều tập trung với việc trích xuất ra kết quả. Sự có mặt của BERT[12] đưa các mô hình Đọc hiểu máy trích xuất lên một tầm mới, khiến cho việc ứng dụng mô hình Đọc hiểu máy vào các bài toán trích xuất thông tin xuất hiện nhiều gần đây [26][22] [13]. Sự ra đời của GPT [52] cũng làm kéo theo quan tâm đến những mô hình Đọc hiểu máy sinh câu trả lời. Những mô hình này thường được gọi là những mô hình ngôn ngữ lớn (Large Language Model - LLM) khi được huấn luyện trên tập dữ liệu cực lớn, và câu trả lời được sinh ra một cách tự nhiên giống con người khi nói chuyện. Sự hình thành của các mô hình ngôn ngữ tiền huấn luyện (pre-trained language model) này làm cho đề tài Đọc hiểu máy trở nên bùng nổ.

2.6 Kiến trúc Transformers

Transformers [45] là một kiến trúc mạng nơ-ron dựa trên cơ chế chú ý (attention mechanism), được đội ngũ nghiên cứu của Google giới thiệu lần đầu vào năm 2017. Kiến trúc này đã mang lại những tiến bộ đáng kể trong lĩnh vực xử lý ngôn ngữ tự nhiên và các nhiệm vụ liên quan khác. Thành phần chính của kiến trúc transformers bao gồm một bộ mã hóa (encoder) và một bộ giải mã (decoder).



Hình 14: Kiến trúc Transformers [45] với hai bộ encoder và decoder

2.6.1 Bộ mã hoá encoder

Bộ mã hoá (encoder [45]) là một thành phần quan trọng của kiến trúc bộ mã hóa - bộ giải mã transformer. Chúng có nhiệm vụ phân tích và biểu diễn chuỗi đầu vào theo cách mà mô hình có thể hiểu được. Bộ mã hóa xử lý chuỗi đầu vào và tạo ra một biểu diễn liên tục, hay còn gọi là embedding của chuỗi đầu vào. Các embedding này sau đó được chuyển đến bộ giải mã để tạo ra chuỗi đầu ra.

Kiến trúc bộ mã hóa transformer thường bao gồm nhiều lớp, mỗi lớp bao gồm một cơ chế tự chú ý (self-attention) và một mạng nơ-ron truyền thẳng (feed-forward neural network). Cơ chế tự chú ý cho phép mô hình đánh giá tầm quan trọng của các phần khác nhau trong chuỗi đầu vào bằng cách tính toán tích vô hướng của các embedding. Cơ chế này còn được gọi là multi-head attention.

Mạng truyền thẳng cho phép mô hình trích xuất các đặc trưng cấp cao hơn từ đầu vào. Mạng này thường bao gồm hai lớp tuyến tính với một hàm kích hoạt ReLU [55] ở giữa. Mạng truyền thẳng giúp mô hình trích xuất ý nghĩa sâu hơn từ dữ liệu đầu vào và biểu diễn đầu vào một cách gọn gàng và hữu ích hơn.

2.6.2 Bộ giải mã decoder

Bộ giải mã (decoder [45]) chịu trách nhiệm tạo ra chuỗi đầu ra trong các tác vụ xử lý ngôn ngữ tự nhiên. Nó bao gồm nhiều lớp, mỗi lớp được tạo thành từ một lớp multi-head self-attention và một mạng nơ-ron truyền thẳng (feedforward neural network).

Bộ giải mã nhận đầu vào là các trạng thái ẩn do bộ mã hóa tạo ra và các token tạo ra trước đó để dự đoán token đầu ra tiếp theo. Tại mỗi bước, bộ giải mã chú ý đến các phần khác nhau của chuỗi đầu vào bằng cách sử dụng cơ chế chú ý của nó, cho phép nó nắm bắt các mối quan hệ phức tạp giữa các chuỗi đầu vào và đầu ra.

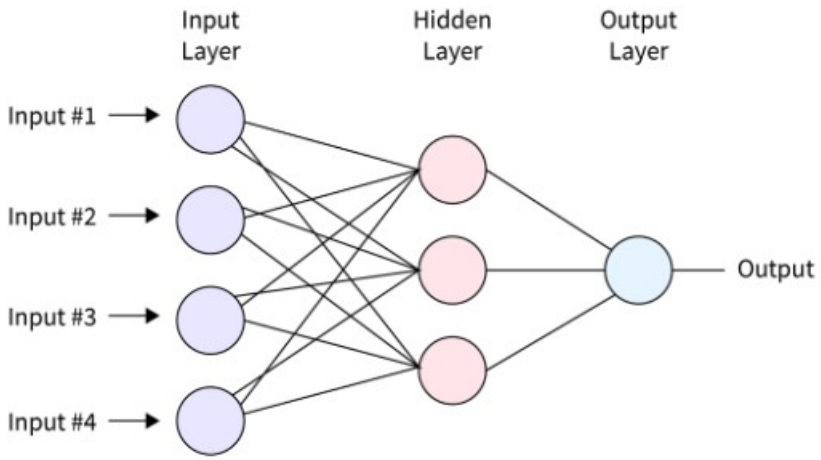
2.6.3 Mạng nơ ron truyền thẳng

Mạng nơ-ron truyền thẳng (Feedforward Neural Network) [41] là một cấu trúc cơ bản trong lĩnh vực mạng nơ-ron nhân tạo và học sâu. Mạng này được gọi là truyền thẳng vì thông tin chỉ chảy theo một hướng từ lớp đầu vào đến lớp đầu ra mà không có vòng lặp. Nó bao gồm ba loại lớp:

- Input Layer: là lớp đầu vào và chịu trách nhiệm nhận dữ liệu đầu vào và chuyển nó đến lớp tiếp theo. Lớp đầu vào có một số lượng nơ-ron tương ứng với số lượng đặc trưng trong dữ liệu đầu vào. Lớp này không thực hiện bất

kỳ tính toán hay biến đổi nào trên dữ liệu. Nó chỉ đóng vai trò là nơi lưu giữ dữ liệu đầu vào.

- Hidden Layer: là một lớp ẩn trong mạng nơ ron truyền thẳng, nằm giữa lớp đầu vào và lớp đầu ra. Nó được gọi là ẩn vì nó không tương tác trực tiếp với môi trường bên ngoài. Thay vào đó, nó chỉ nhận đầu vào từ lớp đầu vào hoặc các lớp ẩn trước đó, sau đó thực hiện các tính toán nội bộ trước khi chuyển đầu ra đến lớp kế tiếp. Chức năng chính của hidden layer là biến đổi dữ liệu đầu vào và trích xuất các đặc trưng hữu ích, cho phép mạng học các mối quan hệ phức tạp và trừu tượng hơn giữa đầu vào và đầu ra.
- Output Layer: là lớp cuối cùng trong kiến trúc mạng. Chức năng chính của nó là tạo ra đầu ra cuối cùng của mạng dựa trên dữ liệu đầu vào đã được xử lý. Lớp đầu ra nhận đầu vào từ lớp ẩn cuối cùng và tạo ra đầu ra của mạng bằng cách áp dụng một tập hợp các biến đổi lên dữ liệu này. Các biến đổi này có thể là một số hàm kích hoạt phổ biến cho lớp đầu ra như hàm sigmoid [33] cho phân loại nhị phân và hàm softmax [21] cho phân loại đa lớp.

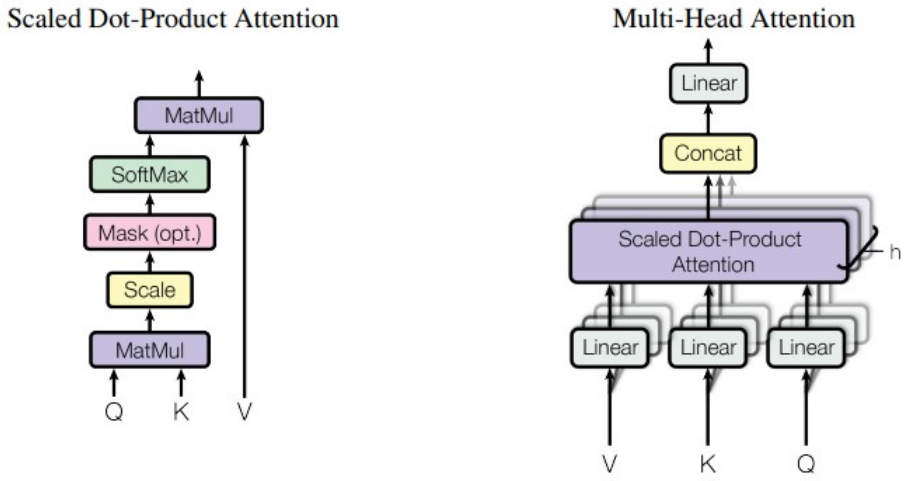


Hình 15: Mạng nơ ron truyền thẳng với ba lớp input, hidden và output layer

Mục đích của mạng nơ-ron truyền thẳng là xấp xỉ một số hàm nhất định. Đầu vào của mạng là một vector giá trị x, được truyền qua mạng với từng lớp một và biến đổi thành đầu ra y. Đầu ra cuối cùng của mạng dự đoán hàm mục tiêu cho đầu vào đã cho. Mạng thực hiện dự đoán này bằng cách sử dụng một tập hợp các tham số theta, được điều chỉnh trong quá trình huấn luyện để giảm thiểu sai số giữa dự đoán của mạng và hàm mục tiêu.

2.6.4 Cơ chế multi-head attention

Cơ chế tự chú ý (multi-head attention hay self attention [45]) cho phép mô hình đánh giá tầm quan trọng của các phần khác nhau trong chuỗi đầu vào bằng cách tính toán tích vô hướng của các embedding.



Hình 16: Cơ chế của multi head attention trong kiến trúc Transformer [45]

Cơ chế multi-head attention bao gồm ba vector có thể học được: vector truy vấn (query vector), vector khóa (key vector) và vector giá trị (value vector). Các vector này được sử dụng để tính toán trọng số attention cho mỗi phần tử đầu vào. Các vector truy vấn và vector khóa được nhân với nhau thông qua một tích vô hướng để tạo ra một ma trận điểm. Ma trận này đại diện cho sự quan trọng của mỗi từ trong chuỗi đầu vào liên quan đến tất cả các từ khác. Điểm số cao thể hiện một từ nên nhận được nhiều sự chú ý hơn, trong khi điểm số thấp thể hiện rằng một từ nên nhận ít sự chú ý hơn.

Ma trận điểm sau đó sẽ được thu nhỏ theo kích thước của các vector truy vấn và vector khóa và được chuyển đổi thành một ma trận xác suất bằng cách sử dụng hàm softmax [21]. Ma trận xác suất sau đó được nhân với vector giá trị, làm tăng sự quan trọng của các phần tử đầu vào với xác suất cao hơn và làm giảm sự quan trọng của các phần tử đầu vào với xác suất thấp hơn.

Đầu ra này của các vector truy vấn, vector khóa và vector giá trị sau đó được truyền qua một lớp tuyến tính để xử lý tiếp. Lớp tuyến tính kết hợp các vector đầu vào thông qua một biến đổi tuyến tính, cho phép mô hình học được mối quan

hệ phức tạp hơn giữa các phần tử đầu vào.

Quá trình self-attention được lặp lại cho mỗi từ trong chuỗi đầu vào. Vì các trọng số attention cho mỗi từ không phụ thuộc vào các từ khác trong chuỗi, nhiều bản sao của mô đun self-attention có thể được sử dụng để xử lý đầu vào đồng thời, làm cho cơ chế attention trở nên đa đầu (multi-head). Điều này cho phép mô hình chú ý đến các phần khác nhau của đầu vào và nắm bắt được các phụ thuộc phức tạp hơn giữa các phần tử đầu vào.

2.7 Các mô hình ngôn ngữ huấn luyện trước

Các mô hình huấn luyện trước là các mạng nơ-ron hoặc các mô hình máy học đã được huấn luyện trước trên một tập dữ liệu lớn, thường yêu cầu nhiều tài nguyên tính toán và thời gian. Những mô hình này sau đó có thể được tinh chỉnh hoặc sử dụng trực tiếp cho các nhiệm vụ cụ thể bằng cách tận dụng các đặc điểm và mẫu đã học từ tập dữ liệu gốc. Mô hình huấn luyện trước đặc biệt có giá trị trong các tình huống mà việc huấn luyện mô hình từ đầu là không khả thi do hạn chế về dữ liệu, tài nguyên tính toán hoặc thời gian.

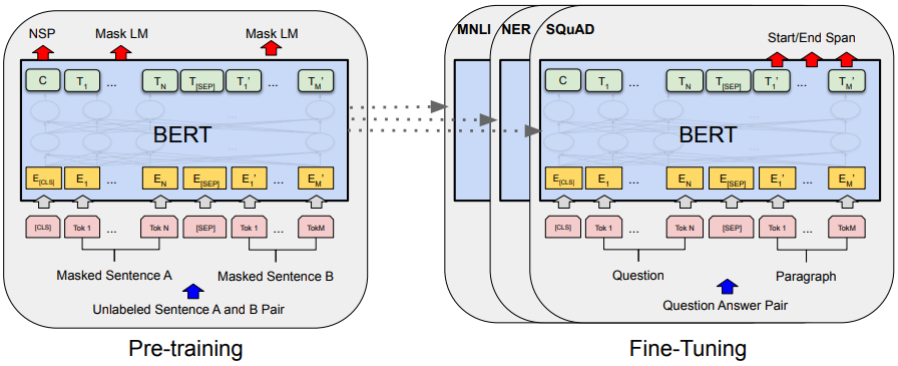
Một số mô hình huấn luyện trước khá nổi tiếng hiện nay như BERT [12], GPT [52], RoBERTa [30],... đóng vai trò khá quan trọng trong lĩnh vực xử lí ngôn ngữ tự nhiên nhờ vào khả năng nắm bắt và hiểu ngữ cảnh của văn bản một cách hiệu quả.

2.7.1 Mô hình BERT

BERT [12] (Bidirectional Encoder Representations from Transformers) là một mô hình học sâu dựa trên kiến trúc Transformers [45], được tạo ra bởi đội ngũ nghiên cứu Google AI Language và hoạt động như một công cụ đa năng cho hơn 11 tác vụ ngôn ngữ phổ biến.

Trước đây, các mô hình ngôn ngữ chỉ có thể đọc văn bản đầu vào một cách tuần tự từ trái sang phải (left-to-right) hoặc từ phải sang trái (right-to-left) nhưng không thể thực hiện cả hai cùng một lúc. BERT khác biệt vì nó được thiết kế để đọc theo cả hai hướng cùng một lúc. Sự ra đời của các mô hình Transformers đã cho phép thực hiện khả năng này, được gọi là tính hai chiều. Mô hình này sử dụng bộ mã hoá Transformers để học biểu diễn từ (text representations) từ một tập dữ liệu không được gán nhãn.

Kiến trúc của BERT gồm hai bước: huấn luyện trước (pre-training) và tinh chỉnh (fine-tuning).



Hình 17: Kiến trúc hai bước của mô hình BERT [12]

Ở bước pre-training, mô hình đã thực hiện hai nhiệm vụ sau:

- Masked Language Modeling (MLM): mô hình được huấn luyện với một phần của dữ liệu bị che giấu và một tỷ lệ nhất định các từ trong câu được thay thế bằng mặt nạ [mask]. Mô hình phải dự đoán các từ gốc của các vị trí này dựa trên ngữ cảnh của các từ xung quanh.
- Next Sentence Prediction (NSP): là nhiệm vụ mà BERT được huấn luyện để dự đoán xem liệu một câu có phải là câu tiếp theo của câu trước đó trong văn bản gốc hay không. Nhiệm vụ này giúp mô hình hiểu mối quan hệ giữa các câu.

Để tinh chỉnh, mô hình BERT trước tiên được khởi tạo với các tham số được huấn luyện trước và tất cả các tham số này được tinh chỉnh bằng cách sử dụng dữ liệu được gắn nhãn từ các tác vụ xuôi dòng (downstream tasks). Mỗi downstream task có các mô hình tinh chỉnh riêng biệt, mặc dù chúng được khởi tạo với cùng các tham số được huấn luyện trước. Trong quá trình này, các tham số của mô hình sẽ được điều chỉnh thông qua huấn luyện có giám sát trên dữ liệu đã gán nhãn cụ thể cho nhiệm vụ đó.

2.7.2 Mô hình ELECTRA

ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) [**?**] là một mô hình ngôn ngữ dựa trên kiến trúc Transformer, được phát triển bởi Clark và cộng sự vào năm 2020. Mô hình này được thiết kế nhằm cải thiện hiệu quả huấn luyện so với các phương pháp truyền thống như BERT, giúp xử lý các tác vụ xử lý ngôn ngữ tự nhiên với ít tài nguyên hơn nhưng

vẫn đạt hiệu suất cao.

Điểm đặc biệt của ELECTRA là cơ chế huấn luyện Replaced Token Detection (RTD). Thay vì che đi một số từ trong câu và yêu cầu mô hình dự đoán các từ bị che như BERT, ELECTRA thay thế một số từ bằng các từ giả và yêu cầu mô hình xác định từ nào đã bị thay thế. Điều này giúp ELECTRA học tốt hơn trên toàn bộ văn bản, thay vì chỉ dựa vào một phần như trong phương pháp Masked Language Modeling (MLM) của BERT.

ELECTRA bao gồm hai thành phần: Generator và Discriminator. Generator có nhiệm vụ tạo ra các từ giả thay thế, tương tự như mô hình BERT. Trong khi đó, Discriminator - thành phần chính của ELECTRA - được huấn luyện để phân biệt các từ thật và từ giả trong câu, tạo ra một mô hình hiệu quả hơn trong việc học ngữ cảnh.

Mặc dù số lượng tham số của ELECTRA nhỏ hơn so với BERT, nhưng mô hình này vẫn đạt được hoặc vượt qua hiệu suất của BERT trên nhiều tác vụ Xử lý ngôn ngữ tự nhiên, như phân loại văn bản, nhận diện thực thể (NER), và trả lời câu hỏi. Đặc biệt, ELECTRA cần ít bước huấn luyện hơn để đạt độ chính xác cao, giúp tiết kiệm thời gian và tài nguyên.

Với những ưu điểm trên, ELECTRA không chỉ hiệu quả hơn về mặt tính toán mà còn cho thấy sự vượt trội trên nhiều bộ dữ liệu chuẩn như GLUE và SQuAD, ngay cả khi sử dụng phiên bản nhỏ hơn như ELECTRA Small. Điều này làm cho ELECTRA trở thành một lựa chọn mạnh mẽ cho các bài toán xử lý ngôn ngữ tự nhiên hiện nay.

2.7.3 Mô hình đa ngôn ngữ (Multilingual Language Models)

Các mô hình ngôn ngữ đa ngôn ngữ (Multilingual Language Models [8]) là các mô hình được huấn luyện trên văn bản từ nhiều ngôn ngữ khác nhau. Chúng học cách mã hóa và tạo ra văn bản trong các ngôn ngữ khác nhau, cho phép chúng hiểu và tạo ra văn bản bằng bất kỳ ngôn ngữ nào mà chúng được huấn luyện. Những mô hình này thường được huấn luyện trên một bộ sưu tập lớn các văn bản đa ngôn ngữ và sau đó được điều chỉnh tinh chỉnh trên các nhiệm vụ cụ thể xuôi dòng (downstream tasks).

Một trong những ưu điểm chính của các mô hình ngôn ngữ đa ngôn ngữ là khả năng chuyển giao kiến thức qua các ngôn ngữ. Bằng cách học các biểu diễn chung, những mô hình này có thể tận dụng kiến thức được học trong một ngôn ngữ để hoạt động tốt trong một ngôn ngữ khác, ngay cả khi dữ liệu huấn luyện cho ngôn

ngữ đó là hạn chế. Điều này làm cho các mô hình ngôn ngữ đa ngôn ngữ trở nên đặc biệt hữu ích trong các tình huống mà một tập hợp đa dạng các ngôn ngữ cần được xử lý hoặc phân tích.

Mô hình xuyên ngôn ngữ (Cross-lingual models) [8] là một loại cụ thể của mô hình ngôn ngữ đa ngôn ngữ, tập trung vào việc hiểu và dịch ngôn ngữ giữa nhiều ngôn ngữ khác nhau. Các mô hình này nhằm mục đích vượt qua rào cản ngôn ngữ bằng cách học mã hóa văn bản trong một ngôn ngữ và giải mã nó sang ngôn ngữ khác. Chúng có thể được sử dụng cho các nhiệm vụ như dịch máy, phân loại tài liệu xuyên ngôn ngữ và truy xuất thông tin xuyên ngôn ngữ.

Các mô hình xuyên ngôn ngữ như XLM [8] và XLM-RoBERTa [7] đạt được sự hiểu biết xuyên ngôn ngữ bằng cách học các biểu diễn không phụ thuộc vào ngôn ngữ. Các biểu diễn này nắm bắt thông tin ngữ nghĩa và cú pháp được chia sẻ giữa các ngôn ngữ, cho phép các mô hình ánh xạ văn bản từ ngôn ngữ này sang ngôn ngữ khác. Điều này đặc biệt hữu ích trong các kịch bản mà dữ liệu được gán nhãn cho mỗi ngôn ngữ là hạn chế, vì các mô hình có thể tận dụng kiến thức đã học từ các ngôn ngữ khác để cải thiện hiệu suất.

2.7.4 Mô hình XLM-RoBERTa

XLM-RoBERTa [7] là phiên bản cải tiến của XLM, được xây dựng dựa trên kiến trúc RoBERTa [30]. RoBERTa là một biến thể của BERT đã được tiền huấn luyện trên một tập dữ liệu lớn hơn và trong nhiều bước huấn luyện hơn, dẫn đến hiệu suất cao hơn trong các nhiệm vụ xử lý ngôn ngữ tự nhiên. Điều này giúp cho XLM-RoBERTa kế thừa khả năng hiểu đa ngôn ngữ của XLM trong khi hưởng lợi từ việc học biểu diễn cải tiến của RoBERTa.

Ý tưởng chính đằng sau XLM-RoBERTa là tận dụng phương pháp tiền huấn luyện của RoBERTa, bao gồm tiền huấn luyện trên quy mô lớn và tinh chỉnh siêu tham số mở rộng, để tạo ra một mô hình xuyên ngôn ngữ mạnh mẽ và hiệu quả hơn.

Kiến trúc của XLM-RoBERTa tương tự như các mô hình dựa trên transformers khác như đã trình bày trước đó. Nó bao gồm các lớp nhúng, các bộ mã hóa transformer và cấu trúc downstream.

Một số cải tiến của XLM-RoBERTa so với XLM có thể kể đến như:

- Tiền huấn luyện trên tập dữ liệu lớn hơn: XLM-RoBERTa sử dụng một tập dữ liệu tiền huấn luyện lớn hơn, tương tự như RoBERTa. Điều này giúp mô hình học các biểu diễn mạnh mẽ và tổng quát hơn từ dữ liệu ngôn ngữ đa dạng và phong phú.

- Số bước huấn luyện nhiều hơn: XLM-RoBERTa trải qua nhiều bước tiền huấn luyện hơn so với XLM. Số bước tăng lên cho phép mô hình hội tụ tốt hơn và nắm bắt được các sắc thái ngôn ngữ tinh tế hơn.
- Loại bỏ nhiệm vụ dự đoán câu kế tiếp (Next Sentence Prediction): mô hình loại bỏ nhiệm vụ dự đoán câu tiếp theo được sử dụng trong mô hình BERT gốc. Thay vào đó, chúng chỉ dựa vào mục tiêu mô hình hóa ngôn ngữ bị che (Masked Language Modeling), thứ mang lại nhiều hiệu quả hơn.

2.7.5 Mô hình Sentence-BERT

SentenceBERT (SBERT) [38] là một cải tiến của mô hình BERT [12], được thiết kế để tạo ra các biểu diễn vector hiệu quả cho các câu và đoạn văn. Mặc dù BERT ban đầu được xây dựng để xử lý các tác vụ như phân loại văn bản hoặc trích xuất thông tin, nó không tối ưu cho việc so sánh các câu do cần phải tính toán mỗi lần một cặp câu mới. SBERT khắc phục vấn đề này bằng cách thêm một tầng pooling lên trên BERT, cho phép biểu diễn mỗi câu dưới dạng một vector cố định.

SBERT sử dụng kiến trúc Siamese [38], trong đó hai câu đầu vào được mã hóa bởi cùng một mạng BERT, sau đó vector biểu diễn của hai câu này được so sánh thông qua một số hàm đo lường khoảng cách như cosine similarity. Quá trình này giúp SBERT có thể so sánh câu một cách nhanh chóng và hiệu quả, lý tưởng cho các tác vụ như tìm kiếm thông tin, so sánh ngữ nghĩa câu, hay tìm kiếm cặp câu tương đồng.

Điểm mạnh của SBERT là khả năng giảm thiểu thời gian tính toán khi làm việc với nhiều câu, đồng thời vẫn giữ được hiệu suất tốt trong việc nắm bắt ngữ nghĩa. Đây là lý do mà SBERT được ưa chuộng trong các ứng dụng xử lý ngôn ngữ tự nhiên, đặc biệt là trong các bài toán liên quan đến nhúng câu và so sánh ngữ nghĩa.

2.7.6 Mô hình SimCSE

SimCSE (Simple Contrastive Learning of Sentence Embeddings) [16] là một phương pháp đơn giản nhưng hiệu quả trong việc tạo ra các biểu diễn nhúng câu. Mô hình này được xây dựng dựa trên BERT [12] và khai thác sức mạnh của học tương phản (contrastive learning) để cải thiện chất lượng nhúng câu.

SimCSE có hai phiên bản: học có giám sát và không giám sát. Trong phiên bản học không giám sát, SimCSE tạo các cặp dương (positive pairs) bằng cách đưa cùng một câu qua BERT hai lần với dropout khác nhau, từ đó tạo ra hai biểu

diễn hơi khác nhau nhưng có cùng ngữ nghĩa. Trong phiên bản có giám sát, cặp dương là các cặp câu tương đồng, còn các cặp âm (negative pairs) là những cặp câu không liên quan, từ tập dữ liệu NLI (Natural Language Inference).

Bằng cách này, SimCSE học được cách tạo ra các vector biểu diễn có thể phân biệt rõ ràng giữa các câu có ngữ nghĩa tương tự và khác nhau. Phương pháp học tương phản giúp tối ưu hóa mô hình bằng cách đẩy các vector biểu diễn của câu tương tự lại gần nhau và đẩy các câu không liên quan ra xa. Từ đó SimCSE tạo ra các nhúng câu có chất lượng cao, dễ áp dụng cho các bài toán như tìm kiếm câu, so sánh câu, hoặc phân loại văn bản.

2.8 Khai phá luật kết hợp

Khai phát luật kết hợp (Association rule mining) [2] là một kỹ thuật trong khai thác dữ liệu, thường được sử dụng để phát hiện mối quan hệ giữa các biến trong một tập dữ liệu lớn. Ý tưởng chính của nó là tìm ra các quy tắc mà từ đó có thể dự đoán sự xuất hiện của một mục này dựa trên sự xuất hiện của các mục khác. Một quy tắc thường được thể hiện dưới dạng "Nếu A, thì B" (A $\rightarrow$ B), trong đó A và B là các tập hợp của các mục.

Để đánh giá một quy tắc, hai chỉ số chính thường được sử dụng là độ tin cậy (confidence) và độ phổ biến (support).

Support của một tập hợp các mục A và B được tính bằng công thức:

$$\text{Support}(A \rightarrow B) = \frac{\text{Số lần xuất hiện của } A \cup B}{\text{Tổng số giao dịch}}$$

Chỉ số này đo lường tần suất mà cả A và B cùng xuất hiện trong tập dữ liệu.

Confidence của quy tắc A $\rightarrow$ B được tính theo công thức:

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Số lần xuất hiện của } A \cup B}{\text{Số lần xuất hiện của A}}$$

Chỉ số này thể hiện xác suất của B xảy ra khi A đã xảy ra.

2.9 Phép đo cosine similarity

Cosine similarity là một phương pháp phổ biến để đo lường độ tương đồng giữa hai vector trong không gian nhiều chiều. Đặc biệt, trong lĩnh vực xử lý ngôn ngữ

tự nhiên, nó được sử dụng để đánh giá mức độ tương đồng giữa các văn bản hoặc các từ.

Cách hoạt động của cosine similarity rất đơn giản. Nó tính toán cosine của góc giữa hai vector, biểu diễn cho hai đối tượng cần so sánh. Kết quả của phép đo này nằm trong khoảng từ -1 đến 1. Nếu cosine similarity bằng 1, điều đó có nghĩa là hai vector hoàn toàn tương đồng, bằng 0 thể hiện rằng chúng không có sự tương đồng, và -1 cho thấy chúng hoàn toàn đối lập.

Công thức tính cosine similarity giữa hai vector A và B được biểu diễn như sau:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|}$$

Trong đó:

- $A.B$ à tích vô hướng của A và B.
- ||A|| và ||B|| là độ dài của các vector A và B.

2.10 Ma trận nhầm lẫn

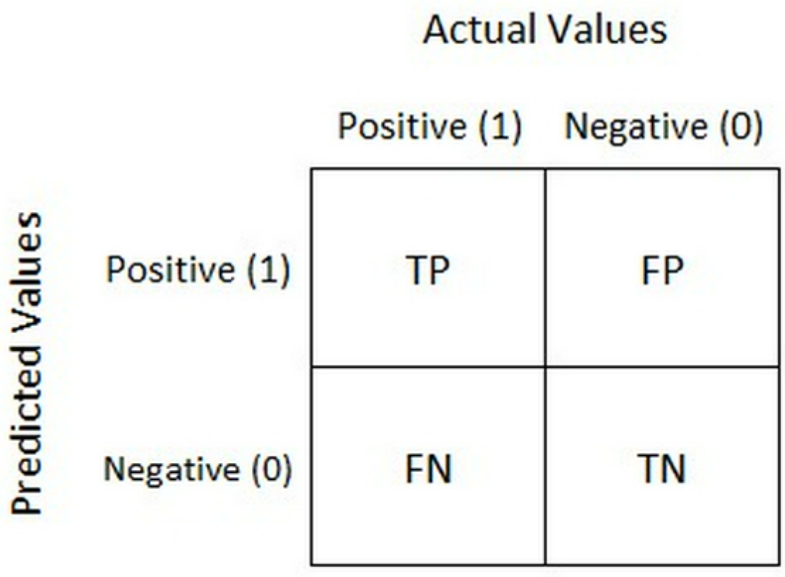
Confusion matrix (hay ma trận nhầm lẫn) là một công cụ dùng để đánh giá kết quả của những bài toán phân loại (classification). Nó xem xét những chỉ số về độ chính xác và độ phủ của các dự đoán cho từng lớp (class) phân loại. Một confusion matrix cho một lớp C gồm 4 đại lượng đối với mỗi lớp được trình bày trong Hình 18.

Trong đó:

- TP (True Positive): là số lượng dự đoán (predicted) chính xác. Đối tượng thuộc lớp C được dự đoán vào lớp C.
- FP (False Positive): là số lượng dự đoán sai lệch. Đối tượng thuộc lớp khác bị dự đoán thành lớp C.
- FN (False Negative): là số lượng dự đoán sai lệch một cách gián tiếp. Đối tượng thuộc lớp C được dự đoán thành lớp khác
- TN (True Negative): là số lượng dự đoán chính xác một cách gián tiếp. Đối tượng thuộc lớp khác không được dự đoán thành lớp C

Độ chính xác (Precision) được tính theo công thức:

$$P = \frac{TP}{TP + FP}$$



Hình 18: Hình ảnh mô tả Confusion Matrix [48]

Vì $TP + FP = N_{pred}$ là số lượng các dự đoán nên thực chất độ chính xác bằng tỉ lệ số dự đoán đúng trên tổng số dự đoán.

$$P = \frac{TP}{N_{pred}}$$

Độ phủ (Recall) được tính theo công thức:

$$R = \frac{TP}{TP + FN}$$

Vì $TP + FN = N_{actual}$ là số lượng các giá trị thực tế (hay còn gọi là ground truth) nên thực chất độ phủ bằng tỉ lệ số dự đoán đúng trên tổng số giá trị thực.

$$R = \frac{TP}{N_{actual}}$$

Độ đo F1 được tính theo công thức:

$$F1 = \frac{2 * P * R}{P + R}$$

Một độ đo khác cũng thường được sử dụng là độ chính xác Accuracy. Accuracy được tính đơn giản là tỉ lệ tổng số dự đoán đúng trên tổng số dự đoán của tất cả các lớp.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

Để tổng hợp kết quả dự đoán từ các nhãn khác nhau, người ta hay dùng đến các phương pháp tính là **micro average**, **macro average** và **weighted average**.

Micro average tính toán các độ đo bằng cách tổng hợp tất cả các mẫu từ các lớp và tính toán trên toàn bộ dữ liệu, không phân biệt từng lớp riêng biệt. Ví dụ cách tính micro precision cho tất cả các lớp:

$$MicroPrecision = \frac{\sum TP}{\sum TP + FP}$$

Precision được tính dựa trên tổng các True Positives và tổng dự đoán trên toàn bộ các nhãn.

Macro average tính toán các độ đo bằng cách tính riêng cho từng lớp và sau đó lấy trung bình các giá trị. Mỗi lớp sẽ được tính một cách độc lập, và sau đó trung bình cộng của tất cả các lớp sẽ cho ra độ đo cuối cùng. Ví dụ cách tính macro precision cho tất cả các lớp:

$$MacroPrecision = \frac{1}{N} \sum_{i=1}^{N} Precision_i$$

Trong đó, N là số lớp, $Precision_i$ là độ chính xác của lớp thứ i.

Một cách tính khác quan tâm đến tỷ lệ của từng lớp hoặc nhãn dựa trên số lượng mẫu của chúng trong tổng thể là Weighted Macro Average.

$$WeightedPrecision = \sum_{i=1}^{n} w_i.Precision_i$$

Trong đó, w_i là trọng số của lớp thứ i, thường được tính bằng tỷ lệ mẫu của lớp đó: $w_i = \frac{n_i}{\sum_{i=1}^{N} n_i}$ với n_i là số lượng mẫu của lớp thứ i.

Cách tính Recall, F1 cho từng phương pháp cũng tương tự.

3 Các công trình liên quan

3.1 Đồ thị tri thức

Trong đề tài này, đồ thị tri thức cần được xây dựng thuộc loại chuyên biệt về một lĩnh vực, cụ thể hơn là lĩnh vực giáo dục. Một số đồ thị tri thức trong lĩnh vực này đã được phát triển như đồ thị tri thức cho môn học KnowEdu [5], mô hình ontology cho chương trình giảng dạy đại học và hồ sơ sinh viên EducOnto [18], lược đồ tri thức cho việc giảng dạy đại học [39], xác định cầu của thị trường lao động (mối quan hệ giữa giáo dục và tuyển dụng) [15]. Tuy nhiên, chưa có nhiều đồ thị được phát triển cho vấn đề hỗ trợ sinh viên trong nhà trường. Khi thực hiện xây dựng đồ thị này, nhóm đề xuất một kiến trúc mới để bao gồm một loại thông tin mới là "ý định". Ý định là một khái niệm trong lĩnh vực hiểu ngôn ngữ tự nhiên (Natural Language Understanding), đóng vai trò rất quan trọng trong các hệ thống hỏi đáp nói chung và trong hệ thống của chúng ta nói riêng.

Ngoài ra, ta cũng cần tham khảo đến các đồ thị tri thức tiếng Việt hiện đang có. Vương Thị Thái Yen cùng các đồng nghiệp tại Đại học Quốc gia Hà Nội đã phát triển phương pháp xây dựng đồ thị tri thức cho lĩnh vực pháp luật ở Việt Nam [47]. Một bài báo khác của Thạc sĩ Bùi Công Tuấn [4] sử dụng kĩ thuật trích xuất ý định mở cho tập dữ liệu giáo dục tiếng Việt của trường Đại học Bách Khoa. Đây cũng là bài báo mà nhóm sẽ sử dụng kết quả của nó để hỗ trợ cho công trình của nhóm.

Như đã đề cập trước đó, việc xây dựng đồ thị tri thức có hai hướng tiếp cận chính là bottom-up và top-down. Với top-down, các thông tin được rút trích trong giai đoạn Knowledge Acquisition sẽ dựa trên một lược đồ được định nghĩa từ trước mà người ta hay gọi là ontology. Với bottom-up, ta sẽ xây dựng ontology từ dữ liệu được rút trích. Đối với dữ liệu phi cấu trúc chẳng hạn như văn bản thuần túy, ta cần dùng các kĩ thuật phức tạp để hiện thực việc trích xuất. Các kĩ thuật dùng để trích tri thức cho hướng bottom-up chủ yếu là các kĩ thuật trích xuất thông tin mở (Open Information Extraction) - một kĩ thuật trích thông tin mà không cần lược đồ cho trước.

Dựa trên những phân tích đó, nhóm đề xuất một kiến trúc mới để xây dựng đồ thị tri thức cho vấn đề hiện tại theo hướng tiếp cận bottom-up với các kĩ thuật trích xuất thông tin mở để khai phá các kiểu thực thể cùng các kĩ thuật trích xuất thông tin trong văn bản tiếng Việt.

3.2 Trích xuất thông tin mở

Trích xuất thông tin mở (Open Information Extraction [51]) là một phương pháp trong xử lý ngôn ngữ tự nhiên nhằm mục đích tự động trích xuất các thông tin quan trọng từ văn bản mà không cần biết trước các kiểu thực thể hoặc quan hệ cụ thể. OIE tập trung vào việc trích xuất các bộ ba thông tin (chủ ngữ - vị ngữ - tân ngữ) từ các văn bản tự nhiên. Các thông tin này có thể bao gồm các thực thể , các thuộc tính của thực thể, và các quan hệ giữa các thực thể. Ví dụ, từ câu "Edison invented the light bulb" OIE có thể trích xuất được bộ ba ("Edison", "invented", "light bulb") [51]. Các thông tin này có thể được sử dụng để xây dựng các đồ thị tri thức hoặc để hiểu rõ hơn về ngữ nghĩa của văn bản.

Các phương pháp tiếp cận trong OIE đã phát triển qua nhiều giai đoạn, từ các phương pháp dựa trên quy tắc đến các phương pháp sử dụng học sâu hiện đại.

- *Phương pháp dựa trên quy tắc (Rule-based method)*: phương pháp này sử dụng các quy tắc ngữ pháp hoặc cú pháp được định nghĩa trước để trích xuất thông tin từ văn bản. Một ví dụ tiêu biểu là hệ thống TEXTRUNNER (2007) [51] được xem là một trong những hệ thống OIE đầu tiên. TEXTRUNNER sử dụng các bộ phân tích cú pháp để xác định cấu trúc câu và áp dụng các quy tắc để trích xuất các bộ ba thông tin từ văn bản web quy mô lớn.

- *Phương pháp dựa trên mẫu (Pattern-based methods)*: phương pháp này dựa trên việc tìm kiếm các mẫu ngữ pháp hoặc cú pháp phổ biến trong văn bản để trích xuất thông tin. Hệ thống REVERB (2011) [14] là một ví dụ, sử dụng các mẫu biểu thức chính quy để xác định và trích xuất các quan hệ hợp lệ trong câu, cải thiện chất lượng trích xuất so với các phương pháp trước đó.

- *Phương pháp học máy (Machine Learning-based methods)*: phương pháp này sử dụng các thuật toán học máy để học cách trích xuất thông tin từ dữ liệu đã được gán nhãn. Hệ thống WOE (2010) [50] sử dụng các mô hình học máy dựa trên Wikipedia và Freebase để cải thiện độ chính xác của việc trích xuất so với các phương pháp dựa trên quy tắc.

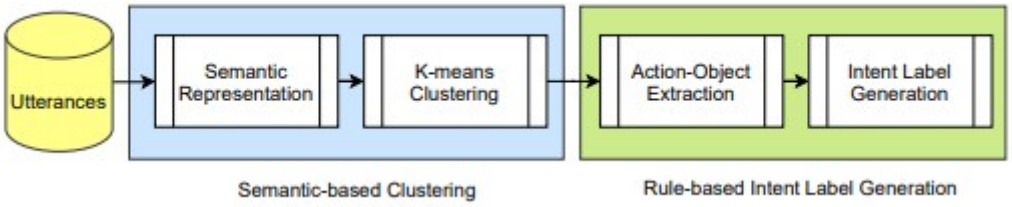
Trong bối cảnh của miền giáo dục, dữ liệu thường là không có cấu trúc và đến từ nhiều nguồn khác nhau như tài liệu giảng dạy, bài viết học thuật, thông tin sinh viên,.... Mục tiêu của việc xây dựng đồ thị tri thức trong miền này là tổ chức và quản lý thông tin về các khái niệm như "khóa học", "giảng viên", "sinh viên", "chương trình đào tạo",... và các quan hệ giữa chúng.

Việc áp dụng các kỹ thuật của OIE có thể giúp ta trích xuất các thực thể và quan hệ từ văn bản trong miền giáo dục. Ví dụ, OpenIE có thể phát hiện và trích

xuất các bộ ba như ("sinh viên A", "thuộc", "khoá B") hoặc ("sinh viên C", "đăng ký", "môn học D"). Sau đó, các thông tin này được đưa vào quy trình khám phá thực thể dựa trên ý định để xác định các khái niệm có thể đóng vai trò làm các nút trong đồ thị tri thức.

Khám phá ý định dựa trên tập dữ liệu mở cũng là một dạng của trích xuất thông tin mở OIE. Trong đó bài báo [29] đề xuất một phương pháp học không giám sát với hai bước: gom cụm ngữ nghĩa và tạo nhãn dựa trên phân tích cú pháp. Trong đó:

- Ở bước gom cụm ngữ nghĩa, các câu văn bản được chuyển đổi thành các vector trong không gian thông qua mô hình Sentence-BERT [38]. Sau đó, thuật toán K-means [32] được sử dụng để phân nhóm các câu này thành K cụm dựa trên độ tương đồng ngữ nghĩa giữa chúng.
- Sau khi hoàn thành phân cụm, các cặp hành động - đối tượng được trích xuất từ mỗi cụm bằng cách sử dụng bộ phân tích phụ thuộc spaCy. Cặp xuất hiện thường xuyên nhất trong từng cụm được chọn làm nhãn cho cụm đó, đảm bảo rằng nhãn phản ánh chính xác nội dung của các câu trong cụm.



Hình 19: Khám phá ý định mở bằng học không giám sát [29]

Một bài báo khác cũng được nhóm tham khảo tới là khám phá ý định bằng phương pháp học không giám sát trên chính tập dữ liệu giáo dục của trường Đại học Bách Khoa TP.HCM [4].

3.3 Nhận diện thực thể có tên

Nhận diện thực thể có tên (NER) là nhiệm vụ trong Xử lý ngôn ngữ tự nhiên mà nó xác định và phân loại các vật thể có tên trong văn bản phi cấu trúc. Các thực thể thường thuộc về các kiểu ngữ nghĩa rộng như người, địa điểm, tổ chức. Những kiểu thực thể có thể thu hẹp hơn nữa, chẳng hạn sách, tạp chí, báo chí,... Các nghiên cứu NER trên miền tổng quát thường dùng bộ dữ liệu có kiểu thực thể tổng quát như CoNLL-2003 [44] gồm các nhãn PER (người), LOC (địa điểm),

ORG (tổ chức), MISC (không rõ). Tuy nhiên đối với miền chuyên biệt (specific domain), kiểu dữ liệu sẽ cụ thể hơn phụ thuộc vào miền dữ liệu đang làm việc. Ví dụ kiểu gene hay protein trong miền sinh học, kiểu hóa chất trong miền hóa học. Một số nghiên cứu về miền giáo dục như EduNER [25] cũng thiết kê các kiểu dữ liệu riêng để sử dụng. KnowEdu [5] đề xuất một mô hình NER chỉ nhận diện thực thể hay khái niệm mà không quan tâm đến kiểu dữ liệu.

Các mô hình NER cho miền cụ thể có thể dự đoán kiểu cụ thể cho các thực thể ngay hoặc cũng có thể phân làm hai bước. Bước đầu tiên, mô hình NER cho miền tổng quát sẽ được áp dụng để nhận biết các thực thể tổng quát như người, địa điểm, tổ chức. Bước tiếp theo, người ta sẽ phân loại các kiểu chi tiết hơn cho các thực thể vừa trích được. Nhiệm vụ ở bước thứ hai này còn gọi là phân loại kiểu chi tiết (fine-grained entity typing), thường dễ thấy trong các bài toán rút trích thông tin để xây dựng đồ thị tri thức. Mô hình hai bước này áp dụng khi các kiểu cụ thể có thể được phân hóa nhỏ hơn từ các kiểu tổng quát.

Các công trình nổi tiếng về NER cho tiếng Việt hiện nay cũng bao gồm các NER cho miền chung và cho lĩnh vực chuyên biệt. Các bộ dữ liệu trong các cuộc thi của VLSP như VLSP 2016, VLSP 2018 được sử dụng nhiều cho các mô hình NER miền chung. Các bộ dữ liệu này cũng sử dụng các lớp của CoNLL-2003 và hướng dẫn đánh nhãn dựa trên đó. Các bộ dữ liệu cho miền riêng chủ yếu là lĩnh vực y học, chúng sử dụng bộ dữ liệu về COVID-19 nổi tiếng. Bộ dữ liệu cũng có các kiểu thực thể cơ bản như người, địa điểm, tổ chức và bao gồm nhiều kiểu thực thể y học khác.

3.4 Phân loại ý định và Điền trường thông tin

Như đã đề cập, phân loại ý định có thể được xem như là một bài toán phân loại có giám sát, ở đó các đặc trưng của văn bản sẽ được tạo lập và truyền qua một thuật toán phân loại để dự đoán nhãn trong số nhãn định sẵn. Các nghiên cứu từ trước đến nay tập trung vào phát triển các giải thuật, mô hình cho cả kĩ thuật tạo đặc trưng và giải thuật phân lớp. Các kĩ thuật trích đặc trưng ban đầu chỉ trích những thông tin về mối quan hệ giữa các từ trong câu, về sau mở rộng hơn để lấy thông tin cú pháp, thông tin ngữ nghĩa của câu, thông tin ngữ cảnh trong câu. Các kĩ thuật có thể kể đến như dependency parsing, Bag-of-words (BOW), n-gram, word2Vec đến các mô hình học sâu có khả năng học đặc trưng như CNN, GioVe, BiLSTM, BERT. Thuật toán phân lớp từ những thuật toán cổ điển như SVM đến những thuật toán cho mô hình học sâu, thường dùng nhất là Softmax Regression.

Điền thông tin trường được cài đặt như bài toán gán nhãn chuỗi, mục đích

là gán nhãn cho từng 'token' (từ hay tiểu từ) có trong câu. Ngay từ lâu, một số mô hình xác suất như HMM, CRF đã sử dụng cho bài toán. Với thời kỳ của các mô hình học sâu, CRF được sử dụng như một layer phụ trợ cho các mô hình này. Các mô hình được sử dụng như họ RNN, các mô hình Transformers với cơ chế attention (giúp xử lý các ngữ cảnh dài).

Phương pháp joint xuất hiện để khắc phục tình trạng lan truyền lỗi (error propagation) của cơ chế pipeline, cũng như nắm bắt được mối quan hệ giữa ý định và trường mà cho ra kết quả tốt hơn. Phương pháp joint còn có lợi khi chỉ phải huấn luyện và đánh giá trên một mô hình duy nhất mà không phải nhiều mô hình độc lập. Tuy nhiên, các mô hình joint vẫn gặp phải tình trạng cần nhiều dữ liệu được gán nhãn để đạt được độ chính xác cao. Ngoài ra, các mô hình joint đều rất phức tạp và tốn nhiều thời gian để huấn luyện.

Gần đây, việc xem nhiều nhiệm vụ Xử lý ngôn ngữ tự nhiên thành một nhiệm vụ chung duy nhất là Đọc hiểu máy đang được ưa chuộng. Cơ chế của phương pháp này rất đơn giản, giống như việc ta đưa vấn đề cho một chatbot hay trợ lý ảo giải quyết hộ. Với từng nhiệm vụ Phân loại ý định và Điền trường thông tin nói riêng, mô hình Đọc hiểu máy có thể nắm bắt ngữ nghĩa của câu nói mà cho ra kết quả dự đoán tốt mà không cần phải huấn luyện một tập dữ liệu lớn. Mô hình cũng có khả năng dự đoán tốt hơn với những nhãn dữ liệu mới chưa từng thấy trong tập huấn luyện, điều mà gần như các phương pháp học giám sát trước không làm được. Có nhiều mô hình tiền huấn luyện cho nhiệm vụ Đọc hiểu máy ngày nay mà ta có thể sử dụng ngay mà không cần tùy chỉnh phức tạp hoặc huấn luyện lại từ đầu. Điều này cũng là một thế mạnh của phương pháp Đọc hiểu máy. Tuy nhiên, mô hình Đọc hiểu máy cần thêm dữ liệu là câu hỏi bên cạnh ngữ cảnh. Các câu hỏi này không có sẵn ở tập dữ liệu đầu vào, nên sẽ phải được tạo ra thêm nhờ các chiến lược sinh câu hỏi. Chất lượng kết quả của phương pháp Đọc hiểu máy cũng phụ thuộc rất nhiều vào các câu hỏi được sinh ra này.

Một số bài toán khác được giải quyết bằng phương pháp Đọc hiểu máy có thể kể đến như trích xuất thực thể [26], trích xuất mối quan hệ [27] [22], trích xuất sự kiện [23], [28], [31], [13]. Bài toán trích xuất sự kiện có nhiều điểm chung với bài toán Phân loại ý định và Điền trường thông tin mà ta có thể tham khảo.

Gần đây xuất hiện nghiên cứu đầu tiên sử dụng hướng tiếp cận Đọc hiểu máy cho Điền trường thông tin [54], tuy nhiên nghiên cứu này vẫn chưa liên hệ với bài toán Phân loại ý định. Việc giải quyết nhiệm vụ Điền trường thông tin sử dụng hướng tiếp cận Đọc hiểu máy được thực hiện giống như nhiệm vụ NER, nhưng việc sử dụng toàn bộ các kiểu thực thể (slot type) để đặt câu hỏi sẽ làm bộ

dữ liệu lớn hơn rất nhiều nếu số lượng slot type quá lớn. Việc kết hợp với ý định có trong câu sẽ thu hẹp lại phạm vi câu hỏi được tạo ra cho mô hình Đọc hiểu máy.

Có nhiều nghiên cứu về Phân loại ý định và Điền trường thông tin nhưng các nghiên cứu vẫn khá lan man và không được ứng dụng cụ thể trong một lĩnh vực nào cả. Các tập dữ liệu của bài toán hiện có bao gồm các lĩnh vực như hàng không, khách sạn,.. hoặc tập dữ liệu đa miền. Theo những gì nhóm được biết và tìm hiểu, hiện tại vẫn chưa có nghiên cứu nào giải bài toán trên miền giáo dục.

Điều may mắn là một số nghiên cứu bài toán cho ngôn ngữ Việt đã được phát triển, tiêu biểu là bài "Intent Detection and Slot Filling for Vietnamese" [10]. Bài báo đề xuất một framework joint giải quyết cả hai nhiệm vụ Phân loại ý định và Điền trường thông tin sử dụng kiến trúc mô hình BERT kết hợp với thuật toán CRF. Bài viết còn giới thiệu tập dữ liệu PhoATIS được tạo ra dựa trên tập dữ liệu nổi tiếng ATIS thông qua dịch thuật. Và vì thế, miền dữ liệu được sử dụng trong bài báo cũng là lĩnh vực hàng không sử dụng trong ATIS. Bài viết thử nghiệm framework với nhiều mô hình mã hóa khác nhau và cho ra kết quả khả quan.

Với lượng dữ liệu chuẩn bị cho đề tài không quá lớn, các thuật toán học có giám sát có thể không cho ra kết quả đủ tốt. Ngoài ra, miền dữ liệu của nhóm được lấy từ ứng dụng thực tiễn nên lượng nhãn sử dụng cho đề tài sẽ không cố định mà còn có thể xuất hiện thêm nhiều nhãn mới, gây khó khăn cho các thuật toán học có giám sát khi dự đoán. Có nhiều phương pháp để giải quyết các vấn đề kể trêm nhưng nhóm chọn phương pháp Đọc hiểu máy cho bài toán Phân loại ý định và Điền trường thông tin vì tính đơn giản của nó.

3.5 Đọc hiểu máy

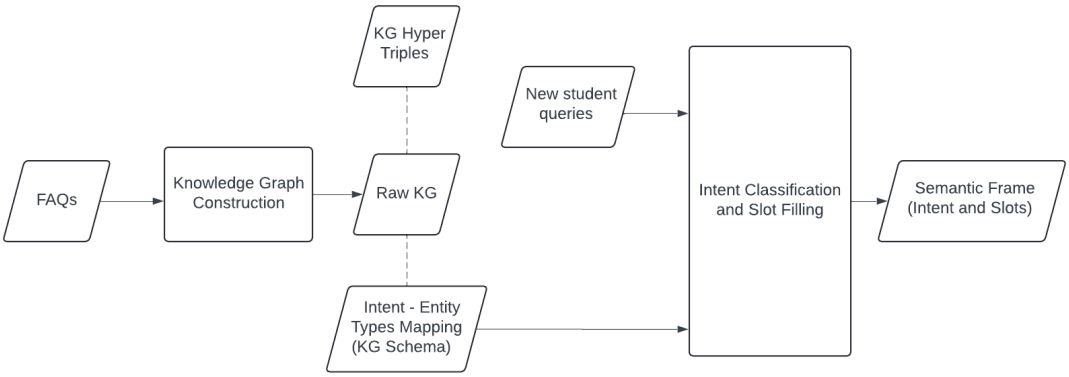
Sự hình thành của các mô hình ngôn ngữ tiền huấn luyện (pre-trained language model) này làm cho đề tài Đọc hiểu máy trở nên bùng nổ. Tuy nhiên, các nghiên cứu, mô hình cho Đọc hiểu máy hiện nay vẫn dựa trên miền chung (generic domain) chứ chưa được phát triển cho một lĩnh vực cụ thể. Các nghiên cứu về đề tài Đọc hiểu máy trên lĩnh vực giáo dục còn khan hiếm.

Những năm gần đây, do ảnh hưởng từ sức nóng của đề tài, nhiều nghiên cứu Đọc hiểu máy cho ngôn ngữ Việt Nam cũng ra đời. Các tập dữ liệu như UIT-ViQuAD[35], UIT-ViQuAD 2.0[20],... được tạo ra để làm benchmark đánh giá cho các mô hình tiếng Việt. Một số mô hình tiền huấn luyện vận dụng những kiến trúc hiện đại cũng được phát triển. Những cuộc thi phát triển mô hình Đọc hiểu máy như VLSP[20] cho tiếng Việt cũng được tổ chức để tìm ra những mô hình các độ chính xác cao.

4 Phương pháp thực hiện

Với mục tiêu đề tài, nhóm sẽ giải quyết hai bài toán chính đó là Xây dựng một đồ thị tri thức cùng với Phát hiện ý định và điền trường thông tin. Hai bài toán này trong đề tài có liên quan với nhau. Với nhiệm vụ Phát hiện ý định và Điền trường thông tin, ta phải có trước một danh sách ý định và danh sách kiểu trường. Tuy nhiên trong một số nghiên cứu sử dụng phương pháp joint cho bài toán, người ta thường cần biết phân phối giữa ý định và kiểu trường, xem các kiểu trường và ý định nào thường đi chung với nhau. Đặc biệt đối với phương pháp Đọc hiểu máy được dùng trong nhiệm vụ Điền trường thông tin được sử dụng trong bài nghiên cứu của nhóm, việc biết được kiểu thực thể nào (trong đề tài này, nhóm sử dụng kiểu thực thể như là kiểu trường) đi với ý định nào sẽ làm giảm bớt đi lượng câu hỏi cần phải tạo dẫn đến sự gia tăng hiệu suất mô hình và gia tăng độ chính xác (vì nó làm giảm khả năng mô hinh dự đoán dư thừa).

Dựa trên thực tế đó, nhóm đề xuất một framework chung cho cả hai bài toán ở hình 20. FAQs sẽ là bộ dữ liệu dùng cho quá trình Xây dựng đồ thị tri thức (Knowledge Graph Construction). Kết quả của quá trình sẽ cho ra một Đồ thị tri thức thô (Raw KG) bao gồm một lược đồ cho biết kiểu thực thể nào sẽ hay đi với ý định nào và các siêu bộ ba mô hình ở mức độ thực thể cụ thể. Lược đồ đồ thị tri thức sẽ được sử dụng làm nguồn thông tin phụ trợ cho nhiệm vụ Phân loại ý định và Điền trường thông tin. Quá trình này rút ra được ý định và các thông tin liên quan trong các truy vấn mới mà sinh viên gửi vào.



Hình 20: Framework chung cho đề tài

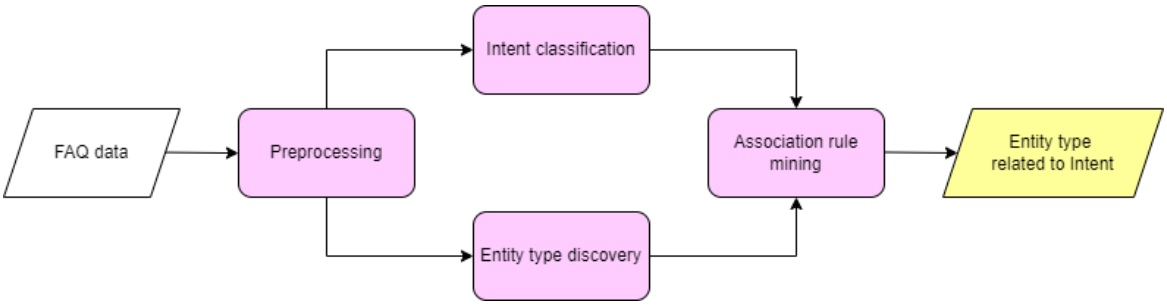
Kết quả của quá trình Phát hiện ý định và Điền trường thông tin có thể được cấu trúc thành dạng siêu bộ ba của đồ thị tri thức. Vì thế, nó có thể được tích hợp thêm vào trong đồ thị tri thức, làm giàu tri thức hơn nữa. Tuy nhiên, vấn đề

này sẽ nằm ngoài phạm vi nghiên cứu của nhóm.

4.1 Xây dựng đồ thị tri thức dựa trên ý định

Một đồ thị tri thức phải bao gồm cả lược đồ và các thông tin biểu diễn dựa trên lược đồ. Các công trình xây dựng đồ thị tri thức theo hướng bottom-up trước đây vẫn chỉ dừng lại ở bước khám phá các mối quan hệ giữa các thực thể với nhau. Các kiểu thực thể thường được cho trước trong các bài này. Tuy nhiên, nhóm nhận thấy việc thiết kế ra các kiểu thực thể này cũng không phải công việc dễ dàng mà cần có nhiều sự giúp đỡ và hợp tác giữa các chuyên gia trong lĩnh vực được khai phá. Đặc biệt, trong các lĩnh vực mới mẻ hoặc nguồn lực ít, rất khó để tìm được các chuyên gia này. Vì vậy, nhóm quyết định thử nghiệm khám phá các kiểu thực thể cho nhiệm vụ xây dựng đồ thị tri thức. Hướng tiếp cận này vừa tốn ít công sức của các chuyên gia, vừa có thể áp dụng cho các miền dữ liệu khác. Sau khi khám phá các kiểu thực thể, nhóm sẽ tìm mối liên hệ giữa nó và các ý định, nhằm xây dựng một lược đồ cho đồ thị.

Dựa trên ý tưởng đó, nhóm đề xuất framework NCA như hình 21 với mục tiêu khám phá được các kiểu thực thể và mối quan hệ giữa kiểu thực thể và ý định người dùng từ dữ liệu có trong miền giáo dục.



Hình 21: Framework NCA

Framework NCA mà nhóm đề xuất gồm những bước xử lí sau:

- *Preprocessing*: dữ liệu FAQs đầu vào bao gồm các đoạn văn bản yêu cầu, câu hỏi có cấu trúc phức tạp và không nhất quán đến từ người dùng. Dữ liệu này đòi hỏi cần phải đi qua một bước tiền xử lý để làm sạch cũng như chuẩn hoá để tối ưu các bước phân tích phía sau.

- *Entity type discovery*: bước này sử dụng kĩ thuật nhận diện thực thể có tên (NER) và các kĩ thuật liên quan đến gom cụm để tìm ra các cụm liên quan đến nhau về ngữ nghĩa, từ đó xác định được kiểu thực thể (entity type).

- *Intent classification*: bước này cố gắng sử dụng phương pháp đọc hiểu máy (Machine Reading Comprehension) để tìm ra ý định từ văn bản đầu vào bằng các câu hỏi đơn giản.

- *Association rule*: bước này áp dụng luật kết hợp để khai phá mối quan hệ giữa ý định và các kiểu thực thể có liên quan đến ý định này.

Tựu chung lại, dữ liệu đầu vào sau khi đi qua các bước xử lí của NCA framework sẽ cho đầu ra là mối quan hệ giữa ý định và các kiểu thực thể. Nguồn tri thức này là một đóng góp hữu ích cho việc xây dựng đồ thị tri thức.

4.1.1 Tiền xử lí dữ liệu

Dữ liệu đầu vào của nhóm được lấy từ hệ thống hỏi đáp (BKSI) của trường đại học Bách Khoa TP.HCM, bao gồm các đoạn hội thoại giữa sinh viên và nhân viên trả lời. Đây là một tập dữ liệu phức tạp, chứa nhiều đoạn văn dài, có cả các kí tự đặc biệt, thông tin không cần thiết, và các đoạn hội thoại thiếu liên kết. Vì vậy, để đảm bảo chất lượng đầu vào cho các bước xử lý tiếp theo, quá trình tiền xử lý dữ liệu là bước vô cùng quan trọng. Phần này sẽ tập trung trình bày các bước tiền xử lý dữ liệu mà nhóm đã thực hiện, nhằm giải quyết các vấn đề thực tế mà dữ liệu đặt ra.

Quá trình xử lý dữ liệu của nhóm bao gồm các bước sau:

- *Loại bỏ các đoạn tin nhắn không có ý nghĩa*: trong hệ thống hội thoại, không thiếu những đoạn tin nhắn chỉ mang tính kỹ thuật hoặc thử nghiệm, chẳng hạn như các tin nhắn thử hệ thống, tin nhắn tạo và đóng câu hỏi, hoặc cập nhật trạng thái hội thoại. Những đoạn tin nhắn này không đóng góp vào giá trị thông tin của tập dữ liệu, do đó cần được loại bỏ để tập trung vào những nội dung thực sự có ý nghĩa.

- *Xử lý các ký tự đặc biệt và văn bản gây nhiễu*: các ký tự đặc biệt xuất hiện trong văn bản có thể gây ảnh hưởng đến quá trình xử lý ngôn ngữ tự nhiên và phân tích dữ liệu. Vì vậy, nhóm đã thực hiện việc làm sạch tập dữ liệu bằng cách loại bỏ hoặc chuyển đổi các ký tự không cần thiết, đảm bảo rằng nội dung tin nhắn trở nên chuẩn hóa và dễ hiểu hơn cho các bước phân tích tiếp theo.

- *Chuẩn hóa văn bản và loại bỏ khoảng trắng dư thừa*: một vấn đề thường gặp khác trong dữ liệu hội thoại là các khoảng trắng dư thừa giữa các từ, hoặc giữa các ký tự và dấu câu. Những lỗi nhỏ này có thể dẫn đến việc nhận diện sai ngữ nghĩa của câu trong quá trình phân tích, làm giảm độ chính xác của

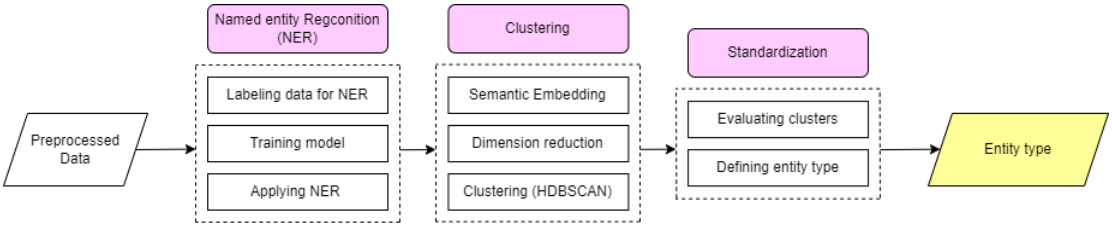
mô hình. Để khắc phục điều này, nhóm đã triển khai một bước làm sạch kỹ lưỡng, loại bỏ tất cả các khoảng trắng dư thừa, đảm bảo rằng văn bản được chuẩn hóa trước khi đi vào các bước phân tích sâu hơn.

- *Tái cấu trúc dữ liệu*: sau khi làm sạch và loại bỏ các yếu tố nhiễu, nhóm chỉ giữ lại những tin nhắn chứa yêu cầu từ phía người dùng (sinh viên). Việc chỉ lấy những yêu cầu từ người dùng cũng giúp làm rõ các mục tiêu chính của tập dữ liệu, giúp dễ dàng hơn trong việc trích xuất thông tin và phân tích hành vi của người dùng.

Những bước xử lý trên đóng vai trò quan trọng trong việc tối ưu hóa tập dữ liệu và nâng cao chất lượng đầu ra. Việc làm sạch và cấu trúc lại dữ liệu giúp nhóm loại bỏ các yếu tố gây nhiễu và tạo nền tảng vững chắc cho các phân tích tiếp theo như trích xuất thực thể, phát hiện ý định, hay phân tích mối quan hệ giữa các thông tin này với nhau.

4.1.2 Khám phá kiểu thực thể

Quy trình mà nhóm xây dựng để tìm ra kiểu thực thể trong miền dữ liệu gồm 3 bước cơ bản như hình 22:



Hình 22: Pipeline dùng cho khám phá kiểu thực thể từ dữ liệu

- *Nhận diện thực thể có trong dữ liệu*: ở bước này nhóm sẽ chuẩn bị một bộ dữ liệu cho nhiệm vụ nhận diện thực thể có tên, huấn luyện nó trên một mô hình huấn luyện trước, sau cùng là áp dụng mô hình đó vào tập dữ liệu để thu được thực thể.

- *Gom cụm ngữ nghĩa*: những thực thể tìm được sẽ bước vào quá trình gom cụm bằng cách mã hoá các từ, cụm từ này thành vector, giảm số chiều và cuối cùng là sử dụng một thuật toán gom cụm để tìm ra các thực thể có cùng ngữ nghĩa.

- *Chuẩn hoá các kiểu thực thể* : dựa vào kết quả gom cụm, chúng ta có thể định nghĩa các kiểu thực thể từ các cụm, kèm theo đó là ánh xạ ngược kiểu thực thể cho các thực thể tương ứng.

4.1.2.a Nhận diện thực thể

Nhận thấy bài toán khám phá kiểu thực thể mà nhóm đang thực hiện cũng là một dạng của bài toán trích xuất thông tin trên nguồn mở, trong đó các cách tiếp cận từ những bài báo mà nhóm tham khảo có đặc điểm chung là đều trích xuất các thông tin (có thể là ý định, thực thể, mối quan hệ,...). Sau đó tiến hành phân tích sâu hơn bằng phương pháp gom cụm để tìm ra sự liên quan ngữ nghĩa giữa các thông tin này.

Để có thể tìm ra các kiểu thực thể trong miền giáo dục, đầu tiên nhóm sẽ cố gắng trích xuất tất cả các từ hoặc cụm từ được xem là thực thể và một trong những phương pháp được nhóm nghĩ đến là sử dụng Named Entity Regconition để thực hiện điều đó.

Như đã đề cập ở phần công trình liên quan, ta có thể sử dụng bộ nhãn miền chung (PER, LOG, ORG và MISC) nhằm trích các thực thể, sau đó ta có thể phân cụm chúng để ra các kiểu thực thể chuyên biệt hơn. Đây là bộ nhãn phổ biến được rất nhiều mô hình NER dành cho tiếng Việt. Với cách tiếp cận này, ta có thể dễ dàng tận dụng được các mô hình NER được huấn luyện trước mà không cần phải huấn luyện lại với nhiều dữ liệu. Mô hình nhóm lựa chọn cho nhiệm vụ rút trích thực thể cũng là một biến thể của mô hình tiền huấn luyện cho nhiệm vụ nhận diện thực thể có tên trên tập dữ liệu VLSP 2018.

Với mục tiêu của là rút trích tất cả các thực thể có trong lĩnh vực giáo dục mà không quan tâm đến kiểu của nó là gì, khi đánh nhãn cho tập dữ liệu, ngoài những thực thể dễ dàng nhận diện được thuộc các kiểu PER, ORG, LOC, nhóm sẽ tìm tất cả những từ, cụm từ nào có vẻ là thực thể và đánh cho nó nhãn MISC. Dưới đây là bảng so sánh cách đánh nhãn của nhóm cùng với cách đánh của VLSP 2018:

	VLSP	**Nhóm**
PER	Tên của người, động vật hay nhân vật hư cấu và danh xưng	Giống VLSP
ORG	Tên các tổ chức, cơ quan, công ti, thương hiệu	Giống VLSP
LOC	Các thực thể có tọa độ địa lí nhất định	Giống VLSP
MISC	Dùng để giải quyết một số trường hợp nhập nhằng và markup tên tác phẩm, sự kiện, thương hiệu có chứa tên thuộc 3 loại trên	Lợi dụng nhãn này để gán cho những thực thể chưa rõ kiểu có trong miền giáo dục

Bảng 3: So sánh cách gán nhãn dữ liệu theo VLSP 2018 và nhóm

Dựa vào bảng trên ta có thể thấy các thực thể chỉ tên người, tổ chức và địa điểm sẽ được giữ nguyên theo cách gán nhãn của VLSP 2018. Riêng với nhãn

MISC dùng để gán cho một số trường hợp nhập nhằng, không phân biệt rõ ràng giữa 3 nhãn PER, ORG, LOC hoặc nằm ngoài 3 nhãn này (theo VLSP 2018), sẽ được nhóm tận dụng để gán cho tất cả các thực thể còn lại. Việc làm này mang lại một số ưu điểm như sau:

- Giúp mô hình tìm ra những thực thể có trong miền giáo dục mà không cần phải xác định cụ thể kiểu thực thể của chúng, giải quyết được hai vấn đề nêu trên.
- Giảm thiểu gánh nặng cho quá trình gán nhãn thủ công khi chỉ cần xác định 3 nhãn cố định thay vì hàng chục nhãn có thể có trong tập dữ liệu mới.

Tập dữ liệu đã được gán nhãn cho nhiệm vụ nhận diện thực thể sẽ được dùng để huấn luyện trên một biến thể của mô hình tiền huấn luyện. Để tăng độ hiệu quả và hiểu ngữ cảnh của mô hình, nhóm ưu tiên lựa chọn những mô hình tiên tiến và đã huấn luyện nhiệm vụ nhận diện thực thể có tên trên tập dữ liệu tiếng Việt. Bước cuối cùng trong quá trình này là áp dụng mô hình đầu ra cho toàn bộ tập dữ liệu để thu được tất cả các thực thể, chuẩn bị cho quá trình gom cụm và phân tích sâu hơn.

4.1.2.b Gom cụm

Sau khi đã trích xuất được các thực thể từ văn bản, chúng ta cần phân loại các thực thể này thành từng nhóm có chung một kiểu bằng phương pháp gom cụm. Việc gom cụm không chỉ giúp xác định các kiểu thực thể có trong một miền dữ liệu mới mà còn mở ra khả năng khai thác thêm các thông tin liên quan đến từng kiểu thực thể. Quy trình gom cụm thực thể bao gồm ba bước chính: biểu diễn thực thể thành vector, giảm số chiều, và cuối cùng là áp dụng thuật toán gom cụm.

Bước đầu tiên của quy trình này là chuyển đổi các thực thể được trích xuất thành các vector số. Việc chuyển đổi này có ý nghĩa quan trọng, vì nó đưa các thực thể từ dạng ngôn ngữ tự nhiên sang không gian toán học, nơi mà các thuật toán gom cụm có thể xử lý được. Mục tiêu của tiến trình này là giữ lại thông tin về ngữ nghĩa của thực thể, đảm bảo rằng các thực thể tương đồng về nghĩa sẽ có khoảng cách gần nhau trong không gian vector.

Có nhiều phương pháp biểu diễn từ (word embeddings) để lựa chọn, tùy thuộc vào tính chất của dữ liệu và yêu cầu của bài toán. Trong quá trình thực hiện, nhóm đã thử nghiệm với nhiều mô hình embedding, trong đó có SBERT và SimCSE, vốn nổi tiếng trong việc giữ được ngữ nghĩa của từ/cụm từ sau khi biến đổi thành vector. Các mô hình này đặc biệt phù hợp với nhiệm vụ này vì khả năng ánh xạ các thực thể vào không gian vector mà vẫn duy trì được sự tương đồng về ngữ

nghĩa giữa các thực thể có liên quan.

Sau khi các thực thể đã được biểu diễn dưới dạng vector, không gian vector này thường có số chiều rất lớn, có thể lên đến hàng trăm chiều. Điều này không chỉ làm tăng độ phức tạp của quá trình tính toán mà còn gây khó khăn cho việc phân tích dữ liệu. Do đó, bước tiếp theo là giảm số chiều của các vector để tối ưu hóa quy trình. Giảm số chiều giúp loại bỏ các thông tin dư thừa hoặc không quan trọng, đồng thời duy trì cấu trúc quan hệ giữa các thực thể. Mục tiêu là làm cho các thực thể có ngữ nghĩa tương đồng vẫn được đặt gần nhau trong không gian mới với số chiều ít hơn. Quá trình này không chỉ giúp tăng tốc độ xử lý mà còn hỗ trợ việc trực quan hóa kết quả gom cụm sau này.

Bước cuối cùng là gom cụm các vector đã giảm chiều để tìm ra các nhóm thực thể tương đồng và phát hiện kiểu thực thể. Nhóm đã sử dụng HDBSCAN, một thuật toán gom cụm dựa trên mật độ, có khả năng tự động phát hiện số lượng cụm mà không cần quy định trước, đồng thời loại bỏ những thực thể không có giá trị ngữ nghĩa (điểm nhiễu). Kết quả của quá trình này là các cụm chứa các thực thể thuộc cùng một kiểu, giúp định danh các thực thể đã trích xuất.

Tuy nhiên, điều này không có nghĩa là các cụm được phát hiện luôn hoàn hảo. Trên thực tế, các cụm sau khi được tạo ra vẫn có thể nằm gần nhau trong không gian vector và thậm chí trùng lặp về kiểu thực thể. Chính vì vậy, sau khi thực hiện gom cụm, chúng ta tiến hành thêm một bước chuẩn hóa nhằm đảm bảo các cụm tương đồng về ngữ nghĩa được gộp lại với nhau và loại bỏ sự trùng lặp giữa các kiểu thực thể.

4.1.2.c Chuẩn hoá

Sau khi đã hoàn thành quá trình gom cụm các vector thực thể, bước tiếp theo là tiến hành chuẩn hóa kết quả để đảm bảo tính chính xác và nhất quán của các cụm. Việc chuẩn hóa bao gồm 2 bước chính: đánh giá chất lượng cụm, định nghĩa và hợp nhất các cụm tương đồng. Quá trình này giúp tối ưu hóa số lượng kiểu thực thể thực sự, đồng thời đảm bảo tính toàn vẹn của kết quả.

Silhouette Score là một chỉ số phổ biến để đánh giá chất lượng của các cụm trong bài toán gom cụm. Điểm số này thể hiện mức độ phân tách giữa các cụm và sự đồng nhất trong mỗi cụm. Cụ thể, Silhouette Score đánh giá xem một thực thể nằm đúng trong cụm của nó hay không, và khoảng cách giữa các thực thể trong cùng cụm so với các cụm khác có đủ lớn hay không.

Trong bước này, nhóm sử dụng Silhouette Score để lựa chọn và tinh chỉnh

tham số của các thuật toán gom cụm. Nếu Silhouette Score của các cụm quá thấp, điều đó có nghĩa là các thực thể không được phân tách rõ ràng và có thể bị trộn lẫn giữa các cụm khác nhau. Khi đó, ta sẽ điều chỉnh các tham số như số chiều sau khi giảm hoặc kích thước cụm tối thiểu để cải thiện chất lượng gom cụm.

Mặc dù Silhouette Score là một chỉ số hữu ích, nó không hoàn toàn đảm bảo rằng các cụm được phân tách đúng với kiểu thực thể thực tế. Do đó, bằng cách quan sát trực tiếp và kiểm tra các cụm, ta có định nghĩa kiểu thực thể cho mỗi cụm cũng như phát hiện những cụm có ngữ nghĩa tương tự và tiến hành hợp nhất chúng thành một cụm duy nhất nếu chúng cùng đại diện cho một kiểu thực thể. Quá trình này không chỉ giúp làm giảm số lượng cụm không cần thiết mà còn tăng cường độ chính xác trong việc phát hiện kiểu thực thể thực sự.

Bằng cách hoàn thiện 2 bước trên, quy trình chuẩn hóa sau gom cụm không chỉ giúp cải thiện chất lượng gom cụm mà còn đảm bảo tính chính xác trong việc xác định kiểu thực thể, từ đó mang lại kết quả rõ ràng và có giá trị thực tiễn.

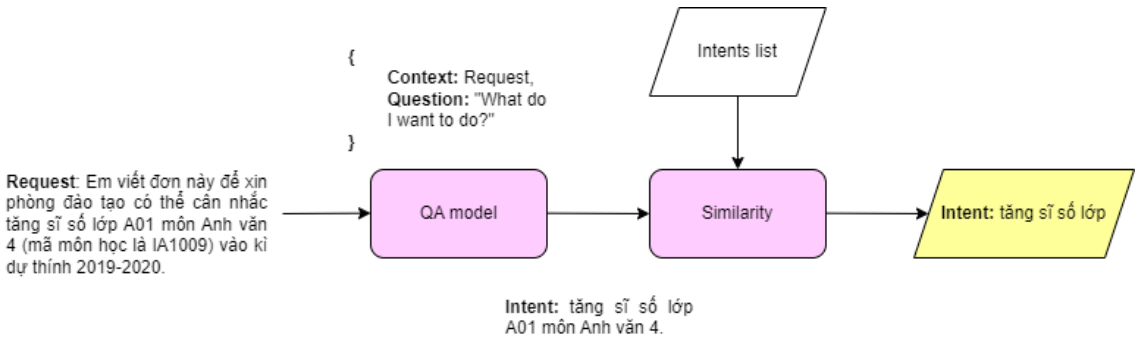
4.1.3 Phân loại ý định

Phân loại ý định từ người dùng là một trong những nhiệm vụ khó khăn bởi vì trong thực tế, các ý định rất đa dạng và phức tạp, nhất là trong lĩnh vực giáo dục khi một yêu cầu, thắc mắc đến từ sinh viên có thể chứa nhiều hơn một ý định. Trong phạm vi của bài toán này, nhóm sẽ xem như mỗi yêu cầu đến từ sinh viên mang một ý định chính và tập trung rút trích nó để tìm ra mối quan hệ giữa ý định và các kiểu thực thể có liên quan. Trích xuất nhiều ý định có thể xem là một phần mở rộng và cần được nghiên cứu thêm ở một đề tài khác.

Lấy ý tưởng từ hướng tiếp cận rút trích sự kiện bằng phương pháp đọc hiểu máy [13] cùng với đó là sự pháp triển vượt bậc của các mô hình tiền huấn luyện hiện nay, sự thừa hưởng từ nghiên cứu về khám phá ý định [4] với một danh sách các ý định có trong miền giáo dục, nhóm đã đề xuất một phương pháp tìm ra ý định người dùng bằng cách kết hợp giữa đọc hiểu máy và phép đo sự tương đồng cosine similarity. Chu trình được thể hiện qua các bước cơ bản như sau (hình 23):

- Các yêu cầu, thắc mắc từ người dùng sẽ được đưa qua một mô hình trả lời câu hỏi để trích ra ý định từ văn bản đầu vào.
- Kế đến, các ý định này sẽ được vector hoá và sử dụng phép đo cosine similarity để phân loại chính xác ý định này vào một danh sách ý định có sẵn, từ đó phân loại được ý định người dùng theo một danh sách chuẩn.

Với đầu vào của mô hình đọc hiểu là một cặp giá trị (câu hỏi, ngữ cảnh), nhóm thiết kế một câu hỏi đơn giản nhưng có ý nghĩa, giúp mô hình tập trung vào ý



Hình 23: Pipeline dùng cho phân loại ý định từ dữ liệu

định của người dùng. Câu hỏi được đặt dưới dạng: "What do I want to do?" (Tôi muốn làm gì?), nhằm dẫn dắt mô hình đọc hiểu máy tìm kiếm câu trả lời trực tiếp liên quan đến hành động hoặc mong muốn của người dùng từ văn bản. Phần ngữ cảnh chính là văn bản đầu vào từ người dùng.

Ví dụ, nếu người dùng nhập vào: “Tôi muốn đăng ký môn học cho học kỳ tới”, mô hình sẽ phân tích câu này và đưa ra kết quả là ý định của người dùng có thể là "đăng ký môn học". Ưu điểm của cách tiếp cận này là mô hình MRC có khả năng phân tích toàn bộ văn bản và tìm ra những phần liên quan trực tiếp đến ý định của người dùng.

Sau khi trích xuất được ý định thô từ mô hình MRC, bước tiếp theo là xác định ý định này tương đồng với ý định nào nhất trong danh sách các ý định đã được định nghĩa trước đó. Để làm điều này, nhóm sử dụng kỹ thuật embedding để chuyển đổi các câu văn thành các vector trong không gian nhiều chiều. Cả câu trả lời được trích xuất và danh sách ý định đều được nhúng thành vector, sau đó áp dụng phép đo cosine similarity, một phương pháp đo lường độ tương đồng giữa hai vector dựa trên góc giữa chúng. Khi giá trị cosine similarity càng cao (gần bằng 1), điều đó có nghĩa là hai vector càng gần nhau, tức là ý định mà người dùng đưa ra có sự tương đồng ngữ nghĩa cao với một ý định trong danh sách.

Dù trên thực tế, phương pháp này có thể không mang lại hiệu quả cao trong mọi trường hợp, đặc biệt khi xử lý các câu phức tạp hoặc có ngữ cảnh đặc biệt, ưu điểm lớn nhất của nó là khả năng tự động hóa quá trình tìm kiếm và chuẩn hoá ý định. Nhờ tính tự động này, hệ thống có thể nhanh chóng đối chiếu ý định của người dùng với danh sách có sẵn mà không đòi hỏi quá nhiều sự can thiệp thủ công, giúp tiết kiệm thời gian và công sức trong các tác vụ thông thường.

4.1.4 Áp dụng luật kết hợp

Trong bài toán khám phá thực thể dựa trên ý định này, nhóm đã áp dụng phương pháp luật kết hợp (association rule mining) để khám phá mối quan hệ giữa ý định của người dùng và các kiểu thực thể liên quan trong miền dữ liệu giáo dục. Phương pháp này không chỉ giúp xác định những kiểu thực thể nào thường xuất hiện cùng với một ý định cụ thể, mà còn thể hiện rõ mối quan hệ tiềm ẩn giữa các yếu tố này, từ đó cung cấp cái nhìn sâu sắc về các nhu cầu của sinh viên.

Khi sinh viên đưa ra một câu hỏi hoặc yêu cầu, họ thường có những ý định cụ thể, chẳng hạn như đăng ký môn học, xin thi bù, hoặc thay đổi lịch học. Tuy nhiên, để đáp ứng những ý định này, thường sẽ có các thông tin liên quan, ví dụ: kỳ thi, môn học, hoặc thời gian. Việc nắm bắt mối quan hệ giữa ý định và các kiểu thực thể liên quan không chỉ giúp hiểu rõ các mối quan tâm của sinh viên, mà còn cho phép xây dựng tri thức chuyên sâu về các tình huống giáo dục.

Sau khi trích xuất ý định và kiểu thực thể từ dữ liệu, nhóm sẽ tiến hành tổng hợp hai loại thông tin này thành các tập hợp dữ liệu có cấu trúc rõ ràng. Mỗi mẫu dữ liệu sẽ bao gồm một ý định từ người dùng và các kiểu thực thể liên quan đã được phát hiện. Ví dụ, một ý định như "đăng ký môn học" có thể đi kèm với các kiểu thực thể như "tên môn học", "học kì", và "thời gian đăng ký".

Kế đến ta sẽ áp dụng luật kết hợp để xác định xem những kiểu thực thể nào có thể xuất hiện cùng với ý định đó, và mức độ phổ biến của các kiểu thực thể này. Ví dụ, trong trường hợp sinh viên có ý định "đăng ký môn học", kiểu thực thể "tên môn học" thường xuất hiện với tần suất cao, và kiểu thực thể "học kì" hoặc "thời gian đăng ký" xuất hiện với tần suất ít hơn. Một số luật có thể được khai phá như sau:

- Luật 1: Nếu ý định là "đăng ký môn học" thì sẽ liên quan đến "tên môn học".
- Luật 2: Nếu ý định là "đăng ký môn học" thì sẽ liên quan đến "tên môn học" và "học kì".

Mối quan hệ giữa ý định và thực thể được khám phá qua luật kết hợp không chỉ là một bước tiến trong việc hiểu rõ hơn về tri thức ngữ cảnh trong lĩnh vực giáo dục, mà còn giúp bộ phận hỗ trợ trả lời sinh viên tốt hơn khi có thể dự đoán và chuẩn bị trước những thông tin cần thiết khi sinh viên đưa ra một yêu cầu cụ thể, từ đó cải thiện trải nghiệm người dùng. Hơn nữa, tri thức này còn có thể được sử dụng để xây dựng một đồ thị tri thức trong lĩnh vực giáo dục và các hệ thống tư vấn học tập, giúp sinh viên dễ dàng tìm được câu trả lời cho những thắc mắc về quá trình học tập.

4.1.5 Xây dựng đồ thị tri thức

Sau khi có tập ánh xạ giữa ý định và các kiểu thực thể. Nhóm sẽ dựa trên đó để thiết kế mô hình đồ thị tri thức mà nhóm xây dựng. Nhóm sẽ dùng mô hình Hyper Relational Knowledge Graph cho đồ thị tri thức trong đề tài. Mô hình này bổ sung các bộ ba (triples) trong đồ thị tri thức các cặp *khóa: giá trị* để cung cấp thông tin cho các bộ ba khi mà bản thân chúng là chưa đủ để thể hiện một sự thật (fact).

Nhóm sẽ xem các ý định như là các mối quan hệ (relation) trong đồ thị này. Ý định sẽ là mối quan hệ giữa sinh viên (người gửi yêu cầu/câu hỏi) và một thực thể chính mà nó thông tin quan trọng nhất trong các thực thể liên quan khác. Các thực thể còn lại sẽ được biểu diễn dưới dạng *khóa: giá trị* với "khóa" là kiểu thực thể và "giá trị" là đối tượng tương ứng với kiểu thực thể đó. Vì thế, với mỗi ý định, một kiểu thực thể sẽ được chọn ra thành kiểu thực thể chính trong main triple. Kiểu thực thể nào cung cấp nhiều thông tin ngữ nghĩa nhất cho ý định đó sẽ được nhóm chọn ra làm kiểu thực thể chính. Các ý định chỉ có một kiểu thực thể liên kết thì sẽ xem kiểu đó như kiểu thực thể chính.

Mô hình toán học cho đồ thị tri thức tập trung vào ý định này như sau: $G = (E, I, T, Et, f)$ với V là tập tất cả các thực thể, I là tập các ý định, Et là tập các kiểu thực thể. $E = (e_1, e_2, ..., e_n)$ là tập các 'fact' hay 'hyper triple' trong đồ thị. Một fact e_i sẽ có dạng $(s, i, e*)(et : e)|et \in f(i)$. Trong đó, $s \in E$ là thực thể thuộc kiểu "sinh viên" (trong tập Et), i là một ý định thuộc tập ý định I và $e* \in E$ là một thực thể chính trong fact. Một fact gồm main triple và nhiều cặp *khóa: giá trị* với các $et \in Et$ là các kiểu thực thể liên kết với ý định i ($s(i)$ là phép ánh xạ lấy ra các kiểu thực thể liên kết với ý định i) và e là thực thể tương ứng thuộc kiểu đó.

Ví dụ một fact được trích được từ câu yêu cầu đến của bạn Nguyễn Văn A như bảng sau:

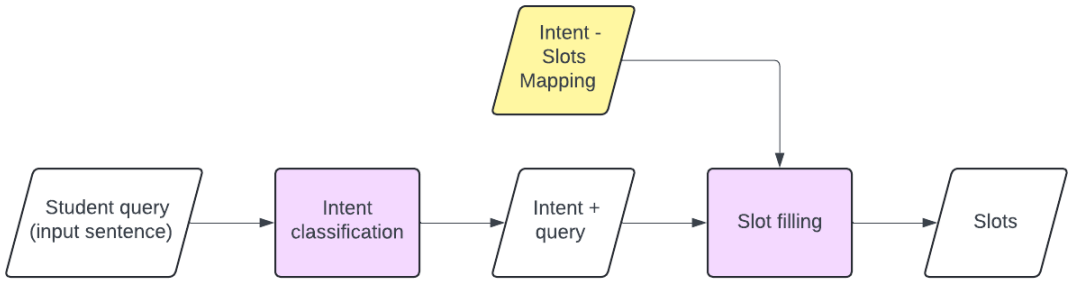
Yêu cầu	Em muốn đăng ký môn học Công nghệ phần mềm ở học kì 241 này ạ. Mong PĐT hỗ trợ giúp em
Fact	(Nguyễn Văn A, đăng kí môn học, Công nghệ phần mềm)[học kì: 241]

Bảng 4: Ví dụ về một fact đồ thị tri thức được rút trích

4.2 Phát hiện ý định và điền trường thông tin

Với phần phát hiện ý định và điền trường thông tin, nhóm sẽ xử lý hai bài toán này tách biệt nhau. Với phương pháp đọc hiểu máy, nhiệm vụ điền trường

thông tin cần dùng đến kết quả của nhiệm vụ phân loại ý định vì nhiều lợi ích nên nhóm xây dựng một framework với cơ chế pipeline với nhiệm vụ phát hiện ý định sẽ thực hiện trước và sau đó là nhiệm vụ "điền trường thông tin. Kiến trúc của framework được mô tả tóm gọn như hình 24.



Hình 24: Framework cho nhiệm vụ Phát hiện ý định và Điền trường thông tin

Đầu vào của quá trình là câu yêu cầu, câu hỏi từ sinh viên lấy từ hệ thống hỗ trợ sinh viên, gọi chung là truy vấn. Sau khi qua quá trình phân loại ý định, ta sẽ có nhãn ý định cho câu truy vấn. Cả ý định, câu truy vấn và một sơ đồ ánh xạ "ý định - trường" sẽ được sử dụng trong quá trình điền trường thông tin. Quá trình này cho ra các giá trị trường có trong câu truy vấn. Kết quả đầu ra của toàn bộ framework mà ta cần sẽ là ý định và các trường thông tin của câu đầu vào.

4.2.1 Phân loại ý định

Với phần phân loại ý định dùng cho framework này, nhóm sử dụng cùng một phương pháp với phần phân loại ý định dùng trong phần xây dựng đồ thị tri thức. Phần nội dung phương pháp, quý độc giả có thể tham khảo ở mục trên, nhóm xin phép không trình bày lại.

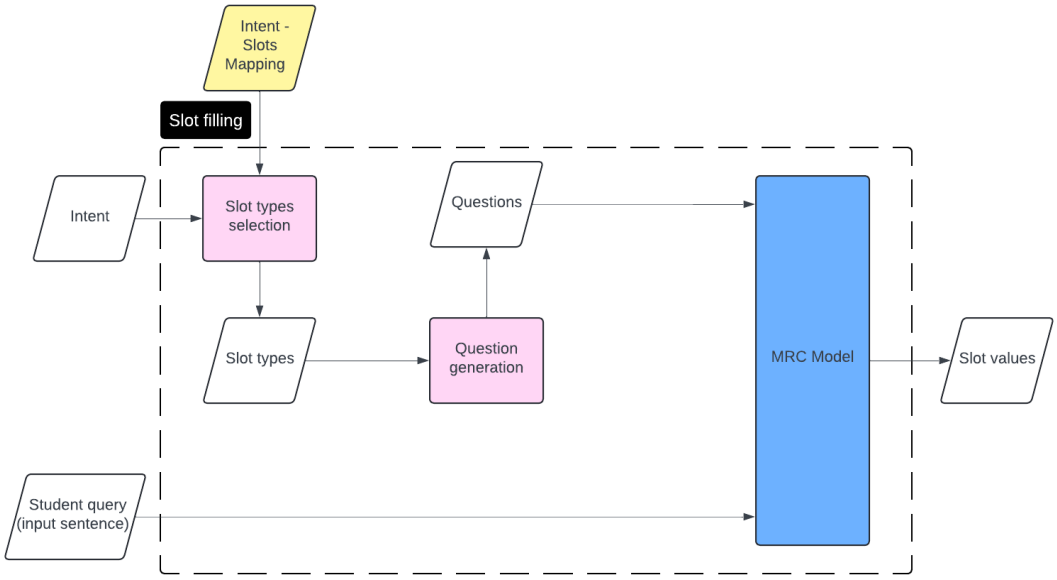
4.2.2 Điền trường thông tin

Vì các ưu điểm đã phân tích của phương pháp trong bối cảnh đề tài, nhóm sẽ chọn phương pháp Đọc hiểu máy để thử nghiệm cho nhiệm vụ Điền trường thông tin vì các ưu điểm của nó. Sơ đồ kiến trúc framework được cho ở hình 25

Các câu hỏi ảo dùng cho mô hình Đọc hiểu máy được sinh ra theo luật (rule-based). Nhóm đề xuất ba mẫu câu hỏi dùng để thử nghiệm trong đề tài này.

Mẫu 1: **kiểu thực thể + "?"**

Ở mẫu này, nhóm sẽ chỉ sử dụng các tên kiểu thực thể (hay kiểu trường) mà đã lấy ra được khi biết ý định trong câu ngữ cảnh dựa trên một sơ đồ "Intent-Slots



Hình 25: Framework cho nhiệm vụ Điền trường thông tin

Mapping" (hình 25, dùng làm câu hỏi cho mô hình đọc hiểu máy.

Mẫu 2: **kiểu thực thể** + **"là gì"** + **"?"**
Ở mẫu này, nhóm sẽ kết hợp các tên kiểu thực thể và cụm từ "là gì" cùng dấu "?" để làm câu hỏi cho mô hình đọc hiểu máy.

Mẫu 3: **Sinh viên muốn** + **ý định** + **"với"** + **kiểu thực thể** + **"là gì"** + **"?"**
Ở mẫu này, các câu hỏi được sinh ra dựa trên sự kết hợp giữa ý định và các kiểu thực thể liên quan đến nó. Nhóm sử dụng nhiều cụm từ kết hợp để tạo nên câu hỏi, làm câu hỏi trở nên tự nhiên hơn. Lấy ví dụ một câu hỏi: "Sinh viên muốn đăng kí môn học với môn học gì ?".

5 Thí nghiệm

5.1 Rút trích các thực thể có tên

5.1.1 Dữ liệu và thiết lập thí nghiệm

Tập dữ liệu mà nhóm dùng để huấn luyện và đánh giá mô hình là các tập dữ liệu văn bản được lấy từ các nguồn của trường Đại học Bách Khoa, trong đó:

- **RawData_From_BKSite**: Tập dữ liệu bao gồm các thông tin từ các website chính thức của nhà trường, các thông tin về khoa, ngành, các quy chế quy định và các trang thông tin môn học.
- **FAQs**: Tập dữ liệu chứa hơn 200,000 câu hỏi/yêu cầu đến từ sinh viên được trích từ hệ thống BKSI của nhà trường. Đây là tập dữ liệu chính dùng để áp dụng trong framework xây dựng đồ thị tri thức, trong đó dữ liệu đánh nhãn chiếm 1% và dữ liệu dùng để khám phá thực thể chiếm 10% trên tổng tập dữ liệu.

Mô hình nhóm sử dụng cho nhiệm vụ nhận diện thực thể có tên (NER) là một biến thể của mô hình ELECTRA [6] được fine-tune cho nhiệm vụ NER trên tập dữ liệu tiếng Việt VLSP 2018. Mô hình được công khai trên nền tảng Hugging Face. Thông tin chi tiết về mô hình và các đánh giá sơ bộ có thể xem ở đường link này: https://huggingface.co/NlpHUST/ner-vietnamese-electra-base.

Bộ siêu tham số dùng để fine-tune mô hình:

- learning rate: 5e-5
- training batch size: 8
- optimizer: Adam với betas=(0.9, 0.999) và epsilon=1e-8
- learning rate scheduler type: linear
- number of epochs: 3

5.1.2 Kết quả và thảo luận

Kết quả khi chạy đánh giá mô hình khi chưa huấn luyện trên tập dữ liệu giáo dục ở bảng 5. Ta thấy hiệu quả dự đoán không quá cao, đặc biệt là nhãn MISC

Kết quả khi chạy đánh giá mô hình khi đã được fine-tune trên tập dữ liệu giáo dục ở bảng 6. Các chỉ số đều tăng lên khá nhiều, đặc biệt là nhãn MISC.

Label	Số lượng mẫu	Precision	Recall	F1
PER	2	0,00%	0,00%	0,00%
ORG	45	33,33%	53,33%	41,03%
LOC	6	42,86%	50,00%	46,15%
MISC	229	20,00%	0,44%	0,85%
Tổng	282	30,23%	9,22%	14,13%

Bảng 5: Kết quả đánh giá mô hình chưa fine-tune

Label	Số lượng mẫu	Precision	Recall	F1
PER	2	100,00%	50,00%	66,66%
ORG	45	37,68%	57,77%	45,61%
LOC	6	42,86%	50,00%	46,15%
MISC	229	68,96%	69,87%	69,41%
Tổng	282	61,49%	67,38%	64,29%

Bảng 6: Kết quả đánh giá mô hình đã fine-tune

Mô hình sau quá trình huấn luyện sẽ được áp dụng để nhận diện các thực thể có trong tập dữ liệu FAQs. Tổng số thực thể tìm được lên đến hơn 22000 thực thể. Đây cũng là số lượng thực thể đầu vào sử dụng cho bước gom cụm kế tiếp.

5.2 Phân cụm thực thể

Trong quá trình gom cụm, nhóm đã thử nghiệm hai mô hình embed tiên tiến hiện nay là simCSE và SBERT để biểu diễn các thực thể thành vector trong không gian nhiều chiều. Bên cạnh đó, UMAP cũng được sử dụng để đưa các vector này về không gian ít chiều hơn, giảm độ phức tạp khi tính toán tránh hiện tượng các vector thưa thớt. Kết quả của thực nghiệm được trình bày ở bảng 7 khi áp dụng thuật toán gom cụm HDBSCAN.

Dựa vào kết qủa ở bảng 7, ta có thể thấy điểm Silhouette của hai thí nghiệm **8** và **9** (min cluster size = 20) không chênh lệch quá nhiều so với thí nghiệm **10** (min cluster size = 10). Tuy nhiên kết quả số lượng cụm trả về của thí nghiệm **10** khá lớn (gần như gấp đôi), điều đó đòi hỏi chúng ta tốn nhiều công sức hơn để quan sát và định nghĩa các kiểu thực thể số các cụm này. Vì thế, nhóm sẽ chọn hai bộ thí nghiệm **8** và **9** để tiến hành phân tích các bước tiếp theo. Bằng cách kiểm tra trực tiếp và gán kiểu thực thể cho cụm dựa trên quan sát các thực thể có trong mỗi cụm, nhóm đã tìm được tổng cộng **23** kiểu thực thể được khai phá từ kết quả gom cụm của thí nghiệm **8**, con số này là **22** tương ứng với thí nghiệm

No	Embedding Model	Dimensions	Min cluster size	Num of Clusters	Silhouette Score
1	simCSE	50	30	66	0.599
2	simCSE	9	60	26	0.601
3	SBERT	16	10	231	0.617
4	simCSE	16	50	36	0.618
5	simCSE	9	50	36	0.622
6	simCSE	60	10	36	0.628
7	SBERT	9	20	117	0.636
8	simCSE	9	20	122	**0.647**
9	SBERT	16	20	112	**0.649**
10	simCSE	9	10	241	0.65

Bảng 7: Bộ thông số dùng cho quá trình phân cụm và kết quả tương ứng

9. Từ đó ta có thể chọn kết quả của thí nghiệm **8** là kết quả gom cụm cuối cùng.

Như vậy thông qua quá trình thu gọn và chuẩn hoá, nhóm đã khám phá ra được **23** loại thực thể có liên quan trong miền dữ liệu giáo dục, kèm theo đó là **5** loại thực thể được tìm ra thêm trong quá trình đánh nhãn dữ liệu cho quá trình đánh giá. Trong thực tế, 5 loại thực thể này tuy vẫn tồn tại trong tập dữ liệu, tuy nhiên có thể do chưa xuất hiện đủ nhiều hoặc chưa rõ ngữ nghĩa khiến mô hình nhúng và thuật toán gom cụm hoạt động chưa tốt, dẫn đến việc phân sai cụm. Đây cũng có thể xem là một thiếu sót của phương pháp này khi chỉ tìm ra những kiểu thực thể nổi trội thay vì tất cả các kiểu thực thể.

Các kiểu thực thể sau khi được định nghĩa và chuẩn hoá sẽ được gán ngược lại cho từng thực thể để sử dụng cho bước khai phá luật kết hợp. Độ chính xác của quá trình tự động gán ngược kiểu thực thể đạt **52.3%** với tổng số **655** kiểu thực thể được gán đúng trên tổng số **1252** nhãn.

Bảng tổng hợp các kiểu thực thể được trình bày ở **phụ lục B**.

5.3 Phân loại ý định

5.3.1 Mô hình

Mô hình nhóm sử dụng cho nhiệm vụ trả lời câu hỏi là một biến thể của mô hình XLM-RoBERTa [7] được tinh chỉnh lại cho nhiệm vụ đọc hiểu máy (phiên bản *large*).

Thông tin chi tiết mô hình có thể xem tại https://huggingface.co/nguyenvulebinh/vi-mrc-large và phụ lục A.

5.3.2 Kết quả

Trong quá trình phân loại ý định bằng phương pháp đọc hiểu máy kết hợp với đo lường độ tương đồng ngữ nghĩa (cosine similarity), hệ thống chỉ đạt được độ chính xác 16.2% trên tổng số 302 nhãn ý định. Kết quả này thấp hơn nhiều so với kỳ vọng ban đầu và phản ánh những hạn chế của phương pháp trong bối cảnh bài toán cụ thể.

Một trong những nguyên nhân khiến phương pháp này chưa hoạt động tốt là do mô hình đọc hiểu thường trích một câu trả lời đầy đủ thay vì chỉ trích những động từ thể hiện ý định. Ví dụ, với một yêu cầu đầu vào "Em muốn đăng kí môn học luận văn tốt nghiệp để kịp tốt nghiệp trong năm tới.", mô hình sẽ trích cả cụm "đăng kí môn học luận văn tốt nghiệp để kịp tốt nghiệp trong năm tới." thay vì chỉ là "đăng kí môn học". Chính vì những thông tin không cần thiết đi kèm phía sau làm cho kết quả phép đo similarity chưa cao, dẫn tới phân loại sai ý định.

Vì lẽ đó, để phục vụ cho bài toán khám phá mối quan hệ giữa ý định và kiểu thực thể bằng association rule, nhóm sẽ sử dụng kết quả gán nhãn thủ công ý định thay vì phương pháp đã đề ra để đảm bảo những luật tạo thành sẽ chính xác và có ý nghĩa.

5.4 Khai phá luật

Để đánh giá hiệu quả của phương pháp khai phá luật kết hợp, nhóm đã tiến hành hai thí nghiệm trên tập dữ liệu với cách tiếp cận khác nhau nhằm so sánh chất lượng và số lượng luật được khám phá. Mục tiêu của thí nghiệm là kiểm chứng xem việc tự động gán nhãn kiểu thực thể bằng phương pháp gom cụm có thể đạt được kết quả tương đương hoặc gần đúng so với gán nhãn thủ công hay không.

Thí nghiệm 1: trong thí nghiệm đầu tiên, nhóm sử dụng kết quả gán nhãn thủ công cho cả ý định và kiểu thực thể liên quan đến ý định. Để khai phá luật kết hợp, chúng tôi sử dụng các giá trị minimum support (min sup) và minimum confidence (min conf) đều bằng 0,5. Sau khi áp dụng khai phá luật kết hợp, có 51 luật kết hợp được khai phá từ tập dữ liệu này. Kết quả thu gọn các luật được trình bày tại **phụ lục B**.

Thí nghiệm 2: trong thí nghiệm thứ hai, ta vẫn giữ kết quả gán nhãn thủ công cho ý định, nhưng kiểu thực thể được gán nhãn tự động dựa trên kết quả của phương pháp gom cụm. Kết quả thu được từ phương pháp này cho thấy có 30 luật được khai phá với cùng giá trị min sup và min conf bằng 0,5. Trong số đó, 23 luật là đúng so với các luật trong tập 51 luật từ thí nghiệm gán nhãn thủ công. Điều này có nghĩa là phương pháp tự động gán kiểu thực thể thông qua gom cụm đã phát hiện ra 23 luật giống với kết quả khai phá thủ công, chiếm 76,67% tính chính xác so với tập luật chuẩn.

Kết quả trên đã cho thấy rằng phương pháp tự động gán kiểu thực thể dựa vào kết quả gom cụm có thể hỗ trợ hiệu quả quá trình khai phá luật kết hợp. Mặc dù số lượng và chất lượng luật không cao bằng phương pháp gán nhãn thủ công, nhưng đây là một giải pháp tiềm năng để tự động hóa việc phân loại kiểu thực thể, đặc biệt trong các hệ thống có quy mô lớn hoặc khi không thể dựa vào chuyên gia để gán nhãn.

5.5 Điền trường thông tin

5.5.1 Dữ liệu và thiết lập thí nghiệm

Dữ liệu được gán nhãn ý định sẽ được sử dụng trong nhiệm vụ Điền trường thông tin. Mỗi điểm dữ liệu đều có nhãn là các trường có thể được rút trích dùng để huấn luyện và đánh giá mô hình.

Mô hình nhóm sử dụng cho thí nghiệm này cũng là mô hình đã áp dụng cho nhiệm vụ phân loại ý định.

Nhóm sẽ thí nghiệm với ba mẫu câu hỏi và so sánh kết quả dự đoán của mô hình trên ba mẫu câu hỏi này.

5.5.2 Kết quả và thảo luận

Kết quả đạt được như Bảng 8:

Các chỉ số độ đo khi chạy mô hình với mẫu câu hỏi 2 có kết quả nhỉnh hơn kết quả khi chạy với mẫu câu hỏi 1. Đáng ngạc nhiên, kết quả khi chạy với câu hỏi 3 lại thấp hơn so với hai mẫu trước.

Sau khi kiểm tra, nhóm nhận thấy số dự đoán đúng khi chạy với mẫu 2 so với khi chạy với mẫu 1 thì chỉ hơn một dự đoán. Với kết quả này, ta chưa thể kết luận được việc sử dụng mẫu câu hỏi 2 là tốt hơn hẳn so với khi sử dụng dụng mẫu câu

Mẫu câu hỏi	P	R	F1
Mẫu 1	24,03	1,67	3,13
Mẫu 2	**24,81**	**1,73**	**3,23**
Mẫu 3	19,69	1,35	2,53

Bảng 8: Kết quả đánh giá mô hình trên các mẫu câu hỏi

hỏi 1 hay không. Với mẫu câu hỏi 3, dường như việc kết hợp các cụm từ để hỏi cùng với ý định và kiểu thực thể gây ra sự mất tự nhiên trong diễn đạt câu hỏi, gây khó hiểu hơn cho mô hình Đọc hiểu máy.

Với cả ba mẫu câu hỏi, độ phủ (Recall) đều ở mức rất thấp so với độ chính xác (Precision), nguyên nhân là có trường hợp có nhiều thực thể có cùng một kiểu thực thể trong câu đầu vào. Tuy nhiên, mô hình Đọc hiểu máy chỉ trích được một span duy nhất cho mỗi kiểu thực thể được dùng để làm câu hỏi. Vì thế, số lượng dự đoán là nhỏ hơn nhiều so với số lượng thực thể thực tế có thể trích. Để khắc phục điểm yếu này, nhóm đề xuất nên chọn một mô hình có khả năng trích được nhiều span cho một câu hỏi để lượng dự đoán có thể tăng lên.

6 Tổng kết

Những đóng góp trong bài nghiên cứu này bao gồm:

- Nhóm phương pháp đầu tiên cho hướng tiếp cận khai phá kiểu thực thể trong nhiệm vụ xây dựng đồ thị tri thức. Trong đó, nhóm sử dụng kĩ thuật học giám sát để trích các thực thể trên miền chung (general domain) dùng làm đầu vào cho thuật toán phân cụm để cho ra kiểu thực thể.
- Nhóm đề xuất mô hình (lược đồ) mới cho đồ thị tri thức chứa thông tin về ý định. Nhóm dùng thuật toán khai phá luật kết hợp để tìm ra các liên kết giữa kiểu thực thể và ý định để thiết kế mô hình này.
- Nhóm sử dụng lược đồ được thiết kế để áp dụng trong nhiệm vụ Phát hiện ý định và Điền trường thông tin. Phương pháp Đọc hiểu máy tiên tiến được thử nghiệm cho bài toán.

Vì nhóm thử nghiệm nhiều phương pháp, hướng tiếp cận mới nên kết quả chưa được khả quan. Nhóm đề xuất một số hướng phát triển đề tài trong tương lai dựa trên kết quả thử nghiệm của nhóm như sau:

- Sử dụng phương pháp học không giám sát để trích xuất tự động các thực thể có trong bộ dữ liệu, giảm thiểu hoặc tránh việc chuẩn bị dữ liệu gắn nhãn.
- Sử dụng chiến lược nhúng dựa trên ngữ cảnh để biểu diễn tốt hơn các thực thể được rút trích, từ đó tối ưu kết quả phân cụm.
- Thực hiện các bước tinh chỉnh để đồ thị hoàn thiện hơn.
- Phát triển chiến lược đặt câu hỏi hợp lý để cải thiện đầu ra của hai nhiệm vụ Phát hiện ý định và Điền trường thông tin.

Phân công nhiệm vụ

Nhiệm vụ	**Sinh viên thực hiện**	Mức độ đóng góp
Tìm hiểu đề tài	Nghiêm, Khoẻ	100%
Trích xuất thực thể	Khoẻ	100%
Phân cụm ngữ nghĩa	Nghiêm	100%
Khai phá luật kết hợp	Nghiêm	100%
Phân loại ý định	Nghiêm	100%
Điền trường thông tin	Khoẻ	100%

Tài liệu tham khảo

[1] Bilal Abu-Salih. Domain-specific knowledge graphs: A survey. *Journal of Network and Computer Applications*, 185:103076, 2021.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, page 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: a nucleus for a web of open data. ISWC'07/ASWC'07, page 722–735, Berlin, Heidelberg, 2007. Springer-Verlag.

[4] Tuan Bui, Oanh Tran, Phuong Nguyen, Bao Ho, Long Nguyen, Thang Bui, and Tho Quan. Cross-data knowledge graph construction for llm-enabled educational question-answering system: A case study at hcmut. In *Proceedings of the 1st ACM Workshop on AI-Powered Qamp;A Systems for Multimedia*, ICMR '24, page 36–43. ACM, June 2024.

[5] Penghe Chen, Yu Lu, Vincent W. Zheng, Xiyang Chen, and Boda Yang. Knowedu: A system to construct knowledge graph for education. *IEEE Access*, 6:31553–31563, 2018.

[6] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020.

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics.

[8] Alexis Conneau and Guillaume Lample. *Cross-lingual language model pretraining*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[9] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces, 2018.

[10] Mai Hoang Dao, Thinh Hung Truong, and Dat Quoc Nguyen. Intent detection and slot filling for vietnamese, 2021.

[11] Dingsheng Deng. Dbscan clustering algorithm based on density. In *2020 7th International Forum on Electrical Engineering and Automation (IFEEA)*, pages 949–953, 2020.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[13] Xinya Du and Claire Cardie. Event extraction by answering (almost) natural questions. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online, November 2020. Association for Computational Linguistics.

[14] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In Regina Barzilay and Mark Johnson, editors, *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[15] Yousra Fettach, Mounir Ghogho, and Boualem Benatallah. Knowledge graphs in education and employability: A survey on applications and techniques. *IEEE Access*, 10:80174–80183, 2022.

[16] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[17] Charles T. Hemphill, John J. Godfrey, and George R. Doddington. The atis spoken language systems pilot corpus. In *Proceedings of the Workshop on Speech and Natural Language*, HLT '90, page 96–101, USA, 1990. Association for Computational Linguistics.

[18] Nicolas Hubert, Armelle Brun, and Davy Monticolo. New ontology and knowledge graph for university curriculum recommendation. 10 2022.

[19] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514, 2022.

[20] Nguyen Kiet, Tran Son, Nguyen Luan, Huynh Tin, Luu Son, and Nguyen Ngan. Vlsp 2021-vimrc challenge: Vietnamese machine reading comprehension. *VNU Journal of Science: Computer Science and Communication Engineering*, 38(2), 2022.

[21] John D. Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, 2001.

[22] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In Roger Levy and Lucia Specia, editors, *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[23] Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. Event extraction as multi-turn question answering. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838, Online, November 2020. Association for Computational Linguistics.

[24] Fengru Li, Wei Xie, Xiaowei Wang, and Zhe Fan. Research on optimization of knowledge graph construction flow chart. In *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, volume 9, pages 1386–1390, 2020.

[25] Xiang Li, Chuanqi Wei, Zhihong Jiang, et al. Eduner: a chinese named entity recognition dataset for education research. *Neural Computing and Applications*, 35:17717–17731, 2023.

[26] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. A unified MRC framework for named entity recognition. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online, July 2020. Association for Computational Linguistics.

[27] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. Entity-relation extraction as multi-turn question answering. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings*

of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1340–1350, Florence, Italy, July 2019. Association for Computational Linguistics.

[28] Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. Event extraction as machine reading comprehension. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online, November 2020. Association for Computational Linguistics.

[29] Pengfei Liu, Youzhang Ning, King Keung Wu, Kun Li, and Helen M. Meng. Open intent discovery through unsupervised semantic clustering and dependency parsing. *ArXiv*, abs/2104.12114, 2021.

[30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019.

[31] Di Lu, Shihao Ran, Joel Tetreault, and Alejandro Jaimes. Event extraction as question generation and answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1666–1688, Toronto, Canada, July 2023. Association for Computational Linguistics.

[32] J. MacQueen. Some methods for classification and analysis of multivariate observations. 1967.

[33] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 52:99–115, 1990.

[34] Leland McInnes, John Healy, and S. Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2:205, 2017.

[35] Kiet Nguyen, Vu Nguyen, Anh Nguyen, and Ngan Nguyen. A Vietnamese dataset for evaluating machine reading comprehension. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2595–2605, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[36] Rrubaa Panchendrarajan and Aravindh Amaresan. Bidirectional lstm-crf for named entity recognition. In *Pacific Asia Conference on Language, Information and Computation*, 2018.

[37] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics.

[38] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

[39] Mariia Rizun. Knowledge graph application in education: a literature review. *Acta Universitatis Lodziensis. Folia Oeconomica*, 3:7–19, 08 2019.

[40] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[41] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. 1986.

[42] Edward W. Schneider. Course modularization applied: The interface system and its implications for sequence control and data analysis. page 21, 11 1973.

[43] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA, 2007. Association for Computing Machinery.

[44] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[45] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

[46] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, sep 2014.

[47] Thi-Hai-Yen Vuong, Minh-Quan Hoang, Tan-Minh Nguyen, Hoang-Trung Nguyen, and Ha-Thanh Nguyen. Constructing a knowledge graph for vietnamese legal cases with heterogeneous graphs, 2023.

[48] Minh Vũ. Tìm hiểu về confusion matrix trong machine learning, 2022.

[49] Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. A survey of joint intent detection and slot filling models in natural language understanding. *ACM Comput. Surv.*, 55(8), dec 2022.

[50] Fei Wu and Daniel S. Weld. Open information extraction using Wikipedia. In Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre, editors, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[51] Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland. TextRunner: Open information extraction on the web. In Bob Carpenter, Amanda Stent, and Jason D. Williams, editors, *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, Rochester, New York, USA, April 2007. Association for Computational Linguistics.

[52] Gokul Yenduri, Manju Ramalingam, Govardanan Chemmalar Selvi, Y. Supriya, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, G. Deepti Raj, Rutvij H. Jhaveri, B. Prabadevi, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. Gpt (generative pre-trained transformer)— a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE Access*, 12:54608–54649, 2023.

[53] Dong Yu, Shizhen Wang, and Li Deng. Sequential labeling using deep-structured conditional random fields. *IEEE Journal of Selected Topics in Signal Processing*, 4(6):965–973, 2010.

[54] Mengshi Yu, Jian Liu, Yufeng Chen, Jinan Xu, and Yujie Zhang. Cross-domain slot filling as machine reading comprehension. pages 3992–3998, 08 2021.

[55] Zekun Zheng, Xiaodong Wang, Xinye Lin, and Shaohe Lv. Get the best of the three worlds: Real-time neural image compression in a non-gpu environment. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 5400–5409, New York, NY, USA, 2021. Association for Computing Machinery.

[56] Lingfeng Zhong, Jia Wu, Qian Li, Hao Peng, and Xindong Wu. A comprehensive survey on automatic knowledge graph construction. *ACM Comput. Surv.*, 56(4), nov 2023.

A Mô hình sử dụng

A.1 nguyenvulebinh/vi-mrc-large

Mô hình này là một biến thể của XLM-RoBERTa được fine-tune cho nhiệm vụ Đọc hiểu máy.

Tập dữ liệu tiếng Việt được sử dụng để fine-tune cho mô hình này là:

- *mailong25:* Tập dữ liệu cá nhân dùng cho task QA cho ngôn ngữ tiếng Việt.
- *UIT-ViQuAD* Tập dữ liệu học có giám sát dành cho nhiệm vụ QA ngôn ngữ Việt. Tập dữ liệu này 23000 cặp câu hỏi - câu trả lời được làm thủ công dựa vào 5109 trích đoạn của 174 bài viết tiếng Việt từ Wikipedia.
- *UIT-ViQuaD 2.0* Tập dữ liệu cải tiến trên UIT-ViQuaD bằng cách thêm vào 12000 câu hỏi không thể trả lời, được sử dụng trong cuộc thi VLSP 2021. Điều này tạo thêm thử thách cho các mô hình vì nó cần phải chọn không trả lời khi gặp những câu hỏi này.
- *MLQA (MultiLingual Question Answering):* Là tập dữ liệu dùng để đánh giá các mô hình QA đa ngôn ngữ. Dữ liệu tiếng Việt cũng chứa trong tập này.

Đây là kết quả (do tác giả công bố) khi đánh giá trên 10% dữ liệu từ tập dữ liệu tiếng Việt trên:

Model	EM	F1
base	76.43	84.16
large	77.32	85.46

Bảng 9: Kết quả đánh giá của mô hình do tác giả công bố

Còn đây là kết quả mô hình phiên bản *large* khi đánh giá dựa vào tập dữ liệu kiểm tra trong cuộc thi VLSP 2021.

Model	EM	F1
public test set	85.847	83.826
private test set	82.072	78.071

Bảng 10: Kết quả đánh giá của mô hình trong cuộc thi VLSP 2021

A.2 NlpHUST/ner-vietnamese-electra-base

Mô hình này là một biến thể của ELECTRA được fine-tune cho nhiệm vụ NER cho tiếng Việt.

Tập dữ liệu tiếng Việt được sử dụng để fine-tune cho mô hình này được lấy từ VLSP shared task 2018.

Đây là kết quả (do tác giả công bố) khi đánh giá trên 10% dữ liệu từ tập dữ liệu tiếng Việt trên:

Lớp	**Số lượng mẫu**	**Precision**	**Recall**	**F1**
PER	2121	96,92%	96,37%	96,64%
ORG	1878	86,10%	90,68%	88,33%
LOC	2360	93,53%	93,77%	93,65%
MISC	174	56,60%	68,97%	62,18%
Tổng	6533	91,22%	93,07%	92,14%

Bảng 11: Kết quả đánh giá mô hình

Bộ siêu tham số dùng để chạy huấn luyện mô hình:

- learning rate: 5e-5
- training batch size: 16
- optimizer: Adam với betas=(0.9, 0.999) và epsilon=1e-8
- learning rate scheduler type: linear
- number of epochs: 10

B Bảng kết quả

B.1 Kiểu thực thể

No	Entity type	Entity Code	Sub-Entity Type	Example
1	Tên người	PER		
2	Tổ chức	ORG	trường, phòng ban công ty	
3	Địa điểm	LOC		cơ sở 2
4	Chương trình đào tạo	EDT		chính quy vừa học vừa làm
5	Khoa	FAC		
6	Ngành	MAJ		
7	Môn học	CRN		
8	Mã môn học	CSC		CO1027
9	Thời gian	DAT	Ngày, tháng, năm	
10	Học kì	SEM		học kì 231
11	Lớp	CLA		MT22B2KH
12	Nhóm	GRO		L01
13	Khoá	COH		K20
14	Mã số sinh viên	SID		1712308
15	Tài liệu	DOC		
16	Dịch vụ	SER		bkpay
17	Loại điểm	TSC		
18	Điểm số	SCO		
19	Số tiền	MON		
20	Đợt	BAT		
21	Thời điểm	TST		giữa kì
22	Đường dẫn	LIN		
23	Loại bệnh	DIS		Covid 19
24	*Sự kiện	EVT		
25	*Kì thi	EXM		
26	*Mã đề thi	EXC		
27	*Chuẩn	BEN		chuẩn av năm 2
28	*Số lượng	NUM		

Bảng 12: Tổng hợp các kiểu thực thể khám phá được

Các kiểu thực thể * là các kiểu được thêm vào trong quá trình gán nhãn.

B.2 Ý định - kiểu thực thể

Ý định	Kiểu thực thể
báo cáo vấn đề	'dịch vụ, hệ thống'*, 'thời gian'
báo lỗi phần mềm, dịch vụ	'môn học'*
cập nhật thời khóa biểu	'dịch vụ, hệ thống', 'môn học', 'học kì'*
cập nhật điểm	'học kì'*
gia hạn nộp chứng chỉ	'chứng chỉ'*, 'thời gian'
huỷ môn học	'môn học'*
hỏi học phí	'dịch vụ, hệ thống', 'học kì'*
hỏi kết quả đăng kí môn học	'đợt'*, 'môn học'
hỏi lịch thi	'môn học'*
hỏi thông tin dự thi	'thời gian'*
hỏi thông tin môn học	'môn học'*
hỏi thời gian nộp chứng chỉ	'chứng chỉ'*, 'học kì'
hỏi thủ tục	'thời gian'*
hỏi điều kiện tốt nghiệp	'khóa'*
hỏi điều kiện đăng kí môn học	'mã môn học', 'môn học'*
khiếu nại điểm	'loại điểm', 'thời gian', 'môn học'*
khôi phục thời khóa biểu	'môn học', 'học kì'*
nhận giấy xác nhận sinh viên	'tài liệu', 'thời gian'*
rút môn học	'nhóm', 'mã môn học'*
tăng sĩ số lớp	'mã môn học', 'môn học'*
xin mở lớp	'mã môn học', 'môn học'*
xin thi bù	'thời gian', 'môn học'*
đăng kí môn học	'mã môn học', 'môn học'*, 'học kì'
đăng kí thi tiếng Anh	'thời gian'*
được hỗ trợ	'môn học'*

Bảng 13: Sơ đồ ánh xạ các kiểu thực thể đến ý định

Các kiểu thực thể được đánh dấu trên bảng là kiểu thực thể chính, được dùng làm main triple trong các hyper triple (fact) trong đồ thị tri thức.