

**ỦY BAN NHÂN DÂN THÀNH PHỐ CẦN THƠ
TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN
KHOA HỌC MÁY TÍNH**

----- *** -----



Đồ án học phần 1
Ngành: Khoa học máy tính - 2022

***Đề tài: Xây dựng hệ thống tư vấn tuyển sinh trường
Đại học Công Nghệ - Kỹ Thuật Cần Thơ***

Giảng viên hướng dẫn	Sinh viên thực hiện
Ths. Lê Anh Nhã Uyên	1. Trương Lê Chương (MSSV: KHMT2211036) 2. Nguyễn Nhật Tiến (MSSV: KHMT2211031)

Cần Thơ, ngày 21 tháng 12 năm 2024

**ỦY BAN NHÂN DÂN THÀNH PHỐ CẦN THƠ
TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN
KHOA HỌC MÁY TÍNH**

----- *** -----



Đồ án học phần 1
Ngành: Khoa học máy tính - 2022

***Đề tài: Xây dựng hệ thống tư vấn tuyển sinh trường
Đại học Công Nghệ - Kỹ Thuật Cần Thơ***

Cán bộ hướng dẫn	Sinh viên thực hiện
Ths. Lê Anh Nhã Uyên	1. Trương Lê Chương (MSSV: KHMT2211036) 2. Nguyễn Nhật Tiến (MSSV: KHMT2211031)

Cần Thơ, ngày 21 tháng 12 năm 2024

NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN PHẢN BIỆN

Đồ án 1

Đề tài: *Xây dựng hệ thống tư vấn tuyển sinh trường Đại học Công Nghệ - Kỹ Thuật Cần Thơ*

Sinh viên thực hiện:

1. Trương Lê Chương (MSSV: KHMT2211036)
2. Nguyễn Nhật Tiến (MSSV: KHMT2211031)

Ngành: Khoa học máy tính – 2022

Giảng viên hướng dẫn: Th.S Lê Anh Nhã Uyên

Nhận xét, đánh giá:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Cần Thơ, ngày ... tháng ... năm 2024

Giảng viên phản biện

NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN PHẢN BIỆN

Đồ án 1

Đề tài: *Xây dựng hệ thống tư vấn tuyển sinh trường Đại học Công Nghệ - Kỹ Thuật Cần Thơ*

Sinh viên thực hiện:

1. Trương Lê Chương (MSSV: KHMT2211036)
2. Nguyễn Nhật Tiến (MSSV: KHMT2211031)

Ngành: Khoa học máy tính – 2022

Giảng viên hướng dẫn: Th.S Lê Anh Nhã Uyên

Nhận xét, đánh giá:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Cần Thơ, ngày ... tháng ... năm 2024

Giảng viên phản biện

LỜI CAM ĐOAN

Nhóm chúng tôi xin cam đoan rằng đồ án này là công trình nghiên cứu của nhóm thực hiện dựa trên những kiến thức đã được học, nghiên cứu và tìm hiểu một số đề tài và các phương án đi trước, không sao chép từ bất cứ công trình đã có trước đó. Mọi thứ được dựa trên sự cố gắng cũng như sự nỗ lực của bản thân.

Những phân có sử dụng tài liệu tham khảo có trong đồ án đã được liệt kê và nêu rõ ra tại phần tài liệu tham khảo.

Cần Thơ, ngày 15 tháng 12 năm 2022

Nhóm sinh viên thực hiện

Sinh viên 1



Trương Lê Chương

Sinh Viên 2



Nguyễn Nhật Tiến

TÓM TẮT ĐỒ ÁN

Đây là đồ án phục vụ cho việc nghiên cứu của sinh viên tìm hiểu về các mô hình ngôn ngữ BERT và PhoBERT. Với tham vọng tìm hiểu, học hỏi những điểm mới mẽ của các kỹ thuật này nên chúng tôi mạnh dạn đăng ký đề tài này mong rằng sẽ một phần nào đó giúp cho mọi người có thể nắm bắt và tìm hiểu được các mô hình ngôn ngữ BERT và PhoBERT.

Bố cục của đề tài: *Xây dựng hệ thống tư vấn tuyển sinh trường Đại học Kỹ Thuật - Công Nghệ Cần Thơ* bao gồm 5 chương:

- Chương 1: Tổng quan
- Chương 2: Cơ sở lý thuyết
- Chương 3: Phương pháp thực hiện
- Chương 4: Kết quả thực nghiệm và đánh giá giải pháp
- Chương 5: Kết luận và hướng phát triển
- Tài liệu tham khảo

LỜI CẢM ƠN

Lời đầu tiên, chúng tôi chân thành gửi lời cảm ơn đến tất cả các Thầy Cô trong khoa Công Nghệ Thông Tin đã hỗ trợ chúng tôi hoàn thành tốt đồ án lần này. Đặc biệt là giảng viên hướng dẫn *Th.S Lê Anh Nhã Uyên* đã tận tình dạy dỗ và truyền đạt những kiến thức quý báu trong suốt quá trình học tập vừa qua. Trong suốt thời gian thực hiện đồ án, chúng tôi đã học hỏi được rất nhiều kiến thức bổ ích về lĩnh vực *Xử Lý Ngôn Ngữ Tự Nhiên (NLP)* và *Học Máy* giúp chúng tôi có thể áp dụng vào công việc nghiên cứu và phát triển chatbot. Những kiến thức này sẽ là hành trang vững chắc để chúng tôi tiếp tục học hỏi, phát triển và ứng dụng trong tương lai.

Đồ án học phần 1 về Xây dựng hệ thống tư vấn tuyển sinh trường Đại học Công Nghệ - Kỹ Thuật Cần Thơ là một lĩnh vực mới mẻ nhưng vô cùng thú vị và có tính thực tiễn cao. Chúng tôi đã học được cách xây dựng và triển khai một chatbot cơ bản, đồng thời hiểu rõ hơn về các thuật toán và kỹ thuật cần thiết để cải thiện hiệu suất của chatbot trong việc giao tiếp và xử lý yêu cầu của người dùng. Tuy nhiên, do kiến thức của chúng tôi còn nhiều hạn chế và kinh nghiệm thực tiễn chưa đủ sâu sắc, mặc dù đã cố gắng hết sức nhưng đồ án này chắc chắn sẽ không thể tránh khỏi những thiếu sót và hạn chế. Chúng tôi rất mong nhận được sự góp ý và chỉ bảo của thầy/cô để đồ án của chúng tôi được hoàn thiện hơn.

Nếu có thêm thời gian và điều kiện, chúng tôi rất mong nhận được sự hướng dẫn sâu sắc hơn từ thầy/cô, giúp chúng tôi có thể triển khai những ý tưởng và công nghệ mới vào việc phát triển chatbot, từ đó nâng cao chất lượng của sản phẩm và mở rộng ứng dụng thực tế của nó.

Cuối cùng, nhóm xin kính chúc thầy/cô luôn dồi dào sức khỏe, thành công trong công tác giảng dạy và nghiên cứu, đồng thời mong rằng thầy/cô sẽ tiếp tục truyền cảm hứng cho chúng tôi và các thế hệ sinh viên sau này trong hành trình học tập và phát triển.

Xin chân thành cảm ơn!

MỤC LỤC

NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN PHẢN BIỆN	i
NHẬN XÉT, ĐÁNH GIÁ CỦA GIẢNG VIÊN PHẢN BIỆN	ii
LỜI CAM ĐOAN	iii
TÓM TẮT ĐỒ ÁN.....	iv
LỜI CẢM ƠN	v
MỤC LỤC	1
PHỤ LỤC HÌNH ẢNH.....	3
PHỤ LỤC BẢNG.....	4
Chương 1: TỔNG QUAN	5
1.1. Giới thiệu tổng quan và lý do chọn đề tài.....	5
1.2. Các nghiên cứu liên quan	6
1.2.1. Ngoài nước	6
1.2.2. Trong nước.....	7
1.3. Mục tiêu và phạm vi nghiên cứu	9
1.3.1. Mục tiêu.....	9
1.3.2. Phạm vi.....	9
1.4. Đối tượng nghiên cứu	9
1.5. Phương pháp nghiên cứu	9
Chương 2: CƠ SỞ LÝ THUYẾT	11
2.1. Máy học (Machine Learning)	11
2.1.1. Giới thiệu.....	11
2.1.2. Phân loại.....	11
2.1.3. Xây dựng mô hình	11
2.2. Học Sâu (Deep Learning)	13
2.3. Chatbot.....	15
2.4. Xử lý ngôn ngữ tự nhiên (NLP)	16
2.4.1. Khái niệm	16
2.5. Mô hình ngôn ngữ BERT	17
2.5.1. Kiến trúc mô hình BERT	17
2.5.2. Quy trình Pre-training và Fine-tuning model BERT	18
2.5.3. Các kiến trúc mô hình BERT.....	20

2.6. Mô hình ngôn ngữ PhoBERT	20
2.6.1. Các thành phần chính của PhoBert.....	21
2.6.2. Cách thức hoạt động của PhoBERT.....	22
2.7. Tiêu chí đánh giá mô hình.....	22
Chương 3: PHƯƠNG PHÁP THỰC HIỆN	24
3.1. Mô tả bài toán.....	24
3.2. Tập dữ liệu.....	24
3.3. Mô hình tổng quát.....	27
3.3.1. Dữ liệu đầu vào (JSON) – input Data (JSON): Patterns & Tags	27
3.3.2. Tiền xử lý dữ liệu - Preprocessing	27
3.3.3. Mô hình PhoBERT – PhoBERT Model	28
3.3.4. Quá trình huấn luyện – Training Process.....	28
3.3.5. Dự đoán và trả lời – Prediction & Response	29
3.4. Các giai đoạn giải quyết	29
3.4.1. Tiền xử lý dữ liệu.....	29
3.4.2. Huấn luyện mô hình.....	37
Chương 4: KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ GIẢI PHÁP	39
4.1. Kết quả thực nghiệm	39
4.1.1. Kết quả huấn luyện qua từng K-fold	39
4.1.2. Kết quả sau khi huấn luyện mô hình	42
4.2. So sánh đánh giá và lựa chọn mô hình.....	44
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	46
5.1. Kết quả đã đạt được	46
5.1.1. Kết quả	46
5.1.2. Hạn chế.....	46
5.2. Hướng phát triển.....	47
TÀI LIỆU THAM KHẢO.....	48

PHỤ LỤC HÌNH ẢNH

Hình 2. 1. Quy trình xây dựng mô hình máy học	13
Hình 2. 2 Kiến trúc của Deep Learning.....	14
Hình 2. 3 Tiến trình Pre-training và Fine-tuning của Bert	19
Hình 3. 1 Sơ đồ tập dữ liệu	26
Hình 3. 2 Mô hình tổng quát.....	27
Hình 3. 3 Kiểm tra dữ liệu	30
Hình 3. 4 Hàm vẽ biểu đồ trực quan các tag.....	30
Hình 3. 5 Biểu đồ trực quan các tag.....	31
Hình 3. 6 Hàm thực hiện loại bỏ dữ liệu dư thừa	32
Hình 3. 7 Biểu đồ từ khóa dạng đám mây (Word Cloud).....	33
Hình 3. 8 Biểu đồ phân phối tần suất (Frequency Distribution Histogram)	33
Hình 3. 9 Biểu đồ phân bố số lượng từ trong mỗi đoạn văn bản	35
Hình 3. 10 Biểu đồ trực quan độ dài trung bình của các từ	36
Hình 3. 11 Tăng cường dữ liệu	37
Hình 3. 12 Thông số các tham số và siêu tham số.....	38
Hình 4. 1 $k = 1$ (PhoBERT).....	39
Hình 4. 2 $k = 2$ (PhoBERT).....	40
Hình 4. 3 $k = 3$ (PhoBERT).....	40
Hình 4. 4 $k = 1$ (BERT)	41
Hình 4. 5 $k = 1$ (BERT)	41
Hình 4. 6 $k = 3$ (BERT)	42
Hình 4. 7 Kết quả mô hình PhoBERT	42
Hình 4. 8 Kết quả mô hình BERT.....	43

PHỤC LỤC BẢNG

Bảng 3. 1 Bảng kí tự các ngành.....	26
Bảng 4. 1 Bảng so sánh hai mô hình	44

Chương 1: TỔNG QUAN

1.1. Giới thiệu tổng quan và lý do chọn đề tài

Trong bối cảnh chuyển đổi số và sự phát triển nhanh chóng của trí tuệ nhân tạo (AI), việc ứng dụng các công nghệ AI, đặc biệt là xử lý ngôn ngữ tự nhiên (NLP), đang trở thành một yếu tố then chốt trong nhiều lĩnh vực, bao gồm cả giáo dục. Hệ thống tư vấn tuyển sinh của các trường đại học, vốn là cánh cửa đầu tiên kết nối giữa nhà trường và sinh viên tiềm năng, đòi hỏi phải được cải tiến để phù hợp với xu hướng số hóa và đáp ứng nhu cầu ngày càng cao về tính nhanh chóng, chính xác và minh bạch.

Trong lĩnh vực giáo dục, hệ thống tư vấn tuyển sinh truyền thống thường gặp nhiều hạn chế. Các quy trình thủ công thường gây mất thời gian, thiếu chính xác và khó cung cấp thông tin đầy đủ, kịp thời cho người học. Do đó, việc xây dựng một hệ thống tư vấn tuyển sinh tự động dựa trên công nghệ AI là một giải pháp tiềm năng, mang lại nhiều lợi ích vượt trội. Trên thế giới cũng đã có nhiều nghiên cứu ứng dụng NLP trong việc phát triển chatbot tư vấn tuyển sinh. Chẳng hạn, các nghiên cứu sử dụng mô hình GPT hoặc seq2seq [1] để tạo ra chatbot có khả năng trả lời câu hỏi tự động, giải đáp thắc mắc về quy trình tuyển sinh, và đưa ra gợi ý phù hợp với nhu cầu của thí sinh. Một số trường đại học đã triển khai các hệ thống chatbot này nhằm cải thiện trải nghiệm người dùng và giảm tải cho bộ phận tuyển sinh. Kết quả từ các nghiên cứu này cho thấy, các hệ thống NLP không chỉ giúp cung cấp thông tin nhanh chóng mà còn có khả năng học hỏi từ phản hồi của người dùng để ngày càng tối ưu hơn.

PhoBERT, được phát triển dựa trên mô hình BERT, là một mô hình xử lý ngôn ngữ tự nhiên tiên tiến được thiết kế đặc biệt cho tiếng Việt. Điểm khác biệt chính giữa PhoBERT và BERT nằm ở tập dữ liệu huấn luyện: BERT sử dụng dữ liệu tiếng Anh tổng quát, trong khi PhoBERT được huấn luyện trên một khối lượng dữ liệu tiếng Việt lớn. Điều này giúp PhoBERT hiểu ngữ nghĩa và ngữ cảnh tiếng Việt tốt hơn, từ đó hỗ trợ hiệu quả cho các bài toán như phân loại văn bản, tóm tắt thông tin, và tạo phản hồi tự động.

Trong đề tài này, chúng tôi thực hiện việc xây dựng hệ thống hỗ trợ tư vấn tuyển sinh dựa trên mô hình PhoBERT và so sánh với mô hình BERT. Hai mô hình này sẽ được huấn luyện và thực nghiệm trên tập dữ liệu tuyển sinh và dùng chung một bộ tham số để đánh giá hiệu năng và độ chính xác. Thông qua đó, chúng tôi phân tích các ưu, nhược điểm của từng mô hình để xác định mô hình phù hợp nhất cho hệ thống hỗ trợ tư vấn tuyển sinh tại trường Đại học Công nghệ Kỹ thuật Cần Thơ.

Đề tài "*Xây dựng hệ thống tư vấn tuyển sinh trường Đại học Công Nghệ - Kỹ Thuật Cần Thơ*" không chỉ nhằm giải quyết các vấn đề hiện tại mà còn hướng tới việc nâng cao hiệu quả và chất lượng của quy trình tư vấn tuyển sinh. Đồng thời, đề tài này còn góp phần tạo ra bước đột phá trong việc ứng dụng công nghệ AI vào giáo dục, hướng tới một hệ thống minh bạch, hiệu quả và tiện lợi cho cả nhà trường và sinh viên.

1.2. Các nghiên cứu liên quan

1.2.1. Ngoài nước

Ở phương diện sức khỏe, bài báo “*BERT-Based Medical Chatbot: Enhancing Healthcare Communication through Natural Language Understanding*” [2] phát triển một hệ thống chatbot y tế dựa trên BERT, tích hợp các phương pháp học sâu để cải thiện khả năng phản hồi các câu hỏi chứa từ ngữ chuyên ngành và cá nhân hóa phản hồi. Hiệu suất mô hình được chứng minh qua các chỉ số ấn tượng như Accuracy là 98%, Precision và Recall đều cao hơn 95%, đặc biệt AUC-ROC lên đến 97%, đưa ra khả năng dự đoán chính xác dựa trên các triệu chứng và truy vấn của người dùng. Chatbot y tế dựa trên BERT này không chỉ khắc phục những hạn chế của các phương pháp truyền thống mà còn đạt được hiệu suất vượt trội với độ chính xác, độ tin cậy và khả năng dự đoán cao, trở thành một công cụ giá trị để nâng cao chất lượng dịch vụ chăm sóc sức khỏe. Hơn nữa, khả năng cá nhân hóa các phản hồi giúp mô hình này dễ dàng đáp ứng các yêu cầu đặc thù của từng bệnh nhân, đồng thời tích hợp hiệu quả với các hệ thống y tế điện tử.

Ở phương diện học tập, “*The Impact of Pre-trained Transformer-Based Language Model Use on Student Learning Outcomes in Higher Education - A Mixed-Methods Research Approach with a Case Study of IMC Fachhochschule Krems*” [3] nghiên cứu sử dụng ChatGPT và các công cụ AI tạo sinh đối với kết quả học tập của sinh viên tại IMC Fachhochschule Krems. Nghiên cứu sử dụng phương pháp tiếp cận hỗn hợp toàn diện, kết hợp phân tích dữ liệu thu thập từ 154 sinh viên IMC, phỏng vấn với 6 giảng viên và nhà thiết kế khóa học của IMC, cùng tổng hợp dữ liệu từ 25 thí nghiệm quốc tế và 4 thí nghiệm quốc gia. Mô hình đã đem lại nhiều tác động tích cực như hỗ trợ sinh viên hoàn thành công việc và bài tập nhanh hơn, giảng viên có thể áp dụng AI nâng cao quá trình giảng dạy, đồng thời thúc đẩy đổi mới và phát triển các kỹ năng tư duy bậc cao (HOTS). Đặc biệt, việc sử dụng các mô hình ngôn ngữ tiên tiến như GPT không chỉ cải thiện hiệu quả học tập mà còn mở rộng khả năng ứng dụng AI vào quản lý giáo dục, chẳng hạn như cá nhân hóa lộ trình học tập hoặc đánh giá chính xác năng lực của từng sinh viên. Điều này góp phần thúc đẩy sự đổi mới trong giáo dục đại học và định hình cách học tập hiện đại hơn.

“*StrucBERT: Incorporating language structures into pre-training for deep language understanding*” [4] là một mô hình ngôn ngữ tiền huấn luyện BERT, được thiết kế dựa trên nghiên cứu về trật tự ngôn ngữ của Elman [5]. Nghiên cứu đã tích hợp cấu trúc ngôn ngữ vào tiền huấn luyện với hai nhiệm vụ là tập trung vào thứ tự từ và câu, tạo ra một mô hình mới – StructBERT. Kết quả thực nghiệm cho thấy mô hình StructBERT đạt GLUE benchmark: 89.0%, SQuAD v1.1: F1 score đạt 93.0%, SNLI: Độ chính xác đạt 91.7%. StructBERT không chỉ cải thiện khả năng hiểu ngôn ngữ mà còn gia tăng hiệu suất trong các ứng dụng đòi hỏi mức độ chính xác cao, chẳng hạn như phân tích văn bản pháp lý hoặc dịch thuật. Điều này mở ra tiềm năng lớn trong việc ứng

dụng các mô hình ngôn ngữ chuyên sâu vào các lĩnh vực yêu cầu xử lý ngữ nghĩa phức tạp.

“*BERTScore: EVALUATING TEXT GENERATION WITH BERT*” [6] giới thiệu BERTScore, một thước đo tự động mới để đánh giá việc sinh ngôn ngữ như dịch máy và chú thích hình ảnh. Khác với phương pháp so sánh bề mặt như BLEU, BERTScore đã sử dụng embedding ngữ cảnh từ mô hình BERT để tính toán điểm tương đồng giữa các từ trong câu tham chiếu và câu dự đoán. Trong dịch máy, BERTScore có độ tương quan tốt hơn với đánh giá của con người so với BLEU trên nhiều bộ dữ liệu phổ biến. Trong chú thích hình ảnh, BERTScore vượt qua SPICE, một thước đo chuyên dụng cho nhiệm vụ này. Ngoài việc cải thiện hiệu suất đánh giá, BERTScore còn mang lại những giá trị lớn trong việc tối ưu hóa các hệ thống sinh ngôn ngữ tự động, từ đó hỗ trợ các ứng dụng như tạo nội dung tự động hoặc phản hồi hội thoại thông minh hơn.

Nghiên cứu “*What does BERT look at? An Analysis of BERT’s Attention*” [7] khám phá các cơ chế chú ý trong các mô hình ngôn ngữ đã huấn luyện trước – đặc biệt là BERT, phân tích cách các attention heads xử lý thông tin ngữ pháp và cú pháp ngôn ngữ. Các attention head trong BERT học được các khía cạnh ngữ pháp mà không cần giám sát rõ ràng cho cú pháp hay đồng tham chiếu. Điều này chỉ ra rằng BERT đã học được các yếu tố ngữ pháp và cấu trúc câu một cách tự động từ dữ liệu chưa được gán nhãn thông qua quá trình huấn luyện tự giám sát. Các phát hiện này không chỉ giải thích lý do thành công của BERT trong xử lý ngôn ngữ tự nhiên mà còn gợi mở hướng phát triển các mô hình chú ý mới với khả năng học sâu hơn về cú pháp và ngữ nghĩa.

Các nghiên cứu quốc tế chứng minh rằng mô hình NLP dựa trên BERT có khả năng cách mạng hóa nhiều lĩnh vực như chăm sóc sức khỏe, giáo dục, và xử lý ngôn ngữ tự nhiên. Những ứng dụng thực tế từ chatbot y tế, hỗ trợ học tập, đến phân tích ngôn ngữ pháp lý đều cho thấy BERT không chỉ cải thiện đáng kể hiệu năng xử lý mà còn mở ra cơ hội phát triển các công cụ thông minh hơn, cá nhân hóa hơn. Kết quả từ các nghiên cứu này không chỉ cung cấp cơ sở khoa học mà còn khẳng định tiềm năng lớn của việc tích hợp AI vào giải quyết các vấn đề thực tế, định hướng sự đổi mới và nâng cao hiệu quả trong mọi lĩnh vực ứng dụng.

1.2.2. Trong nước

“*Một cách tiếp cận xây dựng ứng dụng Chatbot tư vấn tuyển sinh trường Đại học Đà Lạt*” [8] Bài báo này phát triển một hệ thống chatbot hỗ trợ quy trình tuyển sinh đại học, với mục tiêu tự động trả lời các câu hỏi của người dùng bất kể thời gian, giúp giải quyết vấn đề ngay cả ngoài giờ hành chính. Chatbot đóng vai trò là một trợ lý ảo, sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) để hiểu và phản hồi câu hỏi của người dùng một cách thích hợp. Hệ thống chatbot này được xây dựng dựa trên mô hình biểu diễn ngôn ngữ, một công nghệ tiên tiến trong NLP. Các mô hình BERT đã được tinh chỉnh để dự đoán câu trả lời từ câu hỏi đầu vào. Trong thử nghiệm, mô hình salti/bert-

base-multilingual-cased-finetuned-team đã thể hiện kết quả xuất sắc, với điểm F1 đạt 88,6% và điểm Exact Match (EM) đạt 79,6% trên tập dữ liệu thử nghiệm. Bên cạnh đó, đối với chức năng phân lớp ý định câu hỏi, chatbot đạt được tỷ lệ chính xác 99,9% và 100% trên các tập dữ liệu thử nghiệm và kiểm tra. Bài báo khẳng định rằng chatbot với mô hình BERT là công cụ hữu ích cho việc tự động hóa quy trình tuyển sinh tại Đại học Đà Lạt, giúp giảm thiểu sự phụ thuộc vào nhân lực và mang lại hiệu quả công việc cao hơn, đặc biệt trong việc xử lý nhiều cuộc trò chuyện đồng thời.

“A Question-Answering System for Vietnamese Public Administrative Services”
[9] Bài báo này đề xuất phát triển một hệ thống trả lời câu hỏi (QA) chuyên biệt cho ngữ cảnh dịch vụ hành chính công của Việt Nam. Hệ thống này kết hợp các kỹ thuật truy xuất thông tin (IR) và xếp hạng lại (re-ranking) để cung cấp các câu trả lời dựa trên từ vựng và ngữ nghĩa pháp lý cho các câu hỏi liên quan đến các tài liệu pháp lý Việt Nam. Bằng cách sử dụng các mô hình truy xuất dựa trên từ vựng và ngữ nghĩa, hệ thống này có thể trả lời các câu hỏi liên quan đến các tài liệu hành chính công của Việt Nam một cách chính xác hơn so với các mô hình hiện có. Mô hình đã tạo ra một bộ dữ liệu pháp lý gồm 785,996 đoạn văn từ các tài liệu pháp lý, kèm theo 4,547 cặp câu hỏi và câu trả lời trong lĩnh vực dịch vụ công và xây dựng một hệ thống QA chuyên biệt cho dịch vụ công, với các thành phần hoạt động qua các giai đoạn: hiểu, truy xuất, xếp hạng và tập hợp. Thông qua kết quả nghiên cứu cho thấy, hệ thống đã đạt được những cải tiến đáng kể trong việc truy xuất thông tin từ các tài liệu pháp lý Việt Nam và vượt trội hơn so với các mô hình khác.

Bài báo *“SỬ DỤNG BERT CHO TÓM TẮT VĂN BẢN TIẾNG VIỆT”* [10] giới thiệu phương pháp tóm tắt văn bản theo hai hướng: trích rút và tóm lược, sử dụng mô hình ngôn ngữ huấn luyện trước. Đối với bài toán trích rút, chúng tôi sử dụng mô hình BERTSum, trong đó BERT (Bidirectional Encoder Representations from Transformers) được sử dụng để mã hóa các câu đầu vào, và LSTM (Long Short Term Memory Networks) giúp biểu diễn mối quan hệ giữa các câu. Đối với bài toán tóm lược, BERT được sử dụng để mã hóa ngữ nghĩa của văn bản đầu vào và sinh ra bản tóm tắt phù hợp. Phương pháp của chúng tôi được thử nghiệm trên bộ dữ liệu tiếng Việt từ bài báo VNDS (A Vietnamese Dataset for Summarization) và đánh giá bằng chỉ số ROUGE (Recall-Oriented Understudy for Gisting Evaluation). Kết quả thực nghiệm cho thấy mô hình BERT đạt hiệu quả tốt hơn trong bài toán tóm tắt trích rút.

Bài báo *“SỬ DỤNG BERT VÀ CÂU PHỤ TRỢ CHO TRÍCH XUẤT KHÍA CẠNH TRONG VĂN BẢN TIẾNG VIỆT”* [11] trình bày một phương pháp trích xuất khía cạnh mới, sử dụng mô hình ngôn ngữ huấn luyện trước như BERT, với khả năng mô hình hóa nhúng từ theo ngữ cảnh. Khác với các nghiên cứu trước chỉ sử dụng một câu đầu vào để trích xuất khía cạnh, bài báo đề xuất sử dụng câu phụ trợ tạo ra từ các từ khóa khía cạnh, kết hợp với câu đầu vào để tạo thành cặp câu cho BERT. Mô hình đề

xuất cho thấy kết quả khả quan khi sử dụng dữ liệu tiếng Việt từ các bài đánh giá nhà hàng trên mạng xã hội.

“MỘT CẢI TIẾN CỦA PHOBERT NHẪM TĂNGKHẢ NĂNG HIỂU TIẾNG VIỆT CỦA CHATBOT THÔNG TIN KHÁCH SẠN” [12] Chatbot hỗ trợ thông tin du lịch bằng tiếng Việt đang ngày càng được quan tâm, tuy nhiên, các chatbot truyền thống dựa trên quy tắc và kiến thức hạn chế thường gặp khó khăn khi xử lý yêu cầu phức tạp hoặc ngoài kịch bản. Để cải thiện điều này, bài báo đề xuất cải tiến mô hình tiền huấn luyện để xây dựng chatbot hỗ trợ thông tin khách sạn bằng tiếng Việt. Các tác giả điều chỉnh và đánh giá các mô hình tiền huấn luyện, và kết quả thử nghiệm cho thấy mô hình cải tiến đạt hiệu quả cao, với Accuracy 96,4%, F1-score 96,9%, và Precision 97,4%.

1.3. Mục tiêu và phạm vi nghiên cứu

1.3.1. Mục tiêu

Xây dựng hệ thống hỗ trợ tuyển sinh cho trường đại học Công nghệ Kỹ thuật Cần Thơ (CTUET), sử dụng mô hình PhoBERT và BERT để xử lý và tự động hoá các quy trình phân tích dữ liệu ngôn ngữ.

1.3.2. Phạm vi

- Tập trung vào các bài toán phân lớp và tóm tắt dữ liệu văn bản tuyển sinh.
- Đề tài được áp dụng trong phạm vi tuyển sinh của trường đại học Công nghệ Kỹ thuật Cần Thơ (CTUET).

1.4. Đối tượng nghiên cứu

Đối tượng nghiên cứu trong đề tài bao gồm:

- Các dữ liệu tuyển sinh: thông tin trường, thông tin tổ hợp tuyển sinh của ngành, thông tin về các ngành tuyển sinh.
- Mô hình PhoBERT và BERT trong xử lý ngôn ngữ tự nhiên.

1.5. Phương pháp nghiên cứu

- Thu thập dữ liệu tuyển sinh từ hệ thống tuyển sinh của trường Đại học Công nghệ Kỹ thuật Cần Thơ.
- Tiền xử lý dữ liệu kiểm tra dữ liệu có noise (nhiều), null (thiếu), dup (trùng).
- Tham khảo các tài liệu nghiên cứu về mô hình hoặc các tiêu chí liên quan đến đề tài ở trong và ngoài nước.
- Nghiên cứu các lý thuyết thông qua các tài liệu có liên quan đến đề tài.
- Huấn luyện học máy sử dụng mô hình Bert và PhoBert để học các dữ liệu về câu hỏi và câu trả lời trong việc tuyển sinh.

- Tiến hành phân tích, so sánh đánh giá kết quả của hai mô hình PhoBert và Bert.
- Phát triển hệ thống: Tích hợp mô hình xử lý ngôn ngữ vào hệ thống tuyến sinh.
- Kiểm thử và đánh giá: Thực hiện các bài kiểm thử để đánh giá độ chính xác và tính hiệu quả của hệ thống.

Chương 2: CƠ SỞ LÝ THUYẾT

2.1. Máy học (Machine Learning)

2.1.1. Giới thiệu

Máy học (Machine Learning - ML) là một lĩnh vực con trong trí tuệ nhân tạo (Artificial Intelligence - AI), tập trung vào việc phát triển các thuật toán và mô hình cho phép máy tính tự động học hỏi từ dữ liệu, nhận diện các mẫu và đưa ra dự đoán hoặc quyết định. Thay vì được lập trình cụ thể để thực hiện từng nhiệm vụ, các hệ thống máy học có khả năng cải thiện hiệu suất của mình qua thời gian khi được cung cấp thêm dữ liệu mới. Quá trình này mô phỏng cách con người học hỏi từ kinh nghiệm, cho phép máy tính thực hiện nhiều tác vụ phức tạp, từ nhận dạng hình ảnh và xử lý ngôn ngữ tự nhiên đến dự báo và phân tích dữ liệu [13].

2.1.2. Phân loại

Máy học có thể được phân loại thành bốn loại chính:

- Học có giám sát (Supervised Learning): Là phương pháp học trong đó mô hình được huấn luyện với dữ liệu có nhãn, tức là mỗi dữ liệu đầu vào (input) đều có kết quả đầu ra (output) xác định. Mục tiêu của học có giám sát là tìm ra một hàm từ đầu vào đến đầu ra sao cho khi nhận được dữ liệu mới, mô hình có thể dự đoán kết quả chính xác. Các bài toán điển hình trong học có giám sát bao gồm phân loại (classification) và hồi quy (regression).
- Học không giám sát (Unsupervised Learning): Đây là phương pháp học mà trong đó dữ liệu không có nhãn. Mô hình không được cung cấp kết quả đầu ra và phải tự học cách phân nhóm, tìm ra các mẫu trong dữ liệu. Các thuật toán học không giám sát thường được sử dụng để phân cụm (clustering) hoặc khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước (association).
- Học bán giám sát (Semi-supervised Learning): Là sự kết hợp giữa học có giám sát và học không giám sát. Trong phương pháp này, một phần của dữ liệu có nhãn còn lại là dữ liệu không có nhãn. Mô hình học từ cả hai loại dữ liệu này để cải thiện độ chính xác của dự đoán.
- Học tăng cường (Reinforcement Learning): Là phương pháp mà trong đó một tác nhân (agent) học thông qua quá trình thử và sai, tương tác với môi trường và nhận phản hồi dưới dạng các phần thưởng (rewards) hoặc hình phạt (penalties). Mục tiêu của học củng cố là tối đa hóa phần thưởng trong suốt quá trình học [14].

2.1.3. Xây dựng mô hình

Quá trình xây dựng một mô hình máy học được chia thành các bước cụ thể sau, nhằm đảm bảo tính chính xác, hiệu quả và khả năng áp dụng thực tiễn của mô hình:

Bước 1: Thu thập và chuẩn bị dữ liệu: Dữ liệu là yếu tố quan trọng nhất và là nền tảng của mọi mô hình máy học.

- Chất lượng dữ liệu: Dữ liệu thu thập phải đảm bảo không chứa nhiều lỗi hoặc sai sót, đồng thời phải đại diện cho toàn bộ phân phối thực tế của bài toán. Ví dụ: đối với một chatbot hỗ trợ khách sạn, dữ liệu nên bao gồm các câu hỏi từ khách hàng về đặt phòng, dịch vụ, và thông tin liên quan.

- Nguồn dữ liệu: Dữ liệu có thể được thu thập từ nhiều nguồn khác nhau như cơ sở dữ liệu nội bộ, khảo sát người dùng, hoặc các tập dữ liệu mở từ cộng đồng.

- Tính cân bằng dữ liệu: Đảm bảo rằng các lớp trong bài toán phân loại không bị lệch, để mô hình không bị thiên lệch (bias).

Bước 2: Tiền xử lý dữ liệu: Đây là bước xử lý dữ liệu thô thành dạng có thể sử dụng được cho mô hình.

- Giải quyết giá trị thiếu (Missing Data): Thay thế giá trị thiếu bằng trung bình, trung vị, hoặc giá trị phổ biến nhất (đối với dữ liệu định lượng) hoặc loại bỏ các hàng có giá trị thiếu.

- Loại bỏ dữ liệu ngoại lệ (Outliers): Sử dụng các kỹ thuật như IQR (Interquartile Range) hoặc Z-score để phát hiện và xử lý các điểm dữ liệu bất thường.

- Chuẩn hóa hoặc chuẩn hóa dữ liệu (Scaling): Đối với các thuật toán nhạy cảm với độ lớn của dữ liệu (như Logistic Regression, SVM), việc đưa dữ liệu về cùng một khoảng giá trị là rất quan trọng. Các phương pháp phổ biến bao gồm Min-Max Scaling hoặc Standardization.

Bước 3: Chọn mô hình và huấn luyện mô hình:

- Chọn thuật toán: Dựa trên loại bài toán, người phát triển chọn một thuật toán phù hợp. Ví dụ:

- Phân loại (Classification): SVM, Decision Trees, Random Forest, hoặc Neural Networks.

- Hồi quy (Regression): Linear Regression, Ridge, Lasso, hoặc Gradient Boosting.

- Phân chia dữ liệu: Dữ liệu được chia thành tập huấn luyện (training set) và tập kiểm tra (test set) theo tỷ lệ phổ biến như 80/20 hoặc 70/30.

- Huấn luyện: Sử dụng tập huấn luyện để dạy mô hình nhận biết các mẫu (patterns) trong dữ liệu. Các tham số trong thuật toán sẽ được điều chỉnh để tối ưu hóa một hàm mục tiêu (loss function).

Bước 4: Đánh giá mô hình: Sau khi huấn luyện, mô hình được đánh giá trên tập kiểm tra để kiểm tra hiệu suất.

- Chỉ số đánh giá:
 - Độ chính xác (Accuracy): Tỷ lệ dự đoán đúng trên tổng số dự đoán.
 - F1-score: Trung bình điều hòa của Precision và Recall, phù hợp với các bài toán có dữ liệu mất cân bằng.
 - AUC-ROC: Đánh giá khả năng phân biệt giữa các lớp của mô hình, thường được sử dụng cho các bài toán nhị phân.
- Overfitting và Underfitting: Đảm bảo rằng mô hình không quá phù hợp với dữ liệu huấn luyện (overfitting) và không bỏ sót các mẫu quan trọng (underfitting).

Bước 5: Tối ưu hóa và cải thiện mô hình

- Tinh chỉnh tham số (Hyperparameter Tuning): Sử dụng các phương pháp như Grid Search hoặc Random Search để tìm ra tổ hợp tham số tốt nhất. Ví dụ, trong Random Forest, số lượng cây (n_estimators) hoặc độ sâu tối đa của cây (max_depth) có thể được điều chỉnh. Cross-validation: Kỹ thuật chia dữ liệu thành nhiều phần để đánh giá mô hình một cách tổng quát hơn, chẳng hạn như K-Fold Cross-Validation.
- Sử dụng kỹ thuật ensemble: Kết hợp nhiều mô hình lại với nhau (như Bagging hoặc Boosting) để tăng độ chính xác và giảm lỗi của mô hình. Các mô hình như XGBoost, AdaBoost, và Gradient Boosting Machine là những ví dụ nổi bật.



Hình 2. 1. Quy trình xây dựng mô hình máy học

2.2. Học Sâu (Deep Learning)

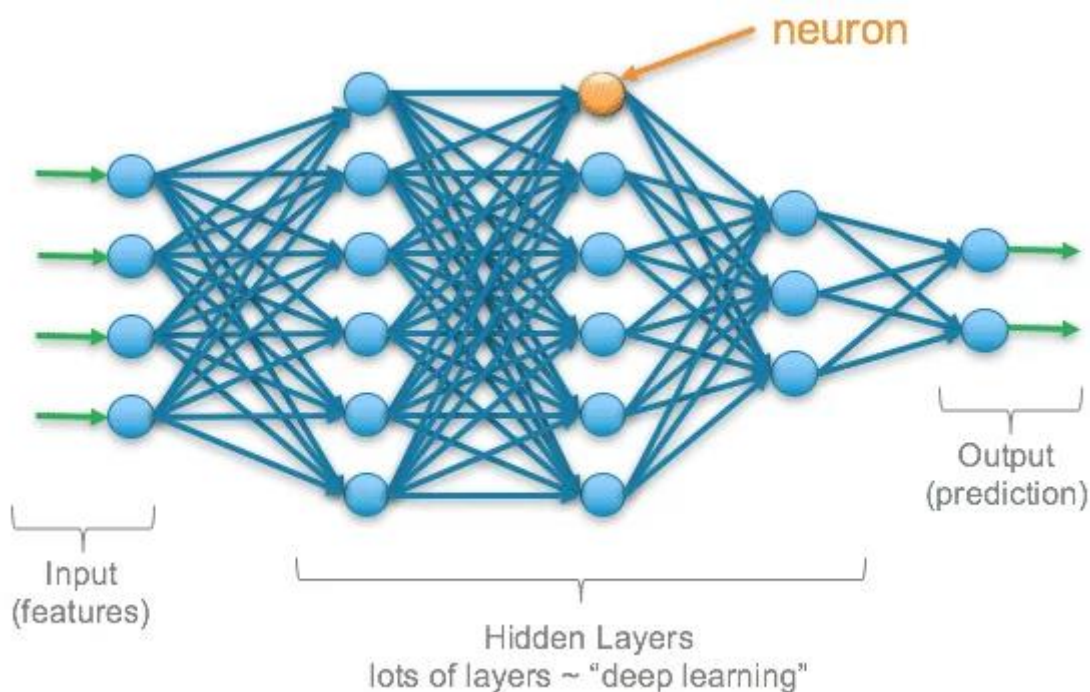
“Deep learning” hay học sâu là một trong các phương pháp của máy học (machine learning) các mô hình được đào tạo bằng cách sử dụng mạng thần kinh nhân tạo để phân tích dữ liệu về nhiều chi tiết khác nhau bằng các thuật toán mô phỏng theo hệ thần kinh của con người [12]. Các mô hình học sâu có thể được dạy để thực hiện các nhiệm vụ

phân loại và nhận dạng các mẫu trong ảnh, văn bản, âm thanh và các dữ liệu khác nhau và nó có thể đạt được độ chính xác cao hơn cả não bộ con người. Nổi bật của mô hình học sâu có thể kể đến như Mạng nơ-ron nhân tạo (ANN), Mạng nơ-ron tích chập (CNN) và Mạng thần kinh hồi quy (RNN),... Mô hình thực hiện việc học hỏi từ một lượng lớn dữ liệu được cung cấp để giải quyết một vấn đề cụ thể tương tự như cách mà con người học một điều mới. Trong đó mô hình sẽ thực hiện một nhiệm vụ lặp đi lặp lại nhiều lần, mỗi lần sẽ chỉnh sửa nhiệm vụ một chút để cải thiện kết quả.

Trong mô hình “Deep learning”, kiến trúc chung của mạng nơ-ron nhân tạo bao gồm: lớp đầu vào, các lớp ẩn và lớp đầu ra.

- Lớp đầu vào (Input layer): lớp này thể hiện cho các dữ liệu đầu vào. Mỗi nơ-ron tương ứng với một thuộc tính (Attribute) hoặc đặc trưng (feature) của dữ liệu đầu vào. Lớp ẩn (Hidden layer): Có thể có nhiều hơn một “hidden layer”. Lớp ẩn thể hiện cho quá trình xử lý thông tin và suy luận của mạng trước khi đưa ra đầu ra. Các neuron trong các lớp ẩn này thực hiện các phép tính tuyến tính (linear) và phi tuyến tính (non-linear) để biểu diễn dữ liệu.

- Lớp đầu ra (Output layer): thể hiện cho giá trị đầu ra của mạng nơ-ron. Sau khi đi qua hàm “hidden layer” cuối cùng, dữ liệu sẽ được chuyển hóa bằng một hàm số được gọi là hàm kích hoạt (activation function) và trả về đầu ra cuối cùng. Lớp đầu ra thường đưa ra dự đoán hoặc kết quả dựa trên dữ liệu đầu vào.



Hình 2. 2 Kiến trúc của Deep Learning

2.3. Chatbot

Chatbot là một ứng dụng của trí tuệ nhân tạo (AI), được thiết kế để tương tác với con người qua giao diện ngôn ngữ tự nhiên. Chúng có khả năng xử lý và phản hồi các yêu cầu của người dùng theo thời gian thực, hỗ trợ nội dung tương tác trong nhiều ngôn ngữ, giải quyết các vấn đề hoặc cung cấp thông tin một cách hiệu quả. Công nghệ chatbot không phải là một chủ đề mới và đã xuất hiện từ năm 1966 với sự ra đời của chương trình chatbot đầu tiên mang tên ELIZA [15]. Đây là một chương trình do Joseph Weizenbaum phát triển, nhằm mô phỏng một nhà trị liệu tâm lý bằng cách trả lời các câu hỏi từ người dùng dựa trên các mẫu câu được lập trình sẵn. ELIZA đánh dấu bước khởi đầu quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và trí tuệ nhân tạo (AI). Các chatbot hiện đại đã tiến xa so với những chatbot đời đầu. Không chỉ trả lời câu hỏi dựa trên kịch bản có sẵn, chúng còn sử dụng các thuật toán học máy (Machine Learning) và học sâu (Deep Learning) để học từ dữ liệu người dùng, từ đó cải thiện hiệu suất và độ chính xác. Những cải tiến này giúp chatbot ngày càng thông minh hơn, có khả năng đưa ra các phản hồi tự nhiên và phù hợp hơn với ngữ cảnh của người dùng.

Chatbot được chia thành hai nhóm chính:

- Chatbot dựa trên quy tắc (Rule-based Chatbot): Loại chatbot này hoạt động theo một tập luật đã được thiết lập trước. Chúng chỉ phụ thuộc vào những túi câu hỏi và câu trả lời cố định, dùng để giải quyết các vấn đề đơn giản. Đây là loại chatbot phổ biến trong các dịch vụ khách hàng hoặc câu hỏi thường gặp (FAQ).
- Chatbot dựa trên AI (AI-based Chatbot): Loại chatbot này sử dụng các kỹ thuật toán học máy (machine learning) và xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) để tìm hiểu và tương tác với người dùng một cách linh hoạt. AI-based chatbot thường được tạo bằng các mô hình như BERT, GPT [16].

Các bước phát triển Chatbot:

- Bước 1: Xác định mục tiêu: Hiểu rõ chatbot sẽ được sử dụng trong ngữ cảnh nào và để giải quyết những vấn đề gì.
- Bước 2: Thu thập dữ liệu: Dữ liệu chất lượng cao là yếu tố quan trọng nhất cho mô hình AI-based chatbot. Việc thu thập dữ liệu thường bao gồm dữ liệu văn bản, câu hỏi, câu trả lời từ người dùng.
- Bước 3: Huấn luyện mô hình: Huấn luyện mô hình chatbot là bước quan trọng để đảm bảo hiệu suất và khả năng hiểu ngôn ngữ tự nhiên. Các mô hình AI phổ biến như Transformer, seq2seq, và GPT được sử dụng để xử lý dữ liệu và tạo ra các câu trả lời phù hợp. Ngoài ra, các phương pháp huấn luyện như huấn luyện có giám sát (supervised learning), không giám sát (unsupervised learning), và học tăng cường (reinforcement learning) cũng được áp dụng. Mỗi phương pháp có ưu điểm và nhược điểm riêng: ví dụ, học có giám sát yêu cầu nhiều dữ liệu được gán nhãn, trong khi học

tăng cường phù hợp với các kịch bản yêu cầu tương tác liên tục. So sánh hiệu quả giữa các phương pháp này thường phụ thuộc vào loại dữ liệu và mục tiêu cụ thể của chatbot.

- Bước 4: Tích hợp và triển khai: Tích hợp chatbot vào các nền tảng như website, Facebook Messenger để dễ dàng triển khai cho người dùng.

2.4. Xử lý ngôn ngữ tự nhiên (NLP)

2.4.1. Khái niệm

Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) [17] là một nhánh của Trí tuệ Nhân tạo (Artificial Intelligence - AI), tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người. NLP cho phép máy tính xử lý và hiểu ngôn ngữ của con người dưới hai dạng chính: tiếng nói (speech) và văn bản (text). Mục tiêu chính của lĩnh vực này bao gồm:

- Hiểu ngôn ngữ tự nhiên: giúp máy tính nhận diện và giải thích chính xác các ngữ nghĩa, ngữ pháp, và ngữ cảnh trong giao tiếp ngôn ngữ.
- Tương tác tự nhiên giữa con người và máy móc: hỗ trợ việc giao tiếp, ra lệnh hoặc điều khiển thiết bị bằng ngôn ngữ tự nhiên.
- Tăng cường khả năng xử lý thông tin: ứng dụng trong phân tích văn bản, xử lý dữ liệu lớn hoặc cải thiện hiệu quả giao tiếp giữa con người.

Lĩnh vực NLP xuất hiện từ thập niên 1940, bắt đầu với những hệ thống dựa trên quy tắc thủ công (rule-based systems) và sau đó phát triển mạnh mẽ nhờ sự ra đời của các thuật toán hiện đại như học máy (machine learning) và học sâu (deep learning). Phương pháp xử lý ngôn ngữ tự nhiên đã trải qua nhiều giai đoạn với những mô hình khác nhau, có thể kể đến:

- Sử dụng Ô-tô-mát (Automata) và mô hình xác suất: Phương pháp này dựa trên lý thuyết xác suất và thống kê để xử lý các vấn đề về ngôn ngữ, như mô hình Markov ẩn (Hidden Markov Model - HMM) [18] hoặc các công cụ như N-gram [19].
- Các phương pháp dựa trên ký hiệu: Tập trung vào ngữ pháp và cấu trúc ngôn ngữ, thường được ứng dụng trong giai đoạn đầu của NLP [20].
- Phương pháp ngẫu nhiên: Tận dụng các thuật toán dựa trên dữ liệu mẫu lớn để dự đoán kết quả [21].
- Học máy truyền thống: Các thuật toán như Support Vector Machines (SVM), Naive Bayes, hoặc Decision Trees được áp dụng để phân loại, trích xuất thông tin hoặc nhận dạng thực thể [22].
- Học sâu: Với sự xuất hiện của Mạng nơ-ron nhân tạo (Artificial Neural Networks) và các mô hình như Transformer, phương pháp này đã cải thiện đáng kể hiệu suất của NLP, đặc biệt trong dịch máy và tạo văn bản [23].

NLP có thể chia thành hai nhánh lớn, tương đối độc lập nhưng hỗ trợ lẫn nhau:

- Xử lý tiếng nói (Speech Processing): Nghiên cứu và phát triển các thuật toán để nhận dạng, tổng hợp, hoặc chuyển đổi tiếng nói. Ví dụ:

- Nhận dạng tiếng nói (Speech Recognition): Chuyển tiếng nói thành văn bản, ứng dụng trong điều khiển bằng giọng nói hoặc trợ lý ảo như Siri, Google Assistant.

- Tổng hợp tiếng nói (Speech Synthesis): Biến văn bản thành tiếng nói, thường được sử dụng trong sách nói hoặc các thiết bị hỗ trợ đọc cho người khiếm thị.

- Xử lý văn bản (Text Processing): Tập trung phân tích và xử lý dữ liệu văn bản. Một số ứng dụng quan trọng bao gồm:

- Tìm kiếm và truy xuất thông tin (Information Retrieval): Trích xuất nội dung phù hợp từ tập dữ liệu lớn.

- Dịch máy (Machine Translation): Tự động dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác, ví dụ: Google Translate.

- Tóm tắt văn bản tự động (Automatic Summarization): Tạo ra các bản tóm tắt ngắn gọn, tập trung vào nội dung chính của văn bản gốc.

2.5. Mô hình ngôn ngữ BERT

BERT (Bidirectional Encoder Representations from Transformers) được Google công bố vào năm 2018 và gần đây đã đạt được hiệu suất xuất sắc trong một loạt các nhiệm vụ xử lý ngôn ngữ tự nhiên (NLP), bao gồm trả lời câu hỏi và suy luận ngôn ngữ. BERT được chứng minh là đơn giản về mặt lý thuyết và mạnh mẽ về mặt thực nghiệm vì nó đạt được kết quả xuất sắc trên mười một nhiệm vụ NLP.

BERT không giống như các mô hình ngôn ngữ truyền thống, vốn chỉ sử dụng một hướng (trái sang phải hoặc phải sang trái), mà BERT sử dụng một phương pháp học bidirectional (hai chiều) để hiểu ngữ cảnh của từ dựa trên cả ngữ cảnh trước và sau của nó. Mô hình BERT được thiết kế để tiền huấn luyện các biểu diễn ngữ nghĩa từ văn bản chưa gán nhãn. Sau khi tiền huấn luyện, BERT có thể được tinh chỉnh cho các nhiệm vụ NLP cụ thể chỉ với một lớp đầu ra bổ sung. Điều này cho phép BERT thực hiện nhiều nhiệm vụ khác nhau mà không cần thay đổi kiến trúc phức tạp cho từng nhiệm vụ cụ thể.

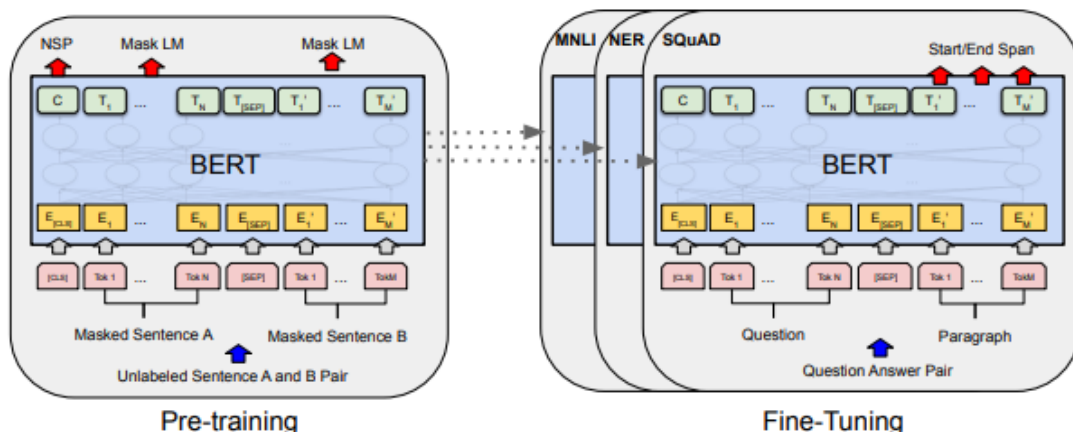
2.5.1. Kiến trúc mô hình BERT

Kiến trúc của BERT dựa trên Transformer, một mô hình học sâu mạnh mẽ đã được giới thiệu trong bài báo "Attention is All You Need" (Vaswani et al., 2017). Mô hình Transformer sử dụng cơ chế "self-attention" để xử lý dữ liệu theo cách song song và hiệu quả, thay vì theo trình tự như các mô hình RNN hay LSTM trước đây.

Các thành phần chính của mô hình BERT:

- Encoder của Transformer: BERT chỉ sử dụng phần encoder trong Transformer, với mục đích mã hóa đầu vào thành các biểu diễn ngữ nghĩa sâu. Mô hình BERT có thể có nhiều lớp encoder, từ 12 lớp (BERT-base) đến 24 lớp (BERT-large), giúp mô hình học được các biểu diễn ngữ nghĩa phức tạp hơn.
- Self-attention: Mô hình sử dụng cơ chế self-attention để tính toán mức độ quan trọng của từng từ trong câu đối với các từ khác. Điều này giúp BERT học được mối quan hệ giữa các từ trong câu một cách toàn diện, từ đó xây dựng biểu diễn ngữ nghĩa chính xác.
- Input Representation: Đầu vào của BERT được đại diện bởi ba thành phần:
 - Token embeddings: Mỗi từ hoặc subword trong câu được ánh xạ thành một vector nhúng.
 - Segment embeddings: Được sử dụng để phân biệt giữa các đoạn văn bản trong các nhiệm vụ như câu trả lời câu hỏi (question answering).
 - Position embeddings: Thể hiện vị trí của các từ trong câu, giúp mô hình hiểu được thứ tự của các từ.
- Masked Language Modeling (MLM): Trong quá trình tiền huấn luyện, BERT sử dụng phương pháp Masked Language Modeling (MLM) để ẩn một phần từ trong câu và yêu cầu mô hình dự đoán từ bị thiếu đó. Điều này giúp BERT học được các biểu diễn ngữ nghĩa mà không cần phải phụ thuộc vào bất kỳ ngữ cảnh cụ thể nào.
- Next Sentence Prediction (NSP): BERT cũng được huấn luyện với một nhiệm vụ thứ hai gọi là Dự đoán Câu Tiếp Theo (NSP). Trong nhiệm vụ này, mô hình học cách xác định liệu một câu có phải là câu tiếp theo hợp lý của một câu trước đó hay không. Điều này hữu ích trong các nhiệm vụ như câu trả lời câu hỏi, nơi cần phải xác định mối quan hệ giữa hai câu.

2.5.2. Quy trình Pre-training và Fine-tuning model BERT



Hình 2. 3 Tiến trình Pre-training và Fine-tuning của Bert

A. Pre-training (Tiền huấn luyện)

BERT được tiền huấn luyện trên một lượng lớn dữ liệu ngôn ngữ tự nhiên (Wikipedia, sách, tài liệu khoa học) để học các biểu diễn ngữ nghĩa chung của từ và câu. Quá trình này không hướng đến một nhiệm vụ cụ thể mà tập trung vào xây dựng kiến thức nền tảng. BERT thực hiện hai nhiệm vụ quan trọng trong giai đoạn tiền huấn luyện:

- Masked Language Modeling (MLM): Một số từ trong câu được "che" lại bằng token đặc biệt [MASK], và BERT phải dự đoán từ bị che đó.
- Next Sentence Prediction (NSP): BERT học mối quan hệ giữa hai câu để hiểu cách chúng liên kết với nhau.
 - Positive Pair: Hai câu liên tiếp trong văn bản.
 - Negative Pair: Hai câu không liên quan, được ghép ngẫu nhiên.

Sau giai đoạn này, BERT đã học các biểu diễn ngữ nghĩa sâu sắc, nhưng chưa được tinh chỉnh cho một tác vụ cụ thể nào.

B. Fine-tuning (Tinh chỉnh)

Quá trình tinh chỉnh sử dụng mô hình BERT đã được tiền huấn luyện để giải quyết một bài toán cụ thể, như phân loại văn bản, trả lời câu hỏi, hoặc dịch máy. Giai đoạn này bao gồm:

- Thêm một hoặc nhiều lớp đầu ra phù hợp với bài toán.
- Tối ưu hóa trọng số của mô hình dựa trên dữ liệu cụ thể.

Tiến trình áp dụng fine-tuning sẽ như sau:

- Bước 1: Embedding toàn bộ các token của cặp câu bằng các véc tơ nhúng từ pretrain model. Các token embedding bao gồm cả 2 token là [CLS] và [SEP] để đánh dấu vị trí bắt đầu của câu hỏi và vị trí ngăn cách giữa 2 câu. 2 token này sẽ được dự báo ở output để xác định các phần Start/End Span của câu output.
- Bước 2: Các embedding véc tơ sau đó sẽ được truyền vào kiến trúc multi-head attention với nhiều block code (thường là 6, 12 hoặc 24 blocks tùy theo kiến trúc BERT). Ta thu được một véc tơ output ở encoder.
- Bước 3: Để dự báo phân phối xác suất cho từng vị trí từ ở decoder, ở mỗi time step chúng ta sẽ truyền vào decoder véc tơ output của encoder và véc tơ embedding input của decoder để tính encoder-decoder attention (cụ thể về encoder-decoder attention là gì các bạn xem lại mục 2.1.1). Sau đó projection qua liner layer và softmax để thu được phân phối xác suất cho output tương ứng ở time step t.

- Bước 4: Trong kết quả trả ra ở output của transformer ta sẽ cố định kết quả của câu Question sao cho trùng với câu Question ở input. Các vị trí còn lại sẽ là thành phần mở rộng Start/End Span tương ứng với câu trả lời tìm được từ câu input.

Lưu ý quá trình huấn luyện chúng ta sẽ fine-tune lại toàn bộ các tham số của model BERT đã cut off top linear layer và huấn luyện lại từ đầu các tham số của linear layer mà chúng ta thêm vào kiến trúc model BERT để customize lại phù hợp với bài toán.

2.5.3. Các kiến trúc mô hình BERT

Hiện tại có nhiều phiên bản khác nhau của mô hình BERT. Các phiên bản đều dựa trên việc thay đổi kiến trúc của Transformer tập trung ở 3 tham số:

- L: số lượng các khối các tầng con trong transformer
- H: kích thước của VECTOR nhúng (hay còn gọi là hidden size)
- A: Số lượng từ đầu trong tầng nhiều từ đầu (multi-head layer), mỗi một từ đầu sẽ thực hiện một cơ chế tự chú ý (self-attention).

Tên gọi của 2 kiến trúc bao gồm:

- BERTBASE (L=12,H=768,A=12): Tổng tham số 110 triệu.
- BERTLARGE (L=24,H=1024,A=16): Tổng tham số 340 triệu.

Như vậy ở kiến trúc BERT Large chúng ta tăng gấp đôi số tầng, tăng kích thước ẩn của vector nhúng gấp 1.33 lần và tăng số lượng từ đầu trong multi-head layer gấp 1.33 lần

2.6. Mô hình ngôn ngữ PhoBERT

BERT là một mô hình ngôn ngữ mạnh mẽ được Google phát triển, nổi bật với khả năng hiểu ngữ nghĩa của văn bản nhờ vào việc sử dụng hai hướng tiếp cận: học từ trái sang phải và phải sang trái. Cho đến nay BERT vẫn được sử dụng cho nhiều bài toán NLP cho kết quả tốt với các phiên bản cải tiến, biến thể như RoBERTa, ALBERT, DistilBERT,... Tuy nhiên, huấn luyện mô hình BERT cho Tiếng Việt lại không hề đơn giản do đó rất khó để có thể áp dụng BERT cho các nhiệm vụ Tiếng Việt dù cho Google cũng có huấn luyện trước cho nhiều ngôn ngữ bao gồm cả tiếng Việt nhưng chưa cho kết quả thực hiện tốt nhất.

PhoBERT được phát triển bởi nhóm nghiên cứu tại VinAI Research, một trong những trung tâm nghiên cứu AI hàng đầu Việt Nam, thuộc Tập đoàn Vingroup. Đây là thành quả hợp tác giữa các nhà khoa học hàng đầu trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). PhoBERT ra mắt lần đầu vào năm 2020 và đã được công bố trong bài báo khoa học tại hội nghị Findings of the Association for Computational Linguistics: EMNLP 2020. PhoBERT được tạo ra với mục tiêu cung cấp một mô hình tiên tiến, có thể hiểu và xử lý ngữ nghĩa văn bản tiếng Việt một cách hiệu quả, phục vụ các ứng dụng

nghư phân loại văn bản, nhận diện thực thể (NER), dịch máy, tóm tắt văn bản, và nhiều tác vụ xử lý ngôn ngữ tự nhiên (NLP) khác.

Đây là một mô hình huấn luyện trước được huấn luyện cho chỉ huấn luyện dành riêng cho tiếng Việt. Tương tự như BERT, PhoBERT cũng có 2 phiên bản là:

- PhoBERTbase với 12 khối transformers.
- PhoBERTlarge với 24 khối transformers.

2.6.1. Các thành phần chính của PhoBert

PhoBERT dựa trên kiến trúc của BERT, sử dụng mạng Transformer với nhiều lớp mã hóa (encoder layers). Các thành phần chính của PhoBERT bao gồm:

- Embeddings: PhoBERT sử dụng embedding từ các từ trong ngữ cảnh tiếng Việt. Các từ được chuyển thành các vector nhúng (word embeddings), giúp mô hình có thể nhận diện và xử lý ngữ nghĩa của các từ trong văn bản.
- Transformer Encoder: PhoBERT sử dụng bộ mã hóa Transformer gồm nhiều lớp. Mỗi lớp bao gồm hai thành phần chính:
 - Self-Attention: Quá trình này giúp mô hình tập trung vào những phần quan trọng của văn bản, bất kể vị trí của từ trong câu. Mỗi từ trong câu có thể "chú ý" đến các từ khác trong câu để hiểu ngữ nghĩa tốt hơn.
 - Feed-forward Neural Networks: Sau khi thực hiện self-attention, đầu ra của mỗi lớp được đưa qua một mạng neural đơn giản để tạo ra các đặc trưng trừu tượng cho văn bản.
- Bidirectionality: Một trong những điểm đặc trưng quan trọng của BERT, và cũng là PhoBERT, là tính hai chiều (bidirectional). Điều này có nghĩa là mô hình không chỉ học ngữ nghĩa của một từ dựa trên các từ phía trước mà còn dựa trên các từ phía sau, giúp hiểu rõ hơn ngữ cảnh toàn bộ câu hoặc đoạn văn.
- Masked Language Model (MLM): Để huấn luyện PhoBERT, một phần các từ trong câu bị ẩn (masked) và mô hình phải dự đoán các từ bị ẩn này. Việc này giúp PhoBERT học được mối quan hệ giữa các từ trong ngữ cảnh rộng lớn hơn, thay vì chỉ học dựa trên các từ lân cận.
- Next Sentence Prediction (NSP): BERT và PhoBERT sử dụng nhiệm vụ dự đoán câu tiếp theo để học được mối quan hệ giữa các câu trong văn bản. Đây là một phương pháp giúp mô hình nắm bắt được ngữ cảnh dài hạn và các mối quan hệ ngữ nghĩa giữa các câu.

2.6.2. Cách thức hoạt động của PhoBERT

- Bước 1: Pre-training (Huấn luyện trước): Trong giai đoạn này, PhoBERT học từ một lượng lớn dữ liệu văn bản tiếng Việt không gán nhãn. Hai nhiệm vụ chính trong giai đoạn pre-training là:

- Masked Language Modeling (MLM): Một phần các từ trong văn bản bị ẩn và mô hình phải dự đoán lại các từ đó.
- Next Sentence Prediction (NSP): Dự đoán xem câu tiếp theo có liên quan đến câu hiện tại hay không.

- Bước 2: Fine-tuning (Tinh chỉnh): Sau khi hoàn tất giai đoạn pre-training, PhoBERT có thể được tinh chỉnh cho các tác vụ NLP cụ thể, chẳng hạn như phân loại văn bản, nhận diện thực thể (NER), hoặc phân tích cảm xúc. Giai đoạn này sử dụng dữ liệu gán nhãn và điều chỉnh các trọng số của mô hình để phù hợp với tác vụ cần giải quyết.

- Bước 3: Inference (Sử dụng mô hình): Khi PhoBERT đã được huấn luyện và tinh chỉnh, nó có thể được sử dụng để dự đoán hoặc phân tích các văn bản tiếng Việt mới. Dựa trên các đặc trưng đã học, mô hình sẽ đưa ra các dự đoán hoặc phân tích liên quan đến văn bản đầu vào.

2.7. Tiêu chí đánh giá mô hình

Đánh giá hiệu suất của mô hình, các số liệu cần được sử dụng là 'Accuracy', 'Precision', 'Recall', 'F1-score' và 'BERTScore':

- Accuracy (Độ chính xác toàn phần) là một trong những chỉ số quan trọng trong việc đánh giá hiệu suất của một mô hình phân loại. Nó đo lường tỷ lệ dự đoán chính xác so với tổng số dự đoán. Độ chính xác cho biết tỷ lệ phần trăm các dự đoán đúng trên tổng số tất cả các dự đoán mà mô hình thực hiện. Công thức tính Accuracy:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Trong đó:

- True Positives (TP): Số trường hợp mô hình dự đoán đúng là "Positive" (dương tính).
- True Negatives (TN): Số trường hợp mô hình dự đoán đúng là "Negative" (âm tính).
- False Positives (FP): Số trường hợp mô hình dự đoán sai là "Positive" nhưng thực tế là "Negative".
- False Negatives (FN): Số trường hợp mô hình dự đoán sai là "Negative" nhưng thực tế là "Positive".

- Precision (độ chính xác): là tỷ lệ “true positive” được dự đoán chính xác so với tổng thể dự đoán. Điểm chính xác dao động từ 0 đến 1, độ chính xác càng cao chỉ ra rằng hầu hết các đối tượng được phát hiện khớp với các đối tượng thực tế. Độ chính xác được xác định như sau:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

- Recall (thu hồi): là tỷ lệ “true positive” được dự đoán chính xác so với tổng thể dữ liệu thực tế. Recall dao động từ 0 đến 1, trong đó điểm số recall cao có nghĩa là hầu hết các đối tượng thực tế cơ bản đã được phát hiện. Recall được xác định như sau:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

- F1-Score: Để có thể kết hợp giữa Precision và Recall, chúng ta có thể tính điểm F-score. F1-score lớn khi cả 2 giá trị Precision và Recall đều lớn. Ngược lại, chỉ cần 1 giá trị nhỏ sẽ làm cho F1-Score nhỏ.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- BERTScore sử dụng các vector nhúng từ mô hình BERT để so sánh mức độ tương đồng ngữ nghĩa giữa các từ trong bản dịch của mô hình và bản tham chiếu. Thay vì chỉ đơn thuần dựa vào độ trùng khớp của các n-gram như BLEU, BERTScore đánh giá mức độ tương đồng ngữ nghĩa và có thể nhận diện các từ có nghĩa tương tự nhưng không trùng khớp chính xác. BERTScore sử dụng cosine similarity (tương đồng cosine) giữa các vector nhúng của các từ trong bản dịch máy và bản tham chiếu để tính toán sự tương đồng ngữ nghĩa.

Công thức tính BERTScore:

$$\text{Precision}_i = \frac{\sum_{j \in \text{Reference}} \text{sim}(h_i, r_j)}{\text{len}(h_i)}$$

$$\text{Recall}_i = \frac{\sum_{j \in \text{Hypothesis}} \text{sim}(h_i, r_j)}{\text{len}(r_j)}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Trong đó:

- h_i là từ thứ i trong **bản dịch máy** (hypothesis).
- r_j là từ thứ j trong **bản tham chiếu** (reference).
- $\text{sim}(h_i, r_j)$ là **cosine similarity** giữa vector nhúng của từ h_i và r_j , tính toán mức độ tương đồng ngữ nghĩa giữa chúng.
- len là độ dài của bản dịch máy và bản tham chiếu.

Chương 3: PHƯƠNG PHÁP THỰC HIỆN

3.1. Mô tả bài toán

Bài toán được đề xuất là xây dựng một hệ thống chatbot hỗ trợ tư vấn tuyển sinh cho Trường Đại học Công nghệ Kỹ thuật Cần Thơ (CTUET). Hệ thống này nhằm cung cấp thông tin một cách nhanh chóng, chính xác, và phù hợp với nhu cầu cá nhân của thí sinh và phụ huynh. Chatbot sẽ hoạt động như một công cụ hỗ trợ hiệu quả, giảm tải công việc cho bộ phận tư vấn tuyển sinh, đồng thời nâng cao trải nghiệm của người dùng khi tiếp cận thông tin. Cụ thể, chatbot được thiết kế để giải quyết các vấn đề sau:

- Trả lời các câu hỏi thường gặp về thông tin tuyển sinh:
 - Cung cấp chi tiết về các ngành học đang được đào tạo tại trường, bao gồm chương trình học, nội dung đào tạo và triển vọng nghề nghiệp.
 - Trả lời các yêu cầu về điều kiện tuyển sinh như điểm chuẩn, các tổ hợp môn xét tuyển và các tiêu chí đặc biệt (nếu có).
 - Hỗ trợ giải đáp thông tin liên quan đến học phí, các chính sách học bổng và hỗ trợ tài chính dành cho sinh viên.
- Hướng dẫn quy trình đăng ký và xác nhận nhập học:
 - Cung cấp hướng dẫn cụ thể về cách nộp hồ sơ đăng ký xét tuyển, từ hình thức trực tuyến đến trực tiếp.
 - Hỗ trợ thông tin chi tiết về các mốc thời gian quan trọng như hạn nộp hồ sơ, thời gian công bố kết quả và thời gian xác nhận nhập học.
 - Hướng dẫn các bước xác nhận nhập học, bao gồm việc chuẩn bị các giấy tờ cần thiết và các thao tác trên hệ thống đăng ký của trường.

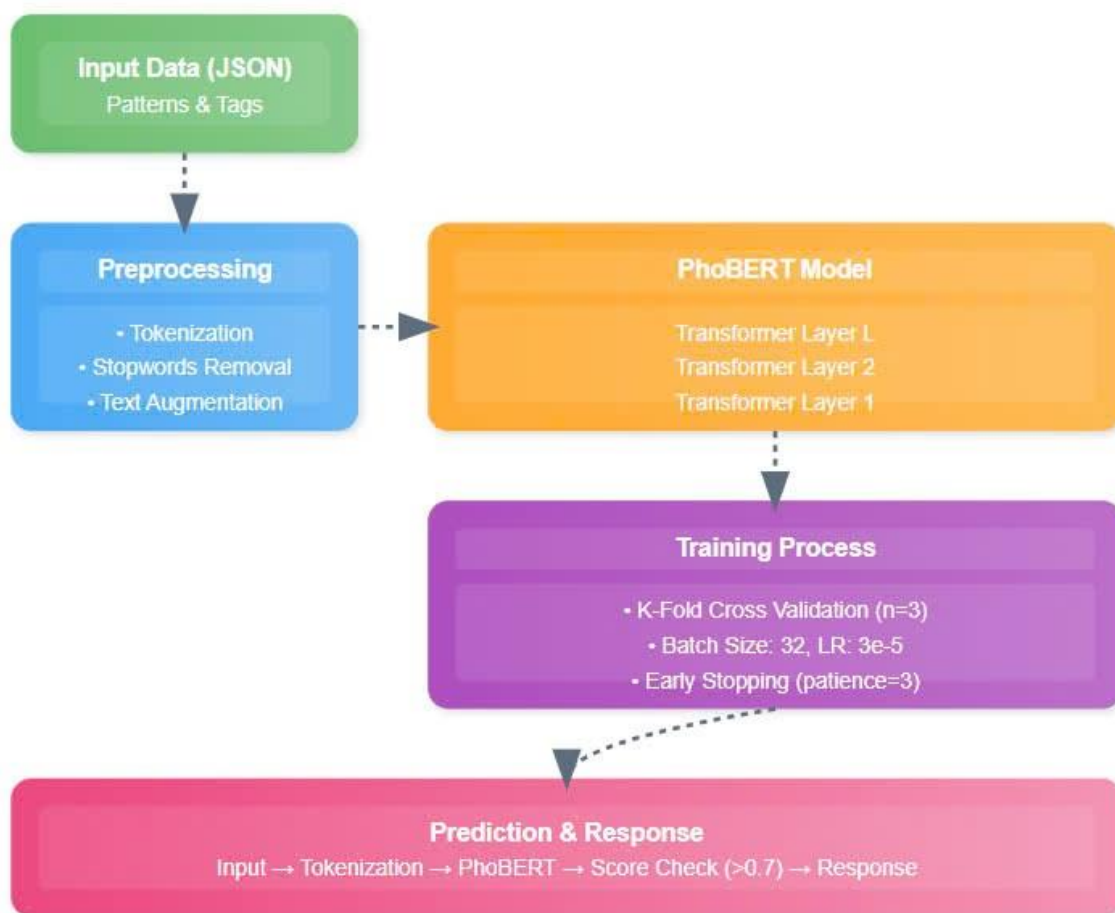
3.2. Tập dữ liệu

Xây dựng một ứng dụng chatbot tự động việc quan trọng nhất là giai đoạn chuẩn bị dữ liệu. Tập dữ liệu trong đề tài gồm hai phần chính:

- Dữ liệu tuyển sinh: Thu thập từ thông tin tuyển sinh của trường gồm có các thông tin về ngành học, chương trình đào tạo, điều kiện tuyển sinh, học phí, học bổng, quy trình đăng ký,... Sau đó chúng tôi xử lý dữ liệu đã thu thập, chuyển đổi thành dữ liệu có cấu trúc và tạo ra khoảng 900 cặp câu hỏi và trả lời. Ngoài ra, các câu hỏi được phân loại thành 50 tag khác nhau.
- Dữ liệu ngôn ngữ: Tập dữ liệu gồm khoảng 100 hội thoại bằng tiếng Việt được thu thập từ nhiều nguồn. Sau khi chọn lọc chúng tôi có khoảng 100 cặp câu hỏi và câu trả lời từ nguồn này.

Tiếng Việt	Tiếng Anh	Viết tắt
Khoa học máy tính	Computer Science	CS
Khoa học dữ liệu	Data Science	DS
Hệ thống thông tin	Information Systems	IS
Công nghệ thông tin	Information Technology	IT
Kỹ thuật phần mềm	Software Engineering	SE
Kỹ thuật hệ thống công nghiệp	Industrial Systems Engineering	ISE
Quản lý công nghiệp	Industrial Management	IM
Logistics và quản lý chuỗi cung ứng	Logistics and Supply Chain Management	LSCM
Quản lý xây dựng	Construction Management	CM
Công nghệ kỹ thuật công trình xây dựng	Civil Engineering Technology	CET
Công nghệ kỹ thuật cơ điện tử	Mechatronics Engineering Technology	MET
Công nghệ kỹ thuật điện, điện tử	Electrical and Electronics Engineering Technology	EET
Công nghệ kỹ thuật điều khiển và tự động hóa	Control and Automation Engineering Technology	CAET
Công nghệ thực phẩm	Food Technology	FT
Công nghệ sinh học	Biotechnology	BT
Công nghệ kỹ thuật năng lượng	Energy Engineering Technology	EET
Quản trị kinh doanh	Business Administration	BAdministration

3.3. Mô hình tổng quát



Hình 3. 2 Mô hình tổng quát

Trong bài toán này, chúng tôi thực hiện việc huấn luyện hai mô hình khác nhau để so sánh hiệu suất và khả năng xử lý của chúng trong ứng dụng chatbot tuyển sinh. Mặc dù chúng tôi sử dụng cả hai mô hình PhoBERT và BERT. Tuy nhiên, do chỉ thay đổi model giữa PhoBERT và BERT nên ở đây chúng tôi chỉ miêu tả cụ thể một mô hình để làm đại diện.

3.3.1. Dữ liệu đầu vào (JSON) – input Data (JSON): Patterns & Tags

- Dữ liệu huấn luyện bao gồm các câu hỏi hoặc câu văn bản (patterns) và nhãn tương ứng (tags).

Ví dụ: Trong bài toán chatbot tuyển sinh, các pattern có thể là các câu hỏi như "Điểm chuẩn ngành CNTT là bao nhiêu?" và tag tương ứng là "Thông tin điểm chuẩn".

- Định dạng JSON là tiêu chuẩn để lưu trữ và trao đổi dữ liệu có cấu trúc, hỗ trợ các bước xử lý tự động.

3.3.2. Tiền xử lý dữ liệu - Preprocessing

- Tokenization: Quá trình chia nhỏ câu thành các token (từ hoặc ký tự). PhoBERT sử dụng bộ tokenizer riêng, phù hợp với ngữ pháp và đặc điểm tiếng Việt.

- Loại bỏ stop words: Stop words là các từ không mang ý nghĩa quan trọng như "là", "của", "và". Loại bỏ chúng giúp giảm nhiễu và tăng hiệu quả mô hình.
- Tăng cường dữ liệu (Data Augmentation): Kỹ thuật như thay thế từ đồng nghĩa, hoán đổi vị trí từ trong câu, giúp mô hình học được nhiều đặc điểm hơn và tăng khả năng khái quát hóa.

3.3.3. Mô hình PhoBERT – PhoBERT Model

Mỗi layer này đều có cấu trúc gồm:

- Multi-head Self-attention: cho phép mô hình tập trung vào các phần khác nhau của câu
- Feed-forward Neural Network: xử lý thông tin từ self-attention
- Layer Normalization: chuẩn hóa dữ liệu
- Residual connections: giúp thông tin flow tốt hơn qua các layer

3.3.3.1. Transformer Layer L

- Layer cuối cùng (Last layer) của mô hình
- Đảm nhiệm việc tổng hợp thông tin từ các layer trước đó
- Thường được sử dụng để trích xuất đặc trưng cuối cùng (final features)

3.3.3.2. Transformer Layer 1:

- Layer đầu tiên của mô hình
- Xử lý trực tiếp từ embedding đầu vào
- Học các đặc trưng cơ bản của từ và cụm từ
- Nắm bắt các thông tin ngữ cảnh đơn giản

3.3.3.3. Transformer Layer 2:

- Layer trung gian
- Xử lý thông tin từ Layer 1
- Học các đặc trưng phức tạp hơn
- Bắt đầu hiểu các mối quan hệ phức tạp giữa các từ

3.3.4. Quá trình huấn luyện – Training Process

- K-fold cross-validation: Dữ liệu được chia thành K phần để huấn luyện và đánh giá. Kỹ thuật này đảm bảo mô hình được đánh giá trên toàn bộ dữ liệu mà không bị lệch.
- Siêu tham số (Hyperparameters):

- Batch size: Số mẫu dữ liệu xử lý trong mỗi lần cập nhật trọng số.
 - Learning rate: Tốc độ học, quyết định bước nhảy trong không gian tham số.
 - Các siêu tham số này cần được tối ưu hóa để đạt hiệu quả tốt nhất.
- Early stopping: Dừng quá trình huấn luyện khi độ chính xác hoặc hàm mất mát trên tập kiểm định không cải thiện sau một số vòng lặp nhất định.

3.3.5. Dự đoán và trả lời – Prediction & Response

- Tokenization đầu vào mới: Câu hỏi mới từ người dùng được tiền xử lý tương tự như trong bước huấn luyện.
- Dự đoán nhãn: Mô hình PhoBERT sẽ dự đoán tag tương ứng cho câu đầu vào.
- Kiểm tra điểm số: Điểm xác suất của nhãn dự đoán sẽ được kiểm tra. Nếu đạt ngưỡng (ví dụ 0.7), nhãn dự đoán được chấp nhận.
- Sinh câu trả lời: Tag dự đoán được ánh xạ tới câu trả lời tương ứng trong cơ sở dữ liệu chatbot.

3.4. Các giai đoạn giải quyết


3.4.1. Tiền xử lý dữ liệu

Sau khi kết nối dữ liệu với mô hình, bước đầu tiên chúng tôi thực hiện kiểm tra kỹ lưỡng bộ dữ liệu đầu vào để đảm bảo dữ liệu không chứa giá trị null. Việc này rất quan trọng nhằm loại bỏ những giá trị thiếu hoặc không hợp lệ trong dữ liệu, điều này không chỉ giúp đảm bảo tính toàn vẹn và chất lượng của bộ dữ liệu mà còn ngăn chặn các lỗi có thể xảy ra trong quá trình huấn luyện mô hình. Khi dữ liệu được xử lý sạch sẽ, các bước xử lý tiếp theo sẽ trở nên hiệu quả hơn, từ đó cải thiện độ chính xác của mô hình. Việc này giúp tối ưu hóa quá trình huấn luyện, giảm thiểu sự can thiệp của dữ liệu nhiễu và giúp mô hình học được những đặc trưng chính xác từ dữ liệu. Mỗi bước kiểm tra và xử lý dữ liệu kỹ càng đóng vai trò quan trọng trong việc đạt được kết quả tối ưu cho các bước huấn luyện và đánh giá mô hình tiếp theo.

```
Pattern      0
Tag          0
dtype: int64
```

Sau khi kiểm tra và làm sạch dữ liệu, chúng tôi tiếp tục thực hiện quá trình trực quan hóa các tag dữ liệu. Việc này giúp chúng tôi có cái nhìn trực quan về sự phân bố và tần suất xuất hiện của các tag trong bộ dữ liệu, từ đó có thể dễ dàng quản lý và phân tích các đặc trưng của dữ liệu.



Chatbot Tags distribution

Tags	Frequency
nature	16
adMajor	15
adZone	15
adLine	15
adini	15
sbFaculty	15
edu	15
adMaster	14
adLine	14
salutation	14
goodbye	13
greeting	13
name	13
sbCET	13
quality	13
intro	13
creator	13
decision	13
introduceFB	13
introduceCAET	13
adLine	13
introduceSE	13
knowIT	13
introduceSE	13
introduceMET	13
introduceIT	13
obCAET	13
knowCAET	13
obFB	13
knowMET	13
introduceAdministration	13
knowBAdministration	13
introduceEL	13
knowEL	13
knowFB	13
ctLuat	13
thucTapLuat	13
introduceDS	13
knowLaw	13
adLine	13
tuition	13
scholarship	13
dorm	13
eGra	13
job	13
sb	13
sbSE	13
cdKTPM	13
sbEET	13
sbCM	13
sbLSCM	13
obMET	13
obEL	13
obBAdministration	13
AP	13
SBEEET	13
cdFB	13
applied	13
knowSE	13
knowSE	13
obSE	13
jobDS	13
sbT	13
sbT	13
obDS	13
knowDS	13
ctCNTT	13
introduceLSCM	13
obLSCM	13
knowLSCM	13
knowLSCM	13
sbBAdministration	13
sbCAET	13
obIT	13
sbMET	13
knowCS	13
introduceCS	13
sbEL	13
sbAC	13
obCS	13
jobCS	13
sbIM	13
sbLaw	13
LkeIT	13

Hình 3. 5 Biểu đồ trực quan các tag

Dựa vào kết quả của biểu đồ, chúng tôi nhận thấy rằng tần suất xuất hiện của các tag có sự chênh lệch lớn. Cụ thể, tag “*nature*” có tần suất xuất hiện cao nhất là 4,5%, trong khi tag “*LikeIT*” có tần suất thấp nhất, chỉ khoảng 0,3%. Sự phân bố không đồng đều này có thể gây ra hiện tượng class imbalance trong mô hình, làm ảnh hưởng đến độ chính xác và hiệu quả huấn luyện. Các tag có tần suất thấp dưới 1% như “*LikeIT*” có thể không được mô hình nhận diện đầy đủ, dẫn đến hiệu suất không tốt đối với những tag này. Để khắc phục vấn đề này, một phương pháp tiềm năng là tăng sinh dữ liệu (data augmentation). Bằng cách áp dụng các kỹ thuật tăng cường dữ liệu, chúng tôi có thể tạo ra các mẫu dữ liệu mới cho các tag ít xuất hiện, giúp cân bằng tần suất giữa các tag và cải thiện khả năng nhận diện của mô hình đối với các tag hiếm gặp. Điều này sẽ giúp mô hình học được đặc trưng của tất cả các tag một cách công bằng và chính xác hơn.

Để xử lý các từ dư thừa và giảm nhiễu trong dữ liệu, chúng tôi sử dụng thuật toán Stopwords để loại bỏ các từ không mang ý nghĩa quan trọng trong ngữ cảnh, giúp mô hình tập trung vào các từ khóa cần thiết. Các từ này bao gồm các từ hỏi, các từ nối, và những từ mang tính chức năng như “là”, “của”, “và”, “nhưng”,...

```
stemmer = PorterStemmer()
stopwords = [
    # Các từ phổ biến trong câu hỏi
    "là", "của", "và", "có", "cho", "đây", "rằng", "nhưng", "tôi",
    "bạn", "với", "về", "thì", "lại", "này", "một", "nhiều", "nào",

    # Từ liên quan đến giáo dục/trường học
    "trường", "ngành", "học", "sinh viên", "đại học",
    "tuyển sinh", "đào tạo", "chương trình", "khoa",
    "năm", "hệ", "môn", "điểm", "thông tin", "giờ",

    # Cấu trúc câu hỏi thường gặp
    "như thế nào", "ra sao", "thế nào", "thì sao", "là gì",
    "ở đâu", "bao nhiêu", "khi nào", "làm sao", "bao giờ",

    # Từ để hỏi
    "ai", "gì", "nào", "đâu", "sao",

    # Các từ nối
    "mà", "các", "những", "từ", "tại", "theo", "sau", "sẽ", "vào",
    "do", "như", "hay", "còn", "bởi", "vì", "mình", "đến", "cũng",

    # Từ chỉ thời gian
    "hiện tại", "hiện nay", "đang", "trong", "nay", "khi",

    # Từ chỉ định
    "này", "đó", "kia", "ấy"
```

```

]
# Ký tự không cần thiết
ignore_words = [ '?', '!', ',', '.', ':', ';', '-', '"',
                  "'", '"', '(', ')', '[', ']', '/', '\\',
                  '+', '=', '@', '#', '$', '%', '^', '&', '*' ]
# Hàm tiền xử lý
def preprocess_pattern(pattern, stopwords):
    words = word_tokenize(pattern.lower())
    filtered_words = [
        stemmer.stem(word) for word in words
        if word not in ignore_words and word not in stopwords
    ]
    return " ".join(filtered_words)
# Áp dụng tiền xử lý
df['Pattern'] = df['Pattern'].apply(lambda x: preprocess_pattern(x, stopwords))

```

Hình 3. 6 Hàm thực hiện loại bỏ dữ liệu dư thừa

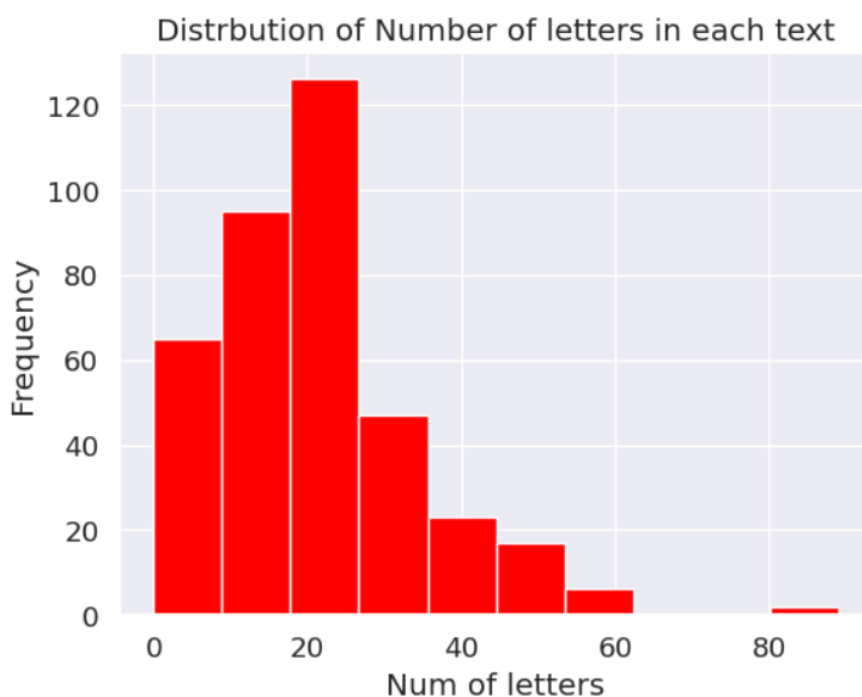
Thông qua việc sử dụng stopwords, chúng tôi loại bỏ các từ không mang giá trị thông tin quan trọng, như các từ hỏi hay từ chức năng, và chuẩn hóa các từ gốc, giúp giảm nhiễu trong dữ liệu. Điều này làm cho mô hình dễ dàng tập trung vào các từ khóa chính, cải thiện hiệu quả và độ chính xác trong các tác vụ phân tích và dự đoán.

Sau khi hoàn tất quá trình tiền xử lý và loại bỏ các dữ liệu dư thừa, chúng tôi tiếp tục sử dụng biểu đồ từ khóa dạng đám mây (Word Cloud) để trực quan hóa các từ khóa phổ biến trong bộ dữ liệu. Mục đích của việc này là để kiểm tra xem bộ dữ liệu có đáp ứng được yêu cầu của bài toán tuyển sinh và có thể áp dụng hiệu quả trong hệ thống chatbot hỗ trợ tư vấn tuyển sinh hay không. Biểu đồ từ khóa dạng đám mây giúp chúng tôi dễ dàng nhận diện và đánh giá sự phân bố của các từ khóa quan trọng trong nội dung dữ liệu. Mỗi từ khóa xuất hiện trong biểu đồ có kích thước tỷ lệ thuận với tần suất xuất hiện của nó trong dữ liệu. Các từ khóa lớn và nổi bật thể hiện những thông tin quan trọng và phổ biến nhất, trong khi các từ nhỏ hơn có thể cho thấy các thông tin ít xuất hiện hoặc ít quan trọng hơn.



Hình 3. 7 Biểu đồ từ khóa dạng đám mây (Word Cloud)

- Từ khóa lớn nhất: “Xét tuyển,” “Tổ hợp,” “Công nghệ,” “Đào tạo”: Đây là những từ khóa xuất hiện với tần suất cao nhất, phản ánh nội dung chính của dữ liệu liên quan đến tuyển sinh, bao gồm các thông tin về hình thức xét tuyển, tổ hợp môn, ngành công nghệ, và các chương trình đào tạo.
- Các từ khóa liên quan: “Kỹ thuật,” “Sinh viên,” “Tuyển sinh,” “Chỉ tiêu,” “Đại học”: Những từ khóa này thể hiện mối quan tâm về các ngành học cụ thể, số lượng tuyển sinh, và quy trình dành cho sinh viên. “Luật,” “Tài chính,” “Quản trị,” “Ngôn ngữ”: Đề cập đến các ngành học phổ biến hoặc được quan tâm trong dữ liệu tuyển sinh.



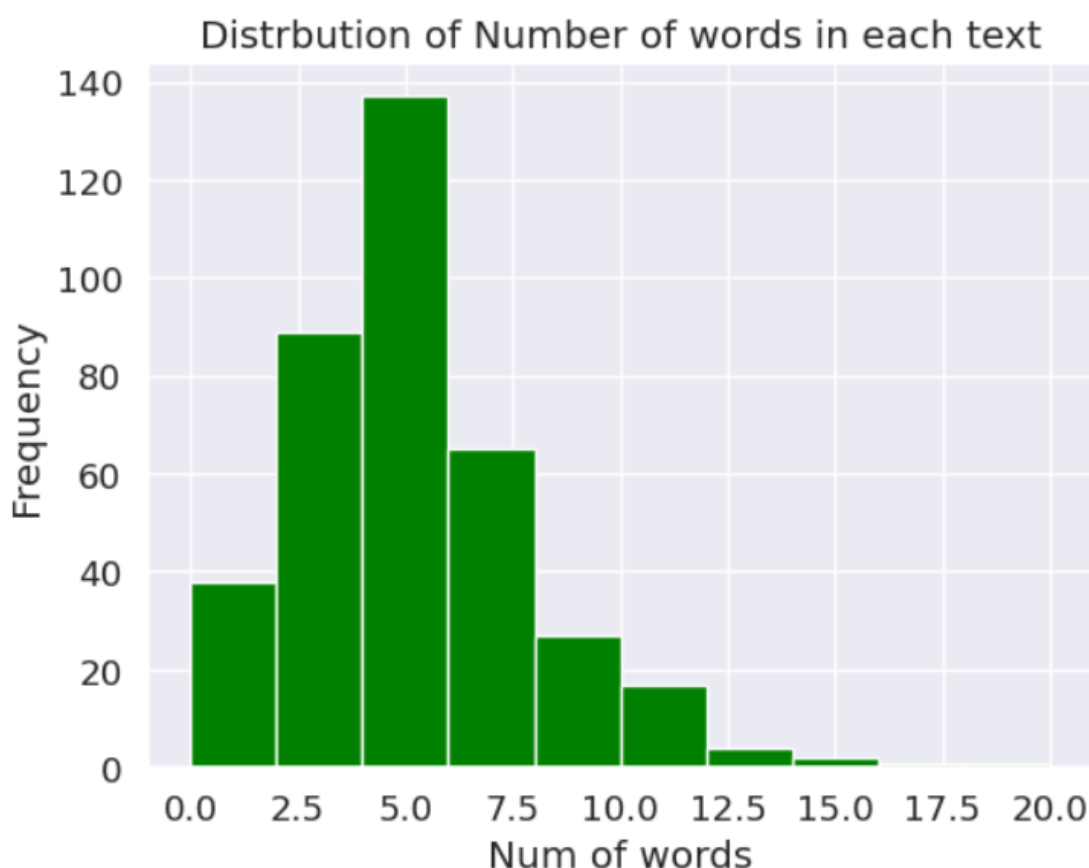
Hình 3. 8 Biểu đồ phân phối tần suất (Frequency Distribution Histogram)

Biểu đồ phân bố số lượng chữ cái trong mỗi đoạn văn này minh họa sự phân bố tần suất xuất hiện của các đoạn văn bản (bao gồm cả câu hỏi và câu trả lời) dựa trên số lượng ký tự trong từng đoạn. Mỗi cột thể hiện số lượng đoạn văn bản có số ký tự trong một phạm vi nhất định, từ đó cho phép chúng ta đánh giá độ dài trung bình của các đoạn văn trong bộ dữ liệu.

- Phần lớn các đoạn văn bản có từ 10 đến 30 ký tự: Biểu đồ cho thấy rằng phần lớn các câu hỏi và câu trả lời trong bộ dữ liệu có độ dài vừa phải, dao động từ 10 đến 30 ký tự. Điều này phản ánh rằng các đoạn văn bản trong hệ thống chatbot thường ngắn gọn và dễ hiểu, phù hợp với phong cách giao tiếp trong môi trường tuyển sinh.

- Đỉnh tần suất ở khoảng 20 ký tự: Đoạn văn bản có khoảng 20 ký tự là phổ biến nhất, cho thấy rằng các câu hỏi và câu trả lời được hình thành chủ yếu ở độ dài này. Điều này là hợp lý vì các câu hỏi về tuyển sinh thường ngắn gọn, trực tiếp, và không cần thiết phải dài dòng. Câu trả lời từ hệ thống chatbot cũng cần phải ngắn gọn, dễ hiểu và cung cấp thông tin cụ thể.

- Một số ít câu rất ngắn (< 10 ký tự) hoặc rất dài (> 60 ký tự): Mặc dù phần lớn các câu hỏi và câu trả lời có độ dài trung bình, một số ít câu lại có độ dài rất ngắn (< 10 ký tự) hoặc rất dài (> 60 ký tự). Các câu rất ngắn có thể là những câu hỏi hoặc câu trả lời đơn giản, dễ trả lời. Các câu dài hơn có thể chứa các câu hỏi phức tạp hoặc yêu cầu cung cấp nhiều thông tin chi tiết hơn, ví dụ như các câu hỏi về yêu cầu xét tuyển cho nhiều tổ hợp môn hoặc các chương trình học đặc biệt.

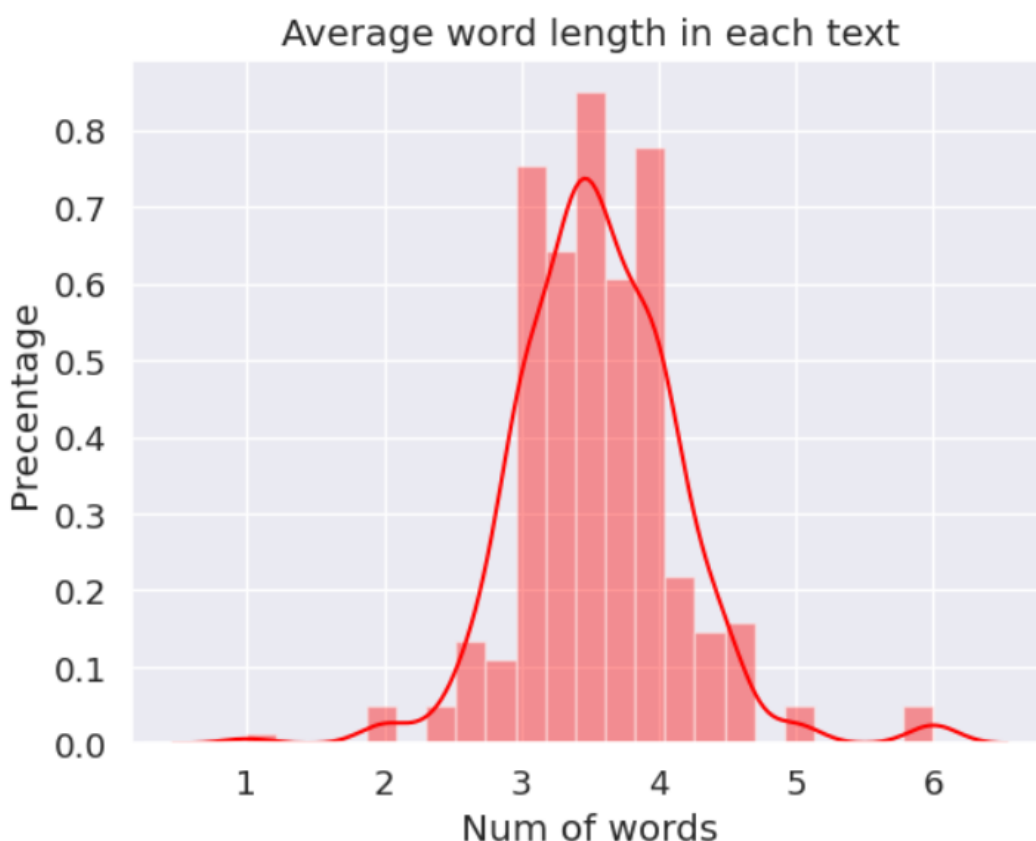


Hình 3. 9 Biểu đồ phân bố số lượng từ trong mỗi đoạn văn bản

Biểu đồ cho thấy rằng phần lớn các câu văn bản chứa từ 3 đến 6 từ, với đỉnh phân bố nằm ở khoảng 5 từ. Điều này cho thấy rằng các câu trong bộ dữ liệu này thường ngắn gọn, súc tích và không quá phức tạp, phản ánh đúng phong cách giao tiếp thường thấy trong môi trường tuyển sinh. Các câu này phù hợp với yêu cầu của một hệ thống chatbot, nơi mà các câu hỏi cần được hiểu rõ ràng và trả lời nhanh chóng.

Mặc dù phần lớn các câu có độ dài từ 3 đến 6 từ, một số ít câu lại có độ dài ngắn (< 3 từ) hoặc dài (> 10 từ). Tuy nhiên, những câu này là trường hợp hiếm gặp, cho thấy tính nhất quán trong phong cách viết của bộ dữ liệu. Các câu ngắn hơn có thể là những câu hỏi đơn giản, trong khi các câu dài hơn có thể chứa các câu hỏi yêu cầu thông tin chi tiết, ví dụ như yêu cầu về các tổ hợp môn hoặc các chương trình đào tạo.

Từ biểu đồ này, có thể nhận thấy rằng bộ dữ liệu này khá phù hợp với bài toán trợ lý ảo. Dữ liệu không chỉ bao gồm các câu hỏi chi tiết, mà còn có các câu hỏi ngắn gọn, giúp hệ thống chatbot có thể xử lý hiệu quả cả các yêu cầu đơn giản lẫn phức tạp. Điều này cho thấy rằng dữ liệu đã được thiết kế để phục vụ cho một môi trường ứng dụng mà yêu cầu độ linh hoạt và sự chính xác trong việc trả lời câu hỏi của người dùng.



Hình 3. 10 Biểu đồ trực quan độ dài trung bình của các từ

Đây là một biểu đồ kết hợp giữa tần suất (histogram) và đường mật độ xác suất (density plot), giúp trực quan hóa độ dài trung bình của các từ trong bộ dữ liệu. Biểu đồ cho thấy rằng hầu hết các văn bản trong tập dữ liệu có số lượng từ trung bình dao động trong khoảng từ 3 đến 4 từ, cho thấy sự đồng đều và nhất quán trong cấu trúc câu hỏi và câu trả lời. Sự phân bố của dữ liệu là đối xứng và gần với dạng phân phối chuẩn, điều này chứng tỏ rằng độ dài của các câu trong bộ dữ liệu không có sự chênh lệch quá lớn. Điều này có thể phản ánh rằng các câu hỏi và câu trả lời trong bài toán trợ lý ảo đều có độ dài hợp lý, không quá ngắn cũng không quá dài, giúp hệ thống dễ dàng xử lý và phản hồi chính xác.

Ở giai đoạn trực quan hóa các tag dữ liệu, chúng tôi nhận thấy sự không đồng đều trong tần suất xuất hiện của các tag trong bộ dữ liệu. Điều này có thể gây ra các vấn đề về hiệu quả và độ chính xác của mô hình học máy, đặc biệt là khi một số tag xuất hiện quá ít trong dữ liệu. Để giải quyết vấn đề này, chúng tôi đã áp dụng kỹ thuật tăng cường dữ liệu (Text Augmentation) nhằm tăng cường sự đa dạng và cân đối của dữ liệu.

```
def augment_text(text):
    # Một số kỹ thuật augmentation đơn giản cho tiếng Việt
    augmented = []
    words = text.split()

    # 1. Xóa ngẫu nhiên một số từ
    if len(words) > 3:
        remove_idx = random.randint(0, len(words)-1)
        augmented.append(' '.join(words[:remove_idx] + words[remove_idx+1:]))

    # 2. Hoán đổi vị trí các từ lân cận
    if len(words) > 3:
        idx = random.randint(0, len(words)-2)
        words[idx], words[idx+1] = words[idx+1], words[idx]
        augmented.append(' '.join(words))

    return augmented
```

Hình 3. 11 Tăng cường dữ liệu

Trong quá trình tiền xử lý, chúng tôi cũng đã sử dụng phương pháp stopwords để loại bỏ những từ không mang nhiều ý nghĩa trong ngữ cảnh câu hỏi, giúp giảm thiểu nhiễu và tập trung vào các từ khóa quan trọng. Việc loại bỏ các stopwords giúp mô hình học được những đặc trưng thực sự quan trọng, từ đó cải thiện hiệu suất khi xử lý và phân loại các câu hỏi trong hệ thống chatbot. Các biểu đồ trực quan hóa được kết hợp để đánh giá và hiểu rõ hơn về bộ dữ liệu. Các biểu đồ này không chỉ giúp nhận diện sự mất cân bằng giữa các tag mà còn cung cấp cái nhìn trực quan về các vấn đề trong dữ liệu, chẳng hạn như sự xuất hiện quá ít hoặc quá nhiều của một số tag. Sau khi phát hiện được những vấn đề này, chúng tôi đã áp dụng giải pháp Text Augmentation, một kỹ thuật tăng cường dữ liệu mạnh mẽ, giúp tạo ra dữ liệu mới từ dữ liệu gốc. Điều này giúp tăng tính phong phú và đa dạng cho bộ dữ liệu, đồng thời cải thiện sự cân đối giữa các tag, đặc biệt trong những trường hợp dữ liệu không đều. Việc áp dụng Text Augmentation không chỉ giúp nâng cao chất lượng dữ liệu mà còn làm tăng độ chính xác trong huấn luyện mô hình. Khi kết hợp với việc loại bỏ stopwords, quá trình tiền xử lý đã giúp làm sạch dữ liệu, loại bỏ các yếu tố dư thừa và tối ưu hóa dữ liệu cho mô hình. Để đảm bảo chất lượng dữ liệu đầu vào, chúng tôi đã trực quan hóa sơ đồ dữ liệu, qua đó xác định rằng dữ liệu đầu vào hoàn toàn phù hợp với mục tiêu và bối cảnh của bài toán trợ lý ảo tư vấn tuyển sinh.

3.4.2. Huấn luyện mô hình

Sau khi đã trải qua quá trình tiền xử lý dữ liệu, nhằm để có thể so sánh trực quan giữa hai mô hình PhoBERT và BERT trong bài toán, chúng tôi sẽ thực hiện huấn luyện cả hai mô hình trên cùng bộ tham số và siêu tham số.

```
training_args = TrainingArguments(  
    output_dir='./output',  
    num_train_epochs=150,  
    per_device_train_batch_size=32,  
    per_device_eval_batch_size=16,  
    learning_rate=3e-5,  
    weight_decay=0.01,  
    warmup_ratio=0.1,  
    logging_steps=10,  
    evaluation_strategy="steps",  
    eval_steps=50,  
    save_strategy="steps",  
    load_best_model_at_end=True,  
    metric_for_best_model="eval_loss",  
    greater_is_better=False,  
)
```

Hình 3. 12 Thông số các tham số và siêu tham số

Chương 4: KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ GIẢI PHÁP

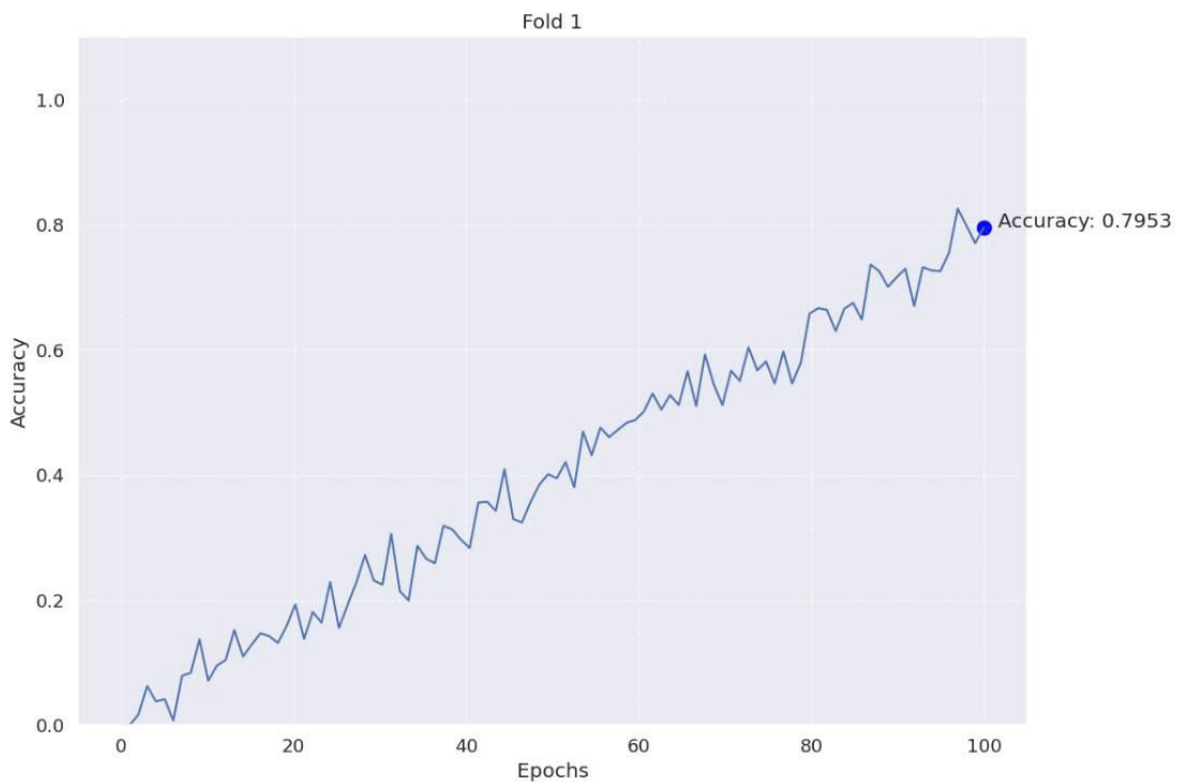
4.1. Kết quả thực nghiệm

4.1.1. Kết quả huấn luyện qua từng K-fold

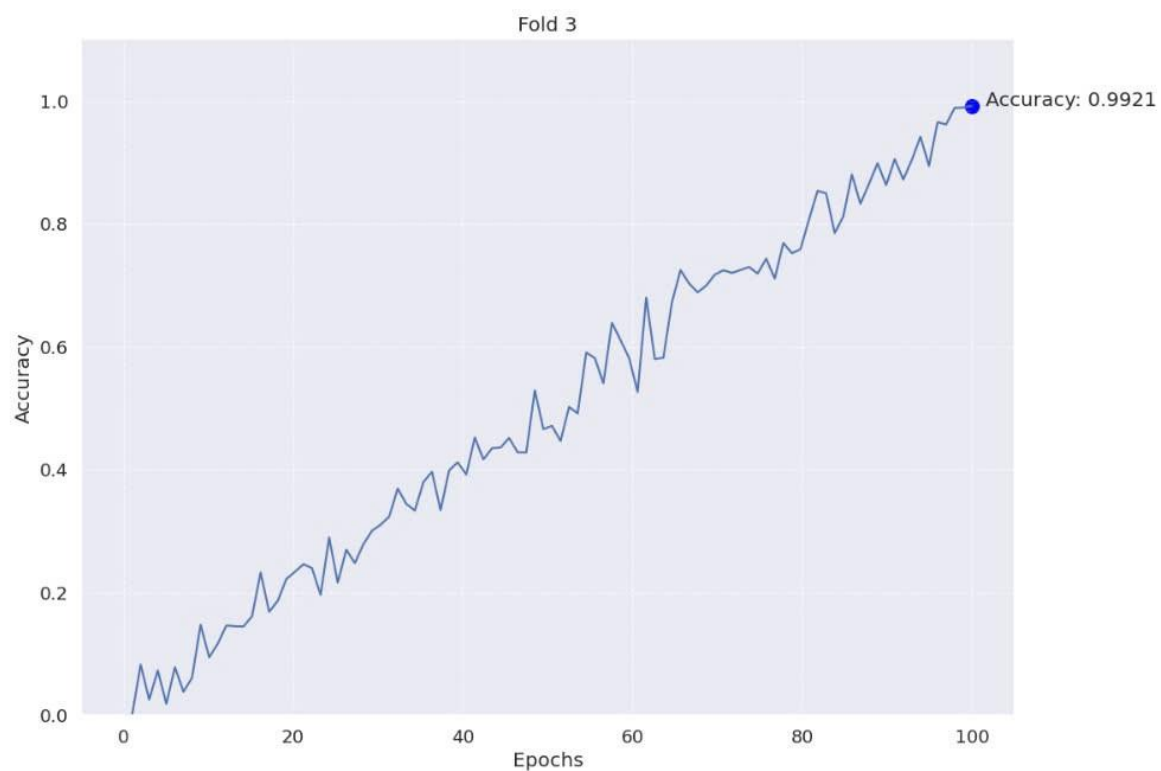
Tại bài toán này chúng tôi lựa chọn cách K-fold cross-validation để thực hiện việc huấn luyện dữ liệu hay vì sử dụng Trainer của Transformer. Do dữ liệu chúng tôi tuy trải qua việc tăng cường dữ liệu nhưng vẫn còn chưa ổn định và mong muốn có thể tận dụng việc huấn luyện qua nhiều fold để tăng tính ổn định của mô hình.

Khi huấn luyện với mô hình K-fold cross-validation, chúng tôi thực hiện kiểm thử với $k=5$ để có thể lựa chọn k phù hợp, nhận thấy tại khoảng $k = 4$ và $k = 5$ mô hình PhoBERT đã bắt đầu xuất hiện việc overfitting còn đối mô hình BERT đã bắt đầu tại $k = 3$. Vì vậy chúng tôi đã quyết sẽ thực hiện việc cả hai mô hình tại $k=3$.

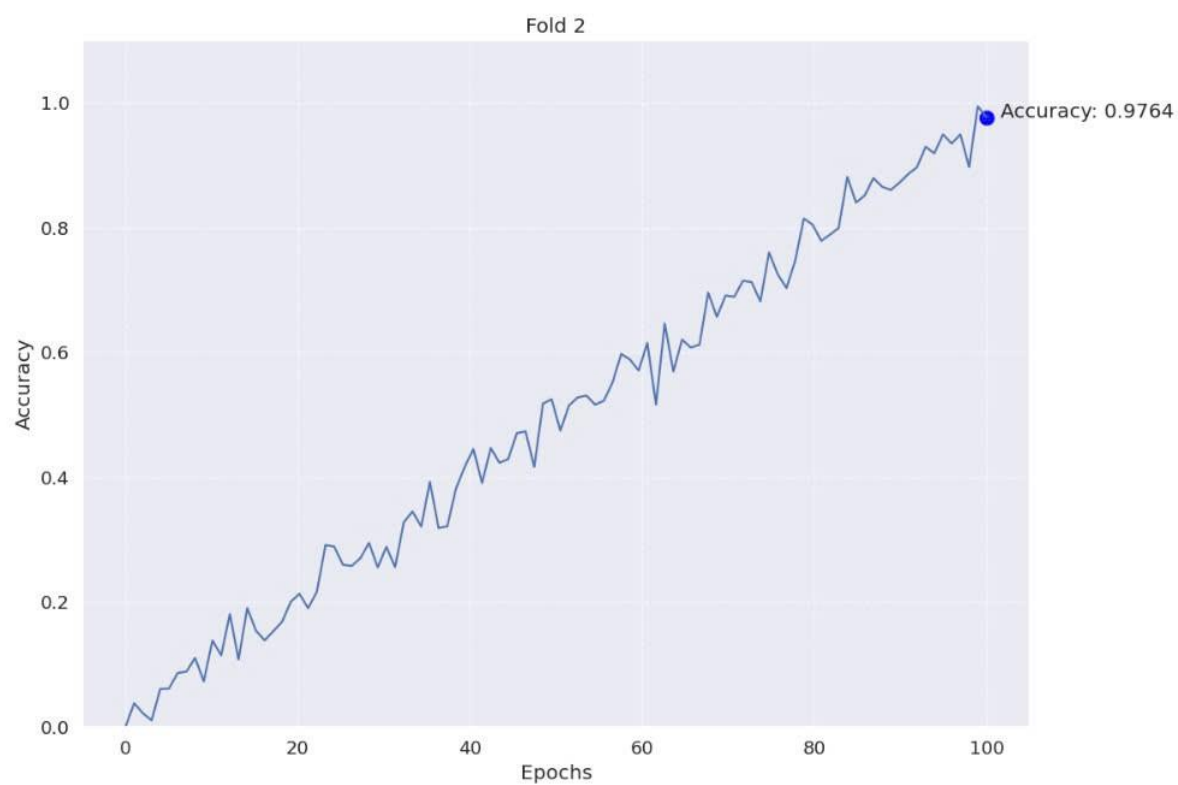
4.1.1.1. Mô hình ngôn ngữ PhoBERT



Hình 4. 1 $k = 1$ (PhoBERT)

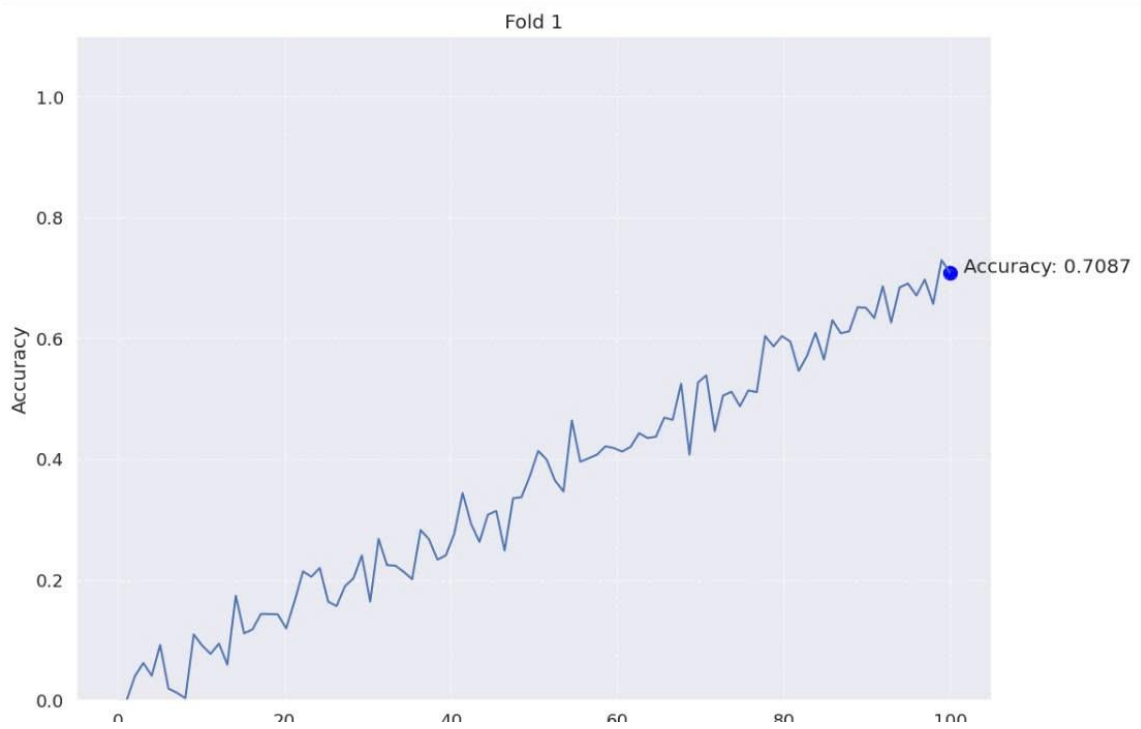


Hình 4. 2 $k = 2$ (PhoBERT)

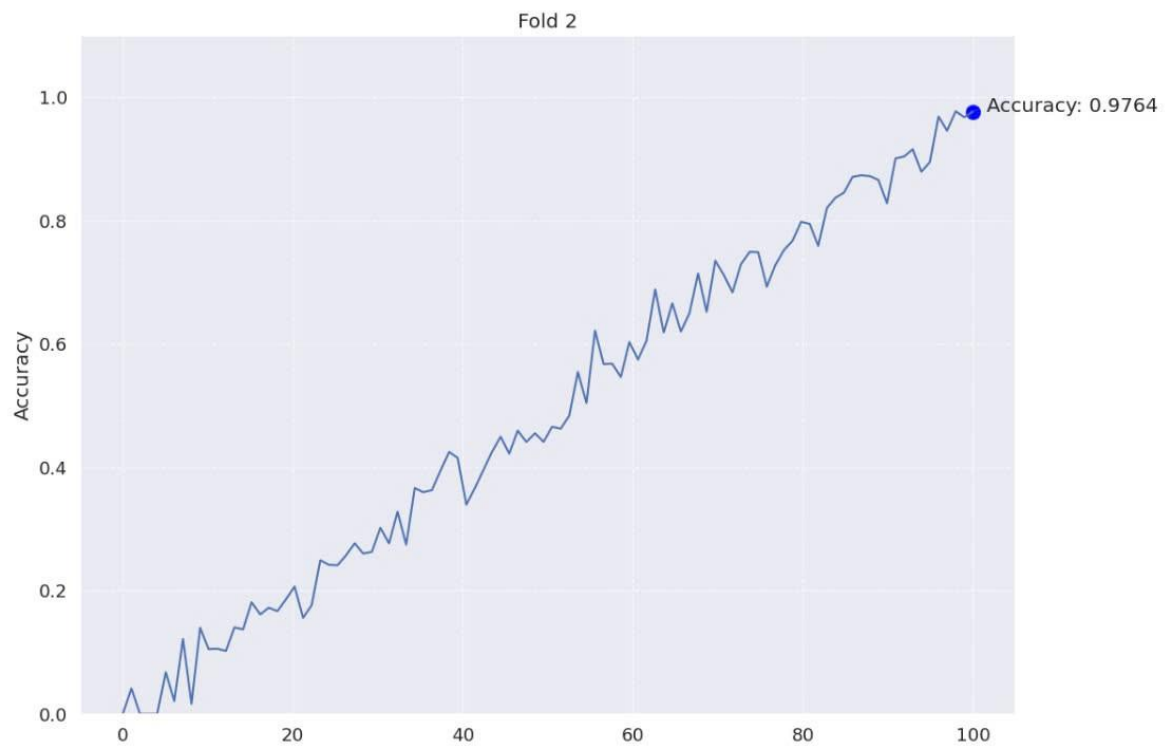


Hình 4. 3 $k = 3$ (PhoBERT)

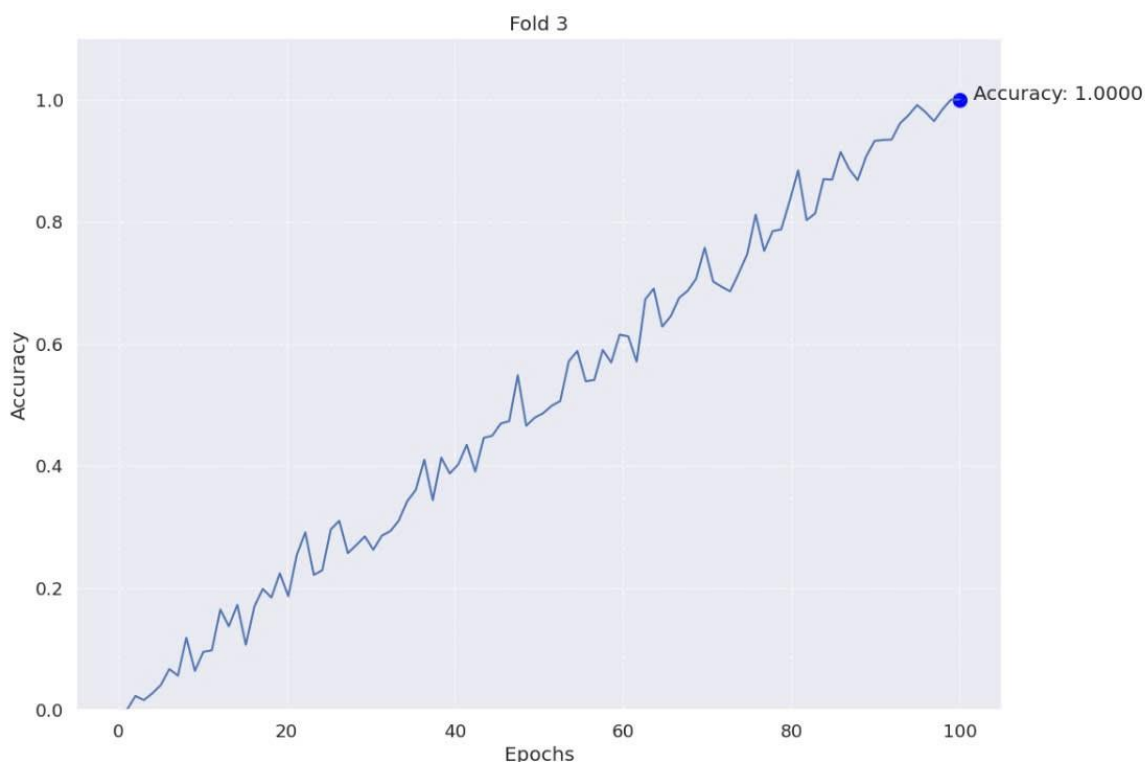
4.1.1.2. Mô hình ngôn ngữ BERT



Hình 4. 4 $k = 1$ (BERT)



Hình 4. 5 $k = 1$ (BERT)



Hình 4. 6 $k = 3$ (BERT)

4.1.2. Kết quả sau khi huấn luyện mô hình

Cả hai mô hình sẽ được huấn luyện thông qua phương pháp k-fold cross-validation với $k=3$. Trong mỗi lần lặp (fold), dữ liệu được chia thành ba phần: hai phần dùng để huấn luyện và phần còn lại để kiểm tra. Sau khi hoàn thành huấn luyện trên cả ba fold, kết quả của từng mô hình sẽ được tổng hợp lại bằng cách tính trung bình các chỉ số đánh giá. Cách tiếp cận này giúp chúng tôi đảm bảo rằng kết quả cuối cùng của mô hình là ổn định và đại diện nhất, đồng thời giảm thiểu tác động của việc phân chia ngẫu nhiên dữ liệu có thể gây ra thiên lệch. Việc lấy trung bình trên các fold không chỉ giúp đánh giá khách quan khả năng của từng mô hình mà còn hỗ trợ so sánh chính xác giữa các mô hình khác nhau để chọn ra mô hình tối ưu nhất.

4.1.2.1. Mô hình PhoBERT

Average Results Across All Folds								
	eval_loss	eval_accuracy	eval_f1	eval_precision	eval_recall	eval_runtime	eval_samples_per_second	eval_steps_per_second
0	0.4251	0.9213	0.9186	0.9332	0.9213	0.1602	802.1883	50.5317
epoch	58.1067							

Hình 4. 7 Kết quả mô hình PhoBERT

Độ chính xác (Accuracy): Độ chính xác trung bình(eval_accuracy) đạt 92.13%, cho thấy mô hình hoạt động rất tốt trong việc phân loại hoặc dự đoán đúng các câu trả lời cho bài toán chatbot tuyển sinh. Đây là một mức độ chính xác cao, phù hợp với bài toán yêu cầu tính chính xác cao.

Giá trị hàm mất mát (eval_loss) đạt 0.4251, cho thấy mô hình đã học tốt và có khả năng tổng quát hóa trên tập dữ liệu kiểm tra. Điều này cũng chứng minh rằng mô hình không gặp phải vấn đề overfitting nghiêm trọng.

Độ chính xác dự đoán (eval_precision) đạt 93.32%, cho thấy mô hình ít khi đưa ra các dự đoán sai. Đây là yếu tố quan trọng để chatbot đảm bảo chất lượng khi cung cấp câu trả lời.

Khả năng bao phủ (eval_recall) đạt 92.13%, chứng minh rằng mô hình nhận diện đúng hầu hết các trường hợp đúng, giúp tăng cường độ tin cậy của chatbot.

Giá trị trung bình của F1-score (eval_f1) đạt 91.86%, thể hiện sự cân bằng tốt giữa độ chính xác (Precision) và khả năng bao phủ (Recall) trong các dự đoán.

4.1.2.2. Mô hình BERT

Average Results Across All Folds									
	eval_loss	eval_Accuracy	eval_F1	eval_Precision	eval_Recall	eval_runtime	eval_samples_per_second	eval_steps_per_second	epoch
0	0.5383	0.892388	0.866764	0.874794	0.876955	0.2085	616.4717	38.8330	37.8167

Hình 4. 8 Kết quả mô hình BERT

Độ chính xác trung bình (eval_accuracy) đạt 89.24%, cho thấy mô hình hoạt động khá tốt trong việc phân loại hoặc dự đoán các câu trả lời cho bài toán chatbot tuyển sinh. Mức độ chính xác này tương đối cao nhưng có thể cần cải thiện thêm để đạt mức tốt hơn.

Giá trị hàm mất mát (eval_loss) đạt 0.5383, phản ánh rằng mô hình có khả năng học tốt, tuy nhiên vẫn còn dư địa để giảm thiểu lỗi để cải thiện độ chính xác và hiệu quả tổng thể.

Độ chính xác dự đoán (eval_precision) đạt 87.47%, phản ánh rằng mô hình có khả năng đưa ra các dự đoán chính xác trong hầu hết các trường hợp.

Khả năng bao phủ (eval_recall) đạt 87.70%, thể hiện rằng mô hình nhận diện đúng phần lớn các trường hợp cần dự đoán.

Giá trị trung bình của F1-score (eval_f1) đạt 86.68%, cho thấy sự cân bằng khá tốt giữa độ chính xác (Precision) và khả năng bao phủ (Recall). Tuy nhiên, điểm F1 chưa quá cao, điều này có thể liên quan đến tính chất của dữ liệu đầu vào hoặc tham số tối ưu hóa mô hình.

4.2. So sánh đánh giá và lựa chọn mô hình

Chỉ số	Hàm mất mát (Loss)	Độ chính xác (Accuracy)	Độ chính xác dự đoán (Precision)	Khả năng bao phủ (Recall)	F1-Score
PhoBERT	0.4251	92.13%	93.32%	92.13%	91.86%
BERT	0.5383	89.24%	87.47%	87.70%	86.68%

Bảng 4. 1 Bảng so sánh hai mô hình

- Hiệu suất cao hơn (PhoBERT): PhoBERT đạt được các chỉ số hiệu suất ấn tượng, với Accuracy lên tới 92.13%, F1-Score là 91.86%, và Precision đạt 93.32%. Các chỉ số này không chỉ cho thấy độ chính xác vượt trội trong việc xử lý các câu hỏi mà còn phản ánh khả năng của chatbot trong việc đưa ra câu trả lời chính xác và phù hợp với các câu hỏi phức tạp mà thí sinh hoặc phụ huynh có thể đưa ra. Đặc biệt, Recall cao (92.13%) chứng tỏ PhoBERT có khả năng nhận diện và xử lý một lượng lớn các trường hợp đa dạng, một yếu tố quan trọng khi chatbot cần cung cấp tư vấn cho nhiều đối tượng và câu hỏi khác nhau trong môi trường tuyển sinh.

- Với thời gian đánh giá chỉ 0.1602 giây và khả năng xử lý lên đến 802.19 mẫu/giây, PhoBERT không chỉ đảm bảo độ chính xác cao mà còn mang lại tốc độ phản hồi cực kỳ nhanh chóng. Điều này rất quan trọng trong việc cải thiện trải nghiệm người dùng, giúp thí sinh và phụ huynh nhận được câu trả lời tức thì, đặc biệt trong bối cảnh chatbot phải xử lý nhiều yêu cầu cùng lúc.

- Khả năng xử lý đồng thời (PhoBERT): Chatbot hỗ trợ tuyển sinh cần có khả năng phục vụ nhiều người dùng cùng lúc, nhất là trong các giai đoạn cao điểm khi lượng thí sinh truy cập vào hệ thống tăng cao. PhoBERT với tốc độ xử lý 50.53 bước/giây đảm bảo khả năng hoạt động ổn định và hiệu quả khi phải xử lý hàng nghìn yêu cầu đồng thời. Điều này giúp chatbot duy trì sự ổn định và liên tục cung cấp dịch vụ mà không bị gián đoạn, một yếu tố quan trọng đối với bất kỳ hệ thống nào yêu cầu sự sẵn sàng và đáp ứng nhanh.

- Tính linh hoạt và bao phủ: PhoBERT không chỉ nổi bật với F1-Score cao mà còn duy trì sự cân bằng tốt giữa việc đưa ra các dự đoán chính xác và khả năng nhận diện các câu hỏi khó. Điều này rất phù hợp với yêu cầu của chatbot tuyển sinh, khi hệ thống cần có khả năng cá nhân hóa các câu trả lời và cung cấp thông tin chi tiết về các ngành học, học phí, điều kiện tuyển sinh, v.v. Sự linh hoạt này giúp PhoBERT đáp ứng nhu cầu đa dạng và thay đổi của người dùng trong quá trình tư vấn.

- Hạn chế của BERT: Mặc dù BERT có khả năng hội tụ nhanh chóng hơn, chỉ cần 37.82 epoch để huấn luyện, nhưng nó phải đánh đổi hiệu suất với các chỉ số như độ chính xác, tốc độ và khả năng dự đoán thấp hơn. Cụ thể, BERT không thể duy trì mức độ chính xác và tốc độ phản hồi cao như PhoBERT, khiến nó không phù hợp với các ứng dụng chatbot yêu cầu chất lượng cao, tính ổn định và khả năng xử lý nhanh chóng. Việc sử dụng BERT có thể làm giảm hiệu suất tổng thể của hệ thống, đặc biệt là khi phải phục vụ nhiều người dùng và xử lý các câu hỏi phức tạp trong thời gian ngắn.

PhoBERT không chỉ vượt trội về mặt hiệu suất, tốc độ và khả năng xử lý đồng thời mà còn thể hiện sự linh hoạt trong việc cá nhân hóa các câu trả lời và bao quát được đa dạng các câu hỏi từ người dùng. Đặc biệt trong môi trường tuyển sinh, nơi yêu cầu chatbot phải đối mặt với lượng người dùng lớn, câu hỏi phức tạp và yêu cầu phản hồi nhanh chóng, PhoBERT là lựa chọn lý tưởng. So với BERT, PhoBERT đã được tối ưu hóa và có thể duy trì hiệu suất ổn định trong các tình huống thực tế, từ đó nâng cao chất lượng và sự hiệu quả của chatbot trong việc hỗ trợ tư vấn tuyển sinh.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đã đạt được

5.1.1. Kết quả

Qua quá trình nghiên cứu và thực nghiệm với hai mô hình ngôn ngữ tiên tiến là PhoBERT và BERT, chúng tôi đã đạt được những kết quả đáng khích lệ trong việc xây dựng một hệ thống chatbot tuyển sinh. Cả hai mô hình đều cho thấy khả năng vượt trội trong việc phân loại và dự đoán các câu trả lời chính xác cho các câu hỏi trong môi trường tuyển sinh.

Kết quả thực nghiệm cho thấy, PhoBERT là mô hình có hiệu suất vượt trội hơn so với BERT về tất cả các chỉ số đánh giá. Cụ thể, PhoBERT đạt được độ chính xác cao (92.13%), F1-score tốt (91.86%), và khả năng dự đoán chính xác (93.32%). Mô hình này cũng thể hiện khả năng xử lý đồng thời cao, với tốc độ xử lý nhanh chóng và khả năng phục vụ nhiều người dùng cùng lúc. Điều này chứng tỏ PhoBERT không chỉ đáp ứng được yêu cầu về chất lượng câu trả lời mà còn đảm bảo tốc độ phản hồi nhanh chóng, rất phù hợp cho các hệ thống chatbot yêu cầu tính sẵn sàng cao trong môi trường tuyển sinh.

Ngược lại, mô hình BERT, dù có kết quả khá tốt với độ chính xác 89.24% và F1-score 86.68%, vẫn có một số hạn chế về tốc độ và khả năng xử lý đồng thời so với PhoBERT. Đặc biệt, BERT không thể duy trì hiệu suất ổn định khi phải phục vụ nhiều người dùng đồng thời hoặc xử lý các câu hỏi phức tạp.

Thông qua các nghiên cứu trên, chúng tôi quyết sẽ định sẽ sử dụng và phát triển mô hình ngôn ngữ PhoBERT để *Xây dựng hệ thống tư vấn tuyển sinh trường Đại học Công Nghệ - Kỹ Thuật Cần Thơ*.

5.1.2. Hạn chế

Dữ liệu đầu vào: Mặc dù đã thực hiện tăng cường dữ liệu, dữ liệu đầu vào cho mô hình vẫn còn giới hạn và không phản ánh đầy đủ tất cả các tình huống có thể xảy ra trong môi trường tuyển sinh. Các câu hỏi phức tạp, các trường hợp ngoại lệ, hay các câu hỏi mang tính chất đặc thù từ người dùng đôi khi vẫn không được nhận diện chính xác, dẫn đến những phản hồi không đúng hoặc không đầy đủ.

Overfitting: Mặc dù sử dụng phương pháp K-fold cross-validation để giảm thiểu overfitting, trong một số trường hợp, mô hình PhoBERT vẫn có thể gặp phải vấn đề này khi huấn luyện trên các bộ dữ liệu nhỏ hoặc không đa dạng. Việc này có thể làm giảm khả năng tổng quát của mô hình khi áp dụng vào thực tế.

Khả năng mở rộng: Một số thử nghiệm cho thấy, mặc dù PhoBERT có thể xử lý nhiều yêu cầu cùng lúc, nhưng khi số lượng người dùng quá lớn hoặc khi có quá nhiều câu hỏi phức tạp, hệ thống có thể gặp khó khăn trong việc duy trì hiệu suất ổn định. Việc

tối ưu hóa khả năng xử lý đồng thời, đặc biệt trong môi trường có khối lượng dữ liệu lớn và phức tạp, vẫn là một thách thức.

Giới hạn trong việc hiểu ngữ cảnh sâu: Mặc dù mô hình PhoBERT đã có khả năng phân tích và trả lời các câu hỏi cơ bản và trung bình, nhưng đối với các câu hỏi yêu cầu hiểu ngữ cảnh sâu hoặc có tính tương tác cao, mô hình đôi khi vẫn gặp khó khăn. Điều này có thể dẫn đến việc chatbot không hiểu đúng ngữ nghĩa câu hỏi hoặc không thể cung cấp câu trả lời đầy đủ, đặc biệt đối với các câu hỏi có tính chất đa nghĩa hoặc phụ thuộc vào ngữ cảnh cụ thể.

5.2. Hướng phát triển

Tối ưu hóa tham số và mô hình: Mặc dù kết quả hiện tại của PhoBERT đã rất ấn tượng, vẫn có thể tiếp tục tối ưu hóa các tham số mô hình như learning rate, batch size, và số lượng epoch huấn luyện để cải thiện thêm độ chính xác và giảm thiểu sự overfitting. Việc nghiên cứu và áp dụng các kỹ thuật regularization hoặc fine-tuning cho các tác vụ cụ thể có thể mang lại những cải tiến đáng kể.

Mở rộng bộ dữ liệu huấn luyện: Một trong những yếu tố quan trọng quyết định chất lượng của mô hình là bộ dữ liệu huấn luyện. Việc tăng cường và mở rộng bộ dữ liệu với các câu hỏi và tình huống thực tế hơn sẽ giúp mô hình học được những phản hồi chính xác hơn và đa dạng hơn, từ đó cải thiện chất lượng chatbot.

Phân tích và cải thiện phương pháp tiền xử lý dữ liệu: Một trong những yếu tố quan trọng trong việc cải thiện độ chính xác của các mô hình ngôn ngữ là tiền xử lý dữ liệu. Cần nghiên cứu và phát triển các phương pháp tiền xử lý nâng cao, chẳng hạn như loại bỏ nhiễu từ các câu hỏi không liên quan, chuẩn hóa ngữ nghĩa, xử lý các từ ngữ đa nghĩa, hay các phương pháp trích xuất thông tin từ các câu hỏi dài. Việc cải thiện tiền xử lý sẽ giúp mô hình học được các đặc trưng chính xác hơn từ dữ liệu đầu vào.

Hỗ trợ đa ngôn ngữ: Với sự phát triển mạnh mẽ của các mô hình ngôn ngữ, việc mở rộng khả năng xử lý đa ngôn ngữ cho chatbot tuyển sinh có thể giúp nó phục vụ nhiều đối tượng người dùng hơn, bao gồm cả những người không nói tiếng Việt. PhoBERT có thể được mở rộng hoặc kết hợp với các mô hình ngôn ngữ khác để hỗ trợ nhiều ngôn ngữ hơn.

Tích hợp với các hệ thống thực tế: Để hệ thống chatbot thực sự hiệu quả trong môi trường tuyển sinh, cần phải tích hợp mô hình với các nền tảng tuyển sinh hiện có, như các hệ thống quản lý học sinh, website tuyển sinh, hoặc các hệ thống CRM (Customer Relationship Management). Việc tích hợp này không chỉ giúp chatbot nâng cao hiệu quả hỗ trợ mà còn giúp tăng tính tự động hóa trong quy trình tuyển sinh.

TÀI LIỆU THAM KHẢO

- [1] Y. W. Chandra and S. Suyanto Suyanto, "Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Model," *Procedia Computer Science*, pp. Volume 157, 2019, Pages 367-374, 12-13 09 2019.
- [2] A. Babu and S. B. Boddu, "BERT-Based Medical Chatbot: Enhancing Healthcare Communication through Natural Language Understanding," *Exploratory Research in Clinical and Social Pharmacy*, 15 02 2024.
- [3] O. Shaaban, "The Impact of Pre-trained Transformer-Based Language Model Use on Student Learning Outcomes in Higher Education - A Mixed-Methods Research Approach with a Case Study of IMC Fachhochschule Krems," 12 2023.
- [4] W. Wang, B. Bi, M. Yan, C. Wu, Z. Bao, J. Xia, L. Peng and L. Si, "STRUCTBERT: INCORPORATING LANGUAGE STRUCTURES," 13 08 2019.
- [5] J. L. Elman, "Finding structure in time," *ScienceDirect*, vol. 14, no. 2, pp. 179-211, 04-08 1990.
- [6] T. Zhang, K. Varsha, W. Felix, W. Q. Kilian and A. Yoav, "BERTSCORE: EVALUATING TEXT GENERATION WITH," 24 02 2020.
- [7] K. Clark, U. Khandelwal, O. Levy and C. D. Manning, "What does BERT look at? An Analysis of BERT's Attention," *Proceedings of the Second BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, p. 276–286, 01 08 2019.
- [8] P. T. T. Nga, N. T. Luong, T. H. Thang and T. D. Quy, "Một cách tiếp cận xây dựng ứng dụng Chatbot tư vấn tuyển sinh trường Đại học Đà Lạt," *Tạp Chí Khoa Học và Công Nghệ Đại học Thái Nguyên*, pp. T. 227, S. 15, 23 08 2022.
- [9] P. D. Anh and L. T. Huong, "A Question-Answering System for Vietnamese Public Administrative Services," *SOICT '23: Proceedings of the 12th International Symposium on Information and Communication Technology*, pp. Pages 85 - 92, 12 2023.
- [10] B. Đ. Thọ, Đ. T. T. Trang and N. T. Huyền, "SỬ DỤNG BERT CHO TÓM TẮT VĂN BẢN TIẾNG VIỆT," ISSN 2354-0575, 02 12 2021.
- [11] N. T. T. Thủy and N. N. Điệp, "SỬ DỤNG BERT VÀ CÂU PHỤ TRỢ CHO TRÍCH XUẤT KHÍA CẠNH TRONG VĂN BẢN TIẾNG VIỆT," *Journal of Science*

and Technology on Information and Communications (PTIT), p. Vol. 1 No. 4, 30 12 2022.

- [12] N. V. Sơn, N. T. M. Nghĩa, H. T. Huế, N. H. Liêm and V. V. M. Nhật, "MỘT CÁI TIẾN CỦA PHOBERT NHẪM TĂNG KHẢ NĂNG HIỂU TIẾNG VIỆT CỦA CHATBOT THÔNG TIN KHÁCH SẠN," *Tạp chí Khoa học Đại học Huế: Kỹ thuật và Công nghệ*; pISSN 2588-1175/ eISSN 2615-9732, pp. Tập 131, Số 2A, 2022, Tr. 5–20, 08 11 2022.
- [13] E. Alpaydın, *Introduction to Machine Learning*(4th ed).
- [14] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MA: MIT Press, 1990.
- [15] S. Hussain, O. A. Sianaki and N. Ababneh , *A Survey on Conversational Agents/Chatbots Classification and Design Techniques*, 2019, p. pp 946–956.
- [16] O. Vinyals and L. Quoc, "A Neural Conversational Model," *ICML Deep Learning Workshop 2015*, 19 06 2015.
- [17] D. Jurafsky and J. H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition," 2008.
- [18] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, 1989.
- [19] F. Jelinek, "Statistical Methods for Speech Recognition," MIT Press, 1997.
- [20] D. Jurafsky and J. H. Martin, "Speech and Language Processing (3rd ed.)," Pearson, 20 08 2024.
- [21] C. M. Bishop, "Pattern Recognition and Machine Learning," Springer, 2006.
- [22] C. CORTES and V. VAPNIK, "Support-vector networks," vol. 20, p. 273–297, 1995.
- [23] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.