

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**HUỲNH HỒNG SƠN – 52000795
TRƯỜNG THÀNH LỢI – 52000776**

**SCHOOLASSISTANT
HỆ THỐNG TƯ VẤN TỰ ĐỘNG
TRONG TRƯỜNG HỌC**

DỰ ÁN CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH KHOA HỌC MÁY TÍNH

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**HUỲNH HỒNG SƠN – 52000795
TRƯỜNG THÀNH LỢI – 52000776**

**SCHOOLASSISTANT
HỆ THỐNG TƯ VẤN TỰ ĐỘNG
TRONG TRƯỜNG HỌC**

DỰ ÁN CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH KHOA HỌC MÁY TÍNH

Người hướng dẫn
TS. Trần Lương Quốc Đại

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

LỜI CẢM ƠN

Chúng em xin bày tỏ lòng cảm ơn sâu sắc tới TS. Trần Lương Quốc Đại về sự hướng dẫn và sự hỗ trợ mà thầy đã dành cho chúng em trong quá trình hoàn thành dự án công nghệ thông tin này.

Thầy đã không ngừng chia sẻ kiến thức, kinh nghiệm và sự động viên tích cực giúp cho chúng em hiểu hơn về đề tài “SchoolAssistant hệ thống tự động trong trường học” cũng như phát triển kỹ năng làm việc nhóm và nâng cao khả năng nghiên cứu. Một lần nữa, chúng em xin chân thành cảm ơn đến TS. Trần Lương Quốc Đại.

TP. Hồ Chí Minh, ngày ... tháng ... năm 2024

Tác giả

(Ký tên và ghi rõ họ tên)

Huỳnh Hồng Sơn

Trương Thành Lợi

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là công trình nghiên cứu của riêng chúng tôi và được sự hướng dẫn khoa học của TS. Trần Lương Quốc Đại. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Dự án của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày ... tháng ... năm 2024

Tác giả

(Ký tên và ghi rõ họ tên)

Huỳnh Hồng Sơn

Trương Thành Lợi

SCHOOLASSISTANT

HỆ THỐNG TƯ VẤN TỰ ĐỘNG TRONG TRƯỜNG HỌC

TÓM TẮT

Mục tiêu: Thông qua các bài nghiên cứu trên các nền tảng và yêu cầu của đề tài cùng với các kiến thức nền tảng về mô hình học máy, học sâu kèm 1 phần xử lý ngôn ngữ tự nhiên thì hệ thống tư vấn trả lời câu hỏi tự động trong trường học được thiết kế để hỗ trợ học sinh và sinh viên trong việc học tập bằng cách tự động trả lời các câu hỏi của họ.

Thành phần: Hệ thống bao gồm các thành phần chính sau:

- +Gói dữ liệu: Gồm các câu hỏi và câu trả lời được thu thập từ người dùng cùng với xử lý tiền dữ liệu để ra được đầu vào cho mô hình.

- +Mô hình xử lý ngôn ngữ tự nhiên: Mô hình LSTM được sử dụng để hiểu ý nghĩa của câu hỏi và tìm kiếm câu trả lời phù hợp trong gói dữ liệu bằng cách phân loại nhãn.

- +Giao diện người dùng: Áp dụng mô hình sau khi đào tạo vào giao diện cho phép người dùng nhập câu hỏi và nhận câu trả lời từ hệ thống thông qua thư viện Gradio.

Dựa trên cách đào tạo mô hình và thông số từ quá trình huấn luyện đưa ra các kết luận trực quan cùng với sự hứa hẹn phát triển về sau.

SCHOOLASSISTANT

HỆ THỐNG TƯ VẤN TỰ ĐỘNG TRONG TRƯỜNG HỌC

ABTRACT

MỤC LỤC

DANH MỤC HÌNH VẼ	viii
DANH MỤC BẢNG BIỂU	ix
DANH MỤC CÁC CHỮ VIẾT TẮT.....	x
CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI.....	1
1.1 Lý do chọn đề tài.....	1
1.2 Tổng quan.....	1
1.2.1 Các công trình nghiên cứu trong lĩnh vực QAS ở Việt Nam	1
1.2.2 Các công trình nghiên cứu trong lĩnh vực QAS ở quốc tế.....	6
1.3 Mục tiêu của nghiên cứu	11
1.4 Cách thức tiếp cận	12
1.5 Đối tượng và phạm vi tiếp cận bài toán	12
1.6 Tóm tắt nội dung sẽ trình bày trong báo cáo.....	12
CHƯƠNG 2. KIẾN THỨC NỀN TẢNG	14
2.1 Hệ thống trả lời câu hỏi.....	14
2.2 Mô hình đặt câu hỏi và trả lời tự động.....	15
2.3 Phân loại các mô hình đặt và trả lời câu hỏi	16
2.3.1 Phân loại hệ thống QA theo miền (Mervin, 2013).....	16
2.3.2 Phân loại mô hình với nhiều hướng.....	16
2.4 Tổng quan về Neural Network.....	17
2.5 Recurrent Neural Network (RNN).....	19
2.6 Long Short-Term Memory (LSTM) và Gated Recurrent Unit (GRU)	20
2.6.1 Long Short-Term Memory (Dancker, 2022; Fouzan, 2023).....	20

2.6.2 Gated Recurrent Unit (Dancker, 2022; Fouzan, 2023).....	22
2.6.3 So sánh RNN, GRU và LSTM	23
2.7 Embedding và Keras Embedding Layer (<i>Embedding — Machine Learning Cho Dữ Liệu Dạng Bảng</i> , n.d.; Team, n.d.).....	24
CHƯƠNG 3. XÂY DỰNG BỘ DỮ LIỆU VÀ MÔ HÌNH CHO BÀI TOÁN QAS	26
3.1 Đề xuất phương pháp cho bài toán.....	26
3.2 Input: Dữ liệu tư vấn về trường học.....	26
3.3 Processing	26
CHƯƠNG 4. THỰC NGHIỆM	28
4.1 Dataset.....	28
4.1.1 Cơ sở của bộ dữ liệu	28
4.1.2 Cấu trúc bộ dữ liệu	28
4.1.3 Tiền xử lý dữ liệu	29
4.2 Khởi tạo mô hình.....	31
4.3 Đánh giá	34
4.4 Nhận xét thực nghiệm	37
4.5 Xây dựng API.....	38
4.6 Xây dựng giao diện web chatbot.....	39
CHƯƠNG 5. KẾT LUẬN.....	41
5.1 Kết luận	41
5.2 Hạn chế và hướng phát triển	41
5.2.1 Hạn chế	41
5.2.2 Hướng phát triển.....	41

TÀI LIỆU THAM KHẢO	42
---------------------------------	-----------

DANH MỤC HÌNH VẼ

Hình 2. 1: Kiến trúc của mô hình seq2seq	15
Hình 2. 2: Cách thức hoạt động của mô hình seq2seq	16
Hình 2. 3: Kiến trúc Neural Network.....	18
Hình 2. 4: Các thành phần chính của LSTM.....	20
Hình 2. 5: Các mô-đun lặp lại trong mạng RNN chỉ chứa một lớp	21
Hình 2. 6: Các mô-đun lặp lại trong một LSTM	22
Hình 2. 7: Kiến trúc của GRU.....	22
 Hình 3. 1: Thuật toán xử lý cho phương pháp đã đề xuất.....	26
 Hình 4. 1: Bộ dữ liệu.....	29
Hình 4. 2: Xử lý dữ liệu	31
Hình 4. 3: Chạy mô hình.....	33
Hình 4. 4: Biểu đồ thể hiện kết quả thực nghiệm mô hình	34
Hình 4. 5: Chatbot trả lời về thông tin ngành khoa học máy tính.....	35
Hình 4. 6: Chatbot trả lời về các thông tin của các chương trình đào tạo.....	36
Hình 4. 7: Chatbot trả lời về các thông tin của phòng ban.....	37

DANH MỤC BẢNG BIỂU

Bảng 2. 1: So sánh RNN, GRU và LSTM	23
---	----

DANH MỤC CÁC CHỮ VIẾT TẮT

BERT	Bidirectional Encoder Representations from Transformers
QA	Question Answering
NLP	Natural Language Processing
NL	Natural Language
AI	Artificial Intelligence
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
ML	Machine Learning
GRU	Gated Recurrent Unit

CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

1.1 Lý do chọn đề tài

Đề tài này là một cơ hội để nhóm có thể tìm hiểu các nội dung hoặc chủ đề liên quan với mục tiêu giải quyết được bài toán và học thêm những kiến thức mới liên quan đến học máy, học sâu và xử lý ngôn ngữ tự nhiên. Cùng với việc xây dựng hệ thống hỏi đáp này sẽ giúp ích trong trường hợp khi các phụ huynh, học sinh, sinh viên muốn tìm hiểu về thông tin tuyển sinh của trường hoặc học phí, học bổng, liên hệ, ... nhưng không có thời gian để tìm kiếm các thông tin, thì có thể sử dụng chatbot trả lời tự động này thay thế. Đồng thời, hỗ trợ trường học trong việc giúp người dùng có thể tự mình tìm hiểu thông tin liên quan thông qua hệ thống.

1.2 Tổng quan

1.2.1 Các công trình nghiên cứu trong lĩnh vực QAS ở Việt Nam

“A Vietnamese Question Answering System” (Hệ thống trả lời câu hỏi)
(D. Nguyen et al., 2009)

Hệ thống trả lời câu hỏi (QA) truyền thống thường chỉ cung cấp danh sách tài liệu liên quan cho người dùng, khiến họ mất thời gian để tìm kiếm thông tin cần thiết. Để giải quyết vấn đề này, chúng tôi đề xuất hệ thống QA tiếng Việt dựa trên ontology - một cấu trúc dữ liệu mô tả các khái niệm và mối quan hệ trong một lĩnh vực cụ thể.

Hệ thống QA tiếng Việt dựa trên ontology có những ưu điểm sau:

- Cung cấp câu trả lời chính xác: Hệ thống hiểu ý nghĩa của câu hỏi và truy vấn cơ sở tri thức ontology để tìm kiếm thông tin chính xác nhất.
- Hiểu ngôn ngữ tự nhiên: Người dùng có thể diễn đạt câu hỏi bằng tiếng Việt một cách tự nhiên, không cần tuân theo cấu trúc cụ thể nào.
- Dễ sử dụng: Giao diện đơn giản, dễ thao tác, phù hợp với mọi đối tượng người dùng.

Đây là hệ thống QA tiếng Việt đầu tiên sử dụng ontology, cho phép người dùng truy vấn thông tin một cách hiệu quả và trực quan hơn. Hệ thống đã được thử

nghiệm và cho thấy kết quả đầy hứa hẹn, mở ra tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực như giáo dục, y tế, kinh doanh, v.v. So sánh với hệ thống QA truyền thống thì QAS dựa trên ontology cung cấp câu trả lời chính xác hơn, hiểu được các câu hỏi phức tạp hơn và cho phép người dùng truy cập thông tin dễ dàng hơn. Còn QAS truyền thống chỉ cung cấp danh sách tài liệu liên quan, khiến người dùng mất thời gian để tìm kiếm thông tin cần thiết. Qua đó, ta có thể thấy hệ thống QA tiếng Việt dựa trên ontology là một bước tiến mới trong lĩnh vực xử lý ngôn ngữ tự nhiên tiếng Việt, giúp người dùng truy cập thông tin dễ dàng và nhanh chóng hơn.

“BERT+vnKG: Using Deep Learning and Knowledge Graph to Improve Vietnamese Question Answering System” (Hệ thống Trả lời câu hỏi tiếng Việt tiên tiến sử dụng BERT và Đồ thị tri thức) (Do & Phan, 2021)

Bài báo "BERT+vnKG: Sử dụng Học sâu và Đồ thị tri thức để Cải thiện Hệ thống Trả lời câu hỏi tiếng Việt" (Do & Phan, 2021) đề xuất một hệ thống QA tiếng Việt tiên tiến sử dụng mô hình BERT và Đồ thị tri thức (KG). Hệ thống này có những ưu điểm vượt trội bao gồm:

- Độ chính xác cao hơn: BERT giúp hệ thống hiểu rõ hơn ý nghĩa của câu hỏi và ngữ cảnh, từ đó đưa ra câu trả lời chính xác hơn.
- Khả năng xử lý ngôn ngữ tự nhiên mạnh mẽ: Hệ thống có khả năng xử lý ngôn ngữ tự nhiên mạnh mẽ, cho phép nó hiểu các câu hỏi phức tạp và diễn đạt bằng tiếng Việt tự nhiên.
- Hiệu quả cao: Việc sử dụng KG giúp hệ thống truy xuất thông tin nhanh chóng và chính xác hơn.

So sánh với hệ thống QA truyền thống: Hệ thống QA truyền thống thường sử dụng mô hình LSTM, vốn có những hạn chế về khả năng. Hệ thống QA dựa trên BERT+vnKG có độ chính xác cao hơn, khả năng xử lý ngôn ngữ tự nhiên mạnh mẽ hơn và hiệu quả cao hơn. Qua đó có thể thấy: Hệ thống QA tiếng Việt dựa trên BERT+vnKG là một bước tiến mới trong lĩnh vực xử lý ngôn ngữ tự nhiên tiếng Việt, giúp người dùng truy cập thông tin dễ dàng và nhanh chóng hơn.

“Building Filters for Vietnamese Chatbot Responses” (Chatbot tiếng Việt hiệu quả với bộ lọc phản hồi thông minh) (Le et al., 2020)

Bài báo "Tạo bộ lọc cho phản hồi của Chatbot tiếng Việt" (Le et al., 2020) đề xuất phương pháp cải thiện hiệu quả chatbot tiếng Việt bằng cách sử dụng bộ lọc phản hồi thông minh. Phương pháp này giúp giảm thiểu các phản hồi sai, không chính xác hoặc gây nhầm lẫn cho người dùng. Lợi ích:

- Cải thiện độ chính xác: Chatbot chỉ đưa ra những phản hồi chính xác và phù hợp với ngữ cảnh.
- Tăng sự hài lòng của người dùng: Tránh gây khó chịu hoặc nhầm lẫn cho người dùng do phản hồi sai.
- Mở rộng ứng dụng: Chatbot có thể được sử dụng trong nhiều lĩnh vực hơn, bao gồm cả các lĩnh vực đòi hỏi tính chính xác cao như pháp lý và tài chính.

Điểm khác biệt so với chatbot truyền thống: Chatbot truyền thống thường sử dụng mô hình tiếng Anh, dẫn đến kết quả đầu ra không chính xác khi sử dụng tiếng Việt. Chatbot sử dụng bộ lọc phản hồi thông minh có thể hiểu được đặc điểm của ngôn ngữ tiếng Việt và đưa ra những phản hồi phù hợp.

1. Thương mại điện tử: Chatbot hỗ trợ khách hàng tìm kiếm sản phẩm phù hợp, giải đáp thắc mắc và hướng dẫn đặt hàng. Tự động hóa quy trình thanh toán, tạo hóa đơn và theo dõi đơn hàng. Gửi lời nhắc nhở về sản phẩm đang được xem, sản phẩm sắp hết hàng hoặc khuyến mãi hấp dẫn.
2. Chăm sóc khách hàng: Trả lời tự động các câu hỏi thường gặp (FAQ) 24/7, giải quyết các vấn đề đơn giản. Thu thập phản hồi của khách hàng về sản phẩm, dịch vụ. Cá nhân hóa trải nghiệm khách hàng bằng cách ghi nhớ sở thích và nhu cầu của họ.
3. Giáo dục: Cung cấp thông tin về các khóa học, chương trình đào tạo, lịch học và quy trình đăng ký. Hỗ trợ sinh viên giải đáp thắc mắc về học tập,

tài liệu và bài tập. Cá nhân hóa lộ trình học tập cho từng học sinh dựa trên năng lực và sở thích.

Qua đó, phương pháp sử dụng bộ lọc phản hồi thông minh là một giải pháp hiệu quả để cải thiện chatbot tiếng Việt. Phương pháp này giúp chatbot đưa ra những phản hồi chính xác, phù hợp và mang lại trải nghiệm tốt hơn cho người dùng.

“Understanding what the users say in chatbots: A case study for the Vietnamese language” (Hiểu người dùng nói gì trong Chatbot: Nghiên cứu trường hợp cho ngôn ngữ tiếng Việt) (Tran & Luong, 2020)

Bài viết trình bày một nghiên cứu về việc hiểu nội dung người dùng truyền đạt trong hệ thống chatbot. Nghiên cứu tập trung vào việc xác định ý định của người dùng và nhận biết ngữ cảnh tương ứng trong câu nói ngụ ý mà họ nhập vào. Điều này cho phép chatbot hiểu rõ hơn những gì người dùng đang truyền đạt và có khả năng thực hiện hành động trong một phạm vi rộng hơn. Phương pháp được đề xuất trong nghiên cứu này bao gồm hai nhiệm vụ chính: phân loại và ghi nhãn chuỗi hai lớp. Khung mô hình này sử dụng mạng neural mạnh để tự động học các đặc trưng hữu ích ở cả mức ký tự và từ. Áp dụng khung mô hình này, nghiên cứu xây dựng một chatbot cho lĩnh vực thương mại điện tử ở Việt Nam, nhằm cung cấp khả năng giao tiếp tốt hơn cho các thương hiệu bán lẻ với khách hàng của họ.

Thử nghiệm trên bốn bộ dữ liệu mới cho thấy mạng neural mạnh mẽ đạt hiệu suất vượt trội so với các phương pháp học máy truyền thống.

Đối với nhiệm vụ xác định ý định: Hệ thống đạt được độ đo F tốt nhất là 82,32%, thể hiện khả năng xác định chính xác ý định của người dùng trong văn bản.

Đối với nhiệm vụ nhận biết ngữ cảnh: Phương pháp đề xuất đạt được các độ đo F trong khoảng từ 78% đến 91%, tùy thuộc vào loại ngữ cảnh cụ thể. Điều này cho thấy hiệu quả cao trong việc nhận diện ngữ cảnh của văn bản, bao gồm cả các ngữ cảnh phức tạp và chuyên biệt.

Kết quả thử nghiệm khẳng định tiềm năng của mạng neural mạnh mẽ trong việc xử lý ngôn ngữ tự nhiên. Hệ thống có thể được ứng dụng hiệu quả trong nhiều lĩnh vực khác nhau, như chatbot, dịch thuật tự động, phân loại văn bản, v.v.

“A Neural Network based Vietnamese Chatbot” (Chatbot tiếng Việt dựa trên mạng nơ-ron) (T. Nguyen & Shcherbakov, 2018)

Hiện nay, chatbot là một chủ đề được quan tâm rất nhiều, và việc sử dụng các mô hình chatbot đã mang lại nhiều thành công. Bài viết này hướng đến mục tiêu phát triển một chatbot tiếng Việt sử dụng mô hình seq2seq kết hợp với cơ chế chú thích phù hợp. Chúng tôi đã xây dựng và thử nghiệm mô hình này trên PyTorch sử dụng GPU. Mô hình được huấn luyện từ đầu đến cuối mà không cần can thiệp thủ công. Mô hình chatbot được xây dựng dựa trên tập dữ liệu nhỏ có khả năng tạo ra câu trả lời cho người dùng. Tuy nhiên, để tạo ra một cuộc trò chuyện có ý nghĩa và thu hút, cần nâng cao chất lượng câu trả lời. Chúng tôi sử dụng một biến thể của mô hình Dịch máy thần kinh của PyTorch được phát triển bởi Sean Robertson vào năm 2017.

Mô hình này sử dụng kiến trúc RNN kết hợp với cơ chế chú ý để tạo ra chatbot hiệu quả.

Cấu trúc: Mạng Seq2seq: Gồm hai mạng nơ-ron tuần hoàn (RNN) - bộ mã hóa và bộ giải mã.

Bộ mã hóa: Biểu diễn đầu vào (câu hỏi hoặc tuyên bố) thành một vector duy nhất và một trạng thái ẩn.

Bộ giải mã: Sử dụng vector và trạng thái ẩn từ bộ mã hóa để tạo ra câu trả lời.

Cơ chế chú ý: Cho phép bộ giải mã tập trung vào các phần quan trọng của đầu vào, giúp tạo ra câu trả lời chính xác và phù hợp hơn.

Ưu điểm: Hiểu rõ ngữ cảnh và ý định của người dùng. Tạo ra các câu trả lời tự nhiên và logic. Tăng hiệu quả và khả năng tương tác của chatbot.

Lợi ích: Nâng cao trải nghiệm cho người dùng. Mở rộng ứng dụng chatbot trong thực tế.

**”NEU-chatbot: Chatbot for admission of National Economics University”
(Chatbot thông minh hỗ trợ tuyển sinh tại Đại học Kinh tế Quốc dân) (T. T. Nguyen et al., 2021)**

Trong bài báo này đã nhấn mạnh sự phổ biến của hệ thống chatbot thông minh trong nhiều lĩnh vực ứng dụng, đặc biệt là trong giáo dục. Nhu cầu sử dụng các dịch vụ tư vấn trực tuyến như chatbot đã tăng cao. Tuy nhiên, việc liên lạc giữa sinh viên tiềm năng và trường đại học thường được thực hiện thông qua các phương pháp thủ công, gây mất thời gian và làm khó khăn cho người quản lý tuyển sinh. Trong bài viết này, tác giả giới thiệu một chatbot dựa trên trí tuệ nhân tạo (AI) cho phép học sinh nhận thông tin cập nhật hàng ngày về chương trình giảng dạy, tuyển sinh, học phí, điểm IELTS writing task II và nhiều thông tin khác. Chatbot được phát triển bằng cách tích hợp các mô hình học sâu vào khung Rasa. Tác giả cũng đề xuất một quy trình hợp lý cho việc xử lý dữ liệu tiếng Việt nhằm đạt được độ chính xác tối ưu và tránh tình trạng quá khớp mô hình. Mô hình chatbot có khả năng phát hiện hơn 50 loại câu hỏi dựa trên thông tin đầu vào của người dùng, và đạt độ chính xác 97,1% trên bộ kiểm tra. Chatbot đã được triển khai trên trang tuyển sinh chính thức của Đại học Kinh tế Quốc dân trên nền tảng Facebook, mạng xã hội phổ biến nhất tại Việt Nam. Nghiên cứu này cung cấp hướng dẫn chi tiết về quá trình xây dựng một chatbot AI từ đầu và các kỹ thuật được sử dụng, có thể áp dụng cho bất kỳ ngôn ngữ nào trên toàn thế giới.

1.2.2 Các công trình nghiên cứu trong lĩnh vực QAS ở quốc tế

***“Knowledge Base Question Answering Based on Deep Learning Models”
(Hệ thống trả lời câu hỏi dựa trên kiến thức sử dụng mô hình học sâu) (Xie et al., 2016)***

Trong nghiên cứu "Knowledge Base Question Answering Based on Deep Learning Models" (Xie et al., 2016), tác giả chủ yếu tập trung vào việc trả lời câu hỏi dựa trên tri thức (KBQA). KBQA nhằm mục đích kết hợp câu hỏi với ý nghĩa có cấu trúc trong cơ sở tri thức. Trong bài báo này, tác giả đề xuất một phương pháp hai giai đoạn.

Hệ thống này sử dụng mô hình tiên tiến để trích xuất các bộ ba kiến thức liên quan đến câu hỏi từ cơ sở tri thức. Cách thức hoạt động:

1. Trích xuất thực thể chủ đề: Sử dụng mô hình TEEM để tự động trích xuất các thực thể chủ đề trong câu hỏi mà không cần dựa vào các đặc trưng thủ công hoặc công cụ ngôn ngữ.

2. Tìm kiếm bộ ba kiến thức: Dựa vào các thực thể chủ đề đã trích xuất, hệ thống sẽ tìm kiếm các bộ ba kiến thức có liên quan từ cơ sở tri thức.

3. Phù hợp câu hỏi và vị từ: Áp dụng mô hình ngữ nghĩa có cấu trúc sâu dựa trên mạng thần kinh tích chập và bộ nhớ ngắn hạn dài hai chiều để so khớp câu hỏi và vị từ trong bộ ba kiến thức ứng viên.

4. Cải thiện tập dữ liệu: Sử dụng phương pháp lặp để truy xuất bộ ba kiến thức từ cơ sở tri thức, giúp tạo ra tập dữ liệu huấn luyện tốt hơn.

Kết quả: Hệ thống đạt được chỉ số F1 trung bình là 79,57% trên tập dữ liệu thử nghiệm, cho thấy hiệu quả cao trong việc trích xuất bộ ba kiến thức liên quan đến câu hỏi.

Lợi ích: Nâng cao khả năng hiểu và đáp ứng nhu cầu của người dùng. Mở rộng ứng dụng hệ thống trong các lĩnh vực như chatbot, dịch thuật tự động, v.v.

Điểm nổi bật: Sử dụng mô hình TEEM tiên tiến để trích xuất thực thể chủ đề tự động. Áp dụng mô hình ngữ nghĩa có cấu trúc sâu để phù hợp câu hỏi và vị từ. Sử dụng phương pháp lặp để cải thiện tập dữ liệu huấn luyện. Với những ưu điểm và hiệu quả cao, hệ thống trích xuất bộ ba kiến thức liên quan đến câu hỏi hứa hẹn sẽ mang đến nhiều ứng dụng thực tế hữu ích.

“A building regulation question answering system: A deep learning methodology” (Hệ thống trả lời câu hỏi về quy định xây dựng: Phương pháp học sâu) (Zhong et al., 2020)

Nghiên cứu này, tác giả tập trung vào việc phát triển một hệ thống trả lời câu hỏi về quy định xây dựng. Quy định đóng vai trò thiết yếu trong việc đảm bảo chất lượng công trình, bảo vệ môi trường và thúc đẩy sự phát triển bền vững. Các kỹ sư và chuyên gia cần truy cập thông tin về quy định để đảm bảo rằng công trình tuân thủ

các tiêu chuẩn. Tuy nhiên, việc truy xuất thông tin cho các nhu cầu cụ thể trong lĩnh vực này thường gặp khó khăn và tốn thời gian. Nghiên cứu này giới thiệu một hệ thống tiên tiến nhằm cải thiện hiệu quả và khả năng truy xuất thông tin cho các truy vấn liên quan đến quy định xây dựng. Hệ thống này kết hợp các kỹ thuật tiên tiến như:

Truy xuất thông tin: Tìm kiếm thông tin liên quan từ bộ sưu tập các quy định xây dựng.

Xử lý ngôn ngữ tự nhiên (NLP): Hiểu rõ ngữ nghĩa và ý định của người dùng trong các câu hỏi.

Học sâu: Tự động học hỏi và cải thiện khả năng truy xuất thông tin theo thời gian.

Với những lợi ích như cung cấp câu trả lời chính xác và nhanh chóng, dễ sử dụng và khả năng học hỏi và thích ứng theo thời gian. Hệ thống này là một bước tiến quan trọng trong việc cải thiện hiệu quả và khả năng truy xuất thông tin quy định xây dựng. Hệ thống có thể được áp dụng cho các lĩnh vực pháp lý khác nhau, giúp nâng cao hiệu quả hoạt động và cung cấp dịch vụ tốt hơn cho người dân.

“A Technical Question Answering System with Transfer Learning” (Hệ thống trả lời câu hỏi kỹ thuật sử dụng học chuyển giao) (Yu et al., 2020)

Trong nghiên cứu này, tác giả tập trung vào việc phát triển một hệ thống trả lời câu hỏi kỹ thuật dựa trên việc chuyển giao học. Trong thời gian gần đây, nhu cầu về các trang web cung cấp các câu trả lời kỹ thuật từ cộng đồng đã tăng đáng kể. Tuy nhiên, chuyên gia thường mất nhiều thời gian để đưa ra phản hồi hữu ích và kịp thời trên các diễn đàn đó. Nhu cầu về các trang web cung cấp câu trả lời kỹ thuật từ cộng đồng đang tăng cao, nhưng chuyên gia thường thiếu thời gian để phản hồi kịp thời. Hệ thống TransTQA được phát triển để giải quyết vấn đề này bằng cách tự động truy xuất các câu trả lời thích hợp dựa trên các câu hỏi tương tự đã được trả lời trong quá khứ. TransTQA sử dụng mạng ALBERT của Siamese để đưa ra phản hồi nhanh chóng và chính xác, đồng thời áp dụng chiến lược học chuyển giao sâu để hỗ trợ nhiều lĩnh vực kỹ thuật. Nhờ vậy, TransTQA có thể cung cấp giải pháp hiệu quả cho

việc truy cập thông tin kỹ thuật một cách nhanh chóng và tiện lợi. Trong tổng quan, nghiên cứu này đóng góp bằng việc phát triển một hệ thống trả lời câu hỏi kỹ thuật sử dụng transfer learning, giúp cung cấp phản hồi nhanh chóng và chính xác. Hệ thống TransTQA có khả năng hỗ trợ cho nhiều lĩnh vực kỹ thuật nhờ áp dụng chiến lược học chuyển giao sâu.

“Deep Learning of Grammatically-Interpretable Representations Through Question-Answering” (Học biểu diễn ngữ pháp có thể diễn giải được bằng mạng nơ-ron sâu) (Palangi et al., 2017)

Trong nghiên cứu mang tựa đề "Deep Learning of Grammatically-Interpretable Representations Through Question-Answering" của Palangi và đồng nghiệp (2017), chúng tôi giới thiệu một kiến trúc mạng nơ-ron sâu mà chúng tôi đã điều chỉnh để giải quyết nhiệm vụ trả lời câu hỏi từ văn bản. Kiến trúc mô hình này cho phép giải thích các biểu diễn bên trong bằng các khái niệm ngôn ngữ cơ bản. Khả năng giải thích này chỉ làm giảm độ chính xác một vài phần trăm so với mô hình gốc BiDAF. Biểu diễn bên trong là Tensor Product Repre, trong đó mô hình chọn biểu tượng cho mỗi từ đầu vào và vai trò để định vị và kết nối các từ. Việc lựa chọn này được thực hiện bằng cơ chế chú ý nhẹ nhàng. Giải thích tổng thể được xây dựng dựa trên giải thích của các biểu tượng và vai trò. Kết quả cho thấy các biểu tượng có thể được hiểu là nghĩa của từ theo từ điển ngôn ngữ, trong khi vai trò có thể được hiểu là các vai trò ngữ pháp gần đúng như chủ đề, từ thắc mắc, xác định, v.v. Phân tích chi tiết cho thấy sự tương ứng cụ thể giữa các vai trò đã học và các thành phần câu được xác định bởi trình phân tích cú pháp tiêu chuẩn. Một số điểm khác biệt cũng được phát hiện có lợi cho mô hình, cho thấy mô hình có khả năng học các khía cạnh ngữ pháp quan trọng sau khi tiếp xúc với văn bản, câu hỏi và câu trả lời mà không yêu cầu kiến thức ngôn ngữ trước đó.

“Evaluating the Performance of Recurrent Neural Network based Question Answering System with Easy and Complex bAbI QA Tasks” (Đánh giá hiệu suất hệ thống trả lời câu hỏi dựa trên mạng nơ-ron tái phát với các nhiệm vụ bAbI đơn giản và phức tạp) (Ramaiyan & Krishnamurthy, 2020)

Nghiên cứu này tập trung vào hệ thống trả lời câu hỏi (QA) trong lĩnh vực xử lý ngôn ngữ tự nhiên. Hệ thống QA cho phép người dùng đặt câu hỏi bằng ngôn ngữ tự nhiên và nhận câu trả lời ngắn gọn thay vì danh sách tài liệu. Công trình này đề xuất sử dụng thuật toán Deep Learning dựa trên mạng thần kinh tái phát (RNN) để giải quyết vấn đề hệ thống QA. Việc lựa chọn RNN cho phép mở rộng và áp dụng mô hình cho nhiều nhiệm vụ QA khác nhau. Hệ thống QA đơn giản dựa trên RNN được triển khai và đánh giá hiệu suất trên các tác vụ QA đơn giản và phức tạp bằng bộ dữ liệu bAbI. Mô hình được đào tạo và kiểm tra với các tập dữ liệu tương ứng, cho thấy sự khác biệt về hiệu suất giữa hai loại nhiệm vụ QA.

“The Implementation of Question Answer System Using Deep Learning” (Hệ thống trả lời câu hỏi sử dụng học sâu)(Fulmal & Al, 2021)

Trong nghiên cứu có tựa đề "The Implementation of Question Answer System Using Deep Learning" của Fulmal và Al (2021), chúng tôi tập trung vào việc phát triển một hệ thống hỏi đáp sử dụng deep learning để cung cấp câu trả lời cho các câu hỏi được đặt ra bởi người dùng. Vấn đề trả lời câu hỏi tự động là một vấn đề quan trọng trong xử lý ngôn ngữ tự nhiên. Mục tiêu của chúng tôi là xây dựng một hệ thống có khả năng tự động trả lời câu hỏi tương tự như cách con người tìm kiếm câu trả lời cho câu hỏi. Đầu tiên, hệ thống nhúng các câu hỏi từ tập dữ liệu để tạo ra biểu diễn vector. Sau đó, mạng nơ-ron nhân tạo (ANN) được đào tạo để tìm kiếm sự tương đồng giữa các câu hỏi. Hệ thống sẽ chọn ra câu trả lời có điểm tương đồng cao nhất với câu hỏi được hỏi. Mục tiêu là tạo ra một mô hình tự động trả lời câu hỏi hiệu quả. Để phân loại các câu hỏi, hệ thống sử dụng thuật toán phân cụm theo cấp bậc, giúp nhóm các câu hỏi có tính chất tương tự nhau. Nhờ vậy, hệ thống có thể cải thiện hiệu quả và độ chính xác trong việc trả lời các câu hỏi của người dùng.

“Code Mixed Question Answering Challenge using Deep Learning Methods” (Hệ thống trả lời câu hỏi hỗn hợp mã sử dụng phương pháp học sâu)(Shyam et al., 2020)

Trên mạng xã hội, việc sử dụng câu hỏi bằng ngôn ngữ hỗn hợp mã (kết hợp hai ngôn ngữ khác nhau) đã trở thành một hình thức phổ biến để thể hiện và giao tiếp. Ví dụ, trên Twitter, người dùng thường kết hợp tiếng mẹ đẻ với tiếng Anh trong các bài đăng của họ. Tuy nhiên, hệ thống trả lời câu hỏi hiện tại không hỗ trợ ngôn ngữ hỗn hợp mã này. Hầu hết các hệ thống trả lời câu hỏi hiện tại chỉ hỗ trợ một ngôn ngữ duy nhất, ví dụ như tiếng Anh, tiếng Pháp hoặc tiếng Đức. Nghiên cứu này giới thiệu một hệ thống mới có thể trả lời các câu hỏi bằng ngôn ngữ hỗn hợp mã, bao gồm tiếng Anh và các ngôn ngữ Ấn Độ như Hindi, Telugu và Tamil. Hệ thống hoạt động bằng cách:

Phân tích câu hỏi của người dùng: Hệ thống sẽ xác định các từ hỗn hợp mã trong câu hỏi và dịch sang tiếng Anh.

Phân loại câu hỏi: Hệ thống sử dụng các thuật toán học sâu như Mạng thần kinh tái phát (RNN) và Mạng chú ý phân cấp (HAN) để phân loại câu hỏi dựa trên nội dung và ngữ cảnh.

Cung cấp câu trả lời: Hệ thống sẽ tìm kiếm câu trả lời phù hợp nhất cho câu hỏi và dịch sang tiếng Anh hỗn hợp với ba ngôn ngữ Ấn Độ.

Giải thích câu trả lời: Hệ thống cung cấp thêm thông tin về cách thức và lý do đưa ra câu trả lời. Hệ thống này được triển khai dưới dạng một ứng dụng web và có thể được sử dụng trong nhiều lĩnh vực khác nhau như giáo dục, dịch vụ khách hàng và nghiên cứu.

1.3 Mục tiêu của nghiên cứu

Nghiên cứu này hướng đến xây dựng một hệ thống chatbot sử dụng học máy và học sâu để tự động trả lời các câu hỏi tiếng Việt trong một lĩnh vực bất kỳ. Hệ thống bao gồm các giai đoạn: phân tích câu hỏi, trích xuất câu trả lời,... Phân tích câu

hỏi đóng vai trò then chốt, ảnh hưởng trực tiếp đến hiệu quả hoạt động của toàn bộ hệ thống. Nếu phân tích câu hỏi không được thực hiện tốt, sẽ không thể tìm thấy câu trả lời phù hợp. Vì vậy, nhóm nghiên cứu đã chọn đề tài "Hệ thống trả lời thông tin tự động cho trường học" để thực hiện nghiên cứu.

1.4 Cách thức tiếp cận

Nghiên cứu tập trung vào các mô hình chatbot tự động sử dụng học máy và học sâu. Trong kiến trúc này, một thành phần được đặt tên là Generator, có khả năng ghi nhớ và cung cấp câu trả lời chính xác cho các câu hỏi đã được quan sát trong quá trình đào tạo. Kiến trúc này cũng bao gồm một thành phần khác được gọi là Retriever-Reader, có khả năng đưa ra câu trả lời cho các câu hỏi chưa được nhìn thấy trong tập dữ liệu huấn luyện. Cuối cùng, kiến trúc này cũng có năng lực trả lời các câu hỏi mới trong giai đoạn thử nghiệm và chọn lựa một câu trả lời từ tập hợp các câu trả lời đã được quan sát trong quá trình đào tạo, được gọi là Retriever-Generator.

1.5 Đối tượng và phạm vi tiếp cận bài toán

Đối tượng: Mô hình hệ thống trả lời câu hỏi tự động (chatbot) cùng với các mô hình học máy, học sâu và xử lý ngôn ngữ tự nhiên (cụ thể là tiếng việt).

Phạm vi: Nghiên cứu về hệ thống tư vấn hỏi đáp về trường học.

1.6 Tóm tắt nội dung sẽ trình bày trong báo cáo

Nội dung báo cáo bao gồm tổng quan về dự án như lý do chọn đề tài, các nghiên cứu liên quan, mục tiêu nghiên cứu, phương pháp cũng như đối tượng và phạm vi đề tài. Cùng với đó, là các kiến thức nền tảng liên quan tới đề tài như các mô hình RNN, LSTM,... và các bài báo có mối quan hệ với chủ đề đang tìm hiểu. Sau đó là chọn ra mô hình phù hợp với khả năng của nhóm rồi từ đó bắt đầu xử lý dữ liệu để đưa vào mô hình cùng tham số thích hợp. Sau khi mô hình được huấn luyện và đạt được các thông số khả quan, giao diện để người dùng tương tác là cần thiết. Để thực hiện điều đó, Gradio là thư viện hỗ trợ tốt cho triển khai giao diện cho người dùng tương tác với mô hình học máy. Bên cạnh đó, sử dụng flask trong python để xây dựng

API để có thể gửi yêu cầu là câu hỏi và nhận về câu trả lời. Cuối cùng là đánh giá mô hình và đưa ra kết luận cũng như hướng phát triển sau này.

CHƯƠNG 2. KIẾN THỨC NỀN TẢNG

2.1 Hệ thống trả lời câu hỏi

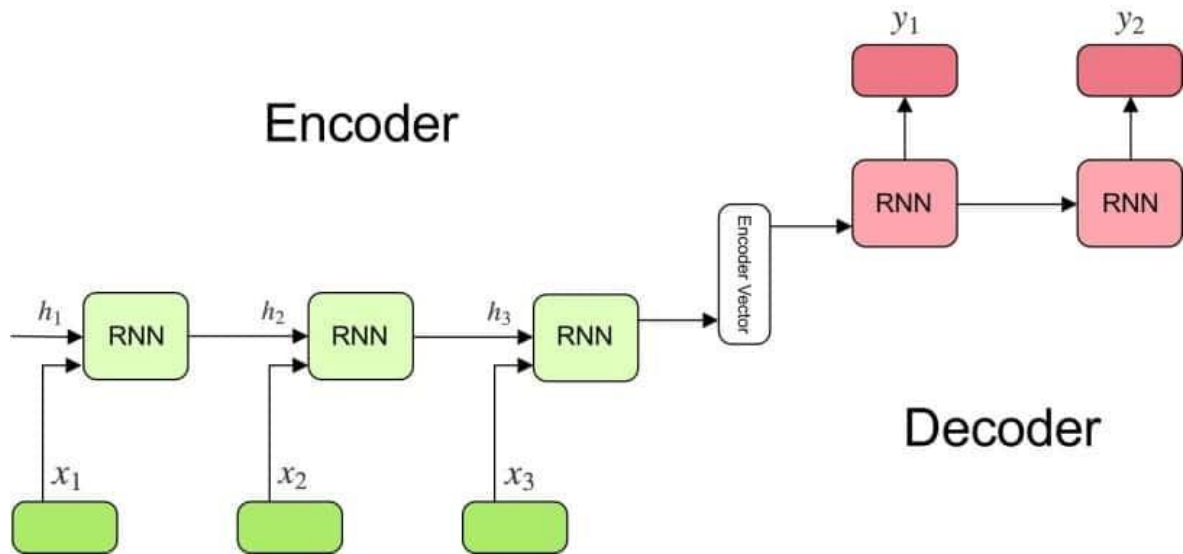
Hệ thống trả lời câu hỏi (QA) (Udani et al., 2013) là một môn khoa học máy tính quan tâm đến việc xây dựng các hệ thống tự động trả lời các câu hỏi được đặt bởi con người bằng ngôn ngữ tự nhiên. Một cách triển khai QA phổ biến là sử dụng truy vấn cơ sở dữ liệu có cấu trúc chứa kiến thức và thông tin liên quan. Một trong những hệ thống QA đầu tiên là ELIZA, được phát triển vào năm 1964. ELIZA đã thành công với ứng dụng DOCTOR, một chương trình máy tính tương tác với người dùng qua giao diện trò chuyện văn bản, đáp ứng câu hỏi và tương tác với người dùng theo cách tương tự như một nhà trị liệu (máy tính chạy ứng dụng DOCTOR). Các hệ thống QA khác được phát triển dựa trên ELIZA bao gồm BASEBALL, trả lời các câu hỏi về giải đấu bóng chày Hoa Kỳ kéo dài một năm, và LUNAR, trả lời các câu hỏi về phân tích địa chất của mẫu đá từ mặt trăng được thu thập bởi chương trình Apollo.

Hệ thống trả lời câu hỏi (QA) (Minh, n.d.) được xây dựng nhằm tìm kiếm câu trả lời cho câu hỏi được đặt bởi người dùng, bằng cách sử dụng tập tài liệu đã được huấn luyện trước đó. Mục tiêu của QA là tương tác với người dùng bằng ngôn ngữ tự nhiên (Natural Language - NL) và cung cấp câu trả lời chính xác và nhanh chóng, thay vì chỉ đưa ra các tài liệu liên quan như các công cụ tìm kiếm thông thường. Việc mở rộng và phát triển hệ thống QA sẽ cải thiện khả năng xử lý tác vụ trong hệ thống một cách hiệu quả. Một hệ thống Trả lời Câu hỏi (QA) hiện đại thường bao gồm ba thành phần chính: mô-đun xử lý truy vấn, mô-đun xử lý dữ liệu và mô-đun chiết xuất phản hồi (Udani et al., 2013). Để nâng cao hiệu năng của các hệ thống này, các kỹ thuật Trí tuệ Nhân tạo (AI) và các thuật toán Học Có Giám sát (Supervised Learning) và Học Không Giám sát (Unsupervised Learning) đã được ứng dụng rộng rãi. Tuy nhiên, việc xử lý Ngôn ngữ Tự nhiên trong các hệ thống Trả lời Câu hỏi vẫn còn gặp phải nhiều khó khăn và thách thức đáng kể cần được giải quyết. Mặc dù vậy, trong những năm gần đây, lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) đã phát triển mạnh mẽ

và nhanh chóng, đồng thời các công nghệ trí tuệ nhân tạo (AI) cũng đã giúp cải thiện độ chính xác của câu trả lời và sự tự nhiên trong cách thể hiện ngôn ngữ.

2.2 Mô hình đặt câu hỏi và trả lời tự động

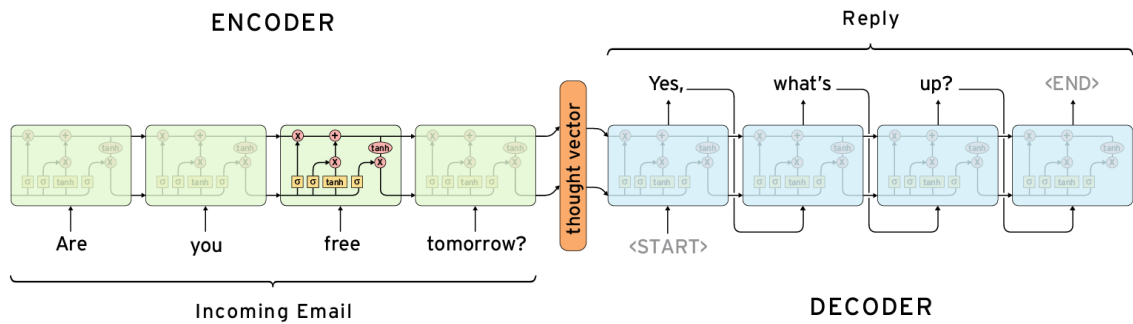
seq2seq (Dušek & Jurcicek, 2016; trituenhantao.io, 2019) sẽ có hai mạng RNN: Một mạng là bộ mã hóa (Encoder), một mạng còn lại là bộ giải mã (Decoder) và còn có véc-tơ mã hóa trung gian (Encoder vector).



Hình 2. 1: Kiến trúc của mô hình encoder-decoder seq2seq

(Nguồn: (trituenhantao.io, 2019))

Bộ mã hóa có nhiệm vụ là nhận chuỗi đầu vào để xử lý từng phần tử ở trong câu, mã hóa các thông tin cần thiết quan trọng có trong chuỗi đầu vào. Sau đó sẽ chuyển đổi các phần tử này thành véc-tơ đặc trưng với một kích thước cố định. Véc-tơ mã hóa trung gian thì nó là một trạng thái ẩn và nằm ở cuối bộ mã hóa và là trạng thái ẩn đầu tiên của bộ giải mã. Để giúp cho bộ giải mã dự đoán một cách chính xác nhất thì véc-tơ này có chức năng tập hợp các thông tin từ đầu vào để làm việc này và đưa nó làm trạng thái ẩn cho đầu vào của bộ giải mã. Đầu ra sẽ được tạo ra khi nhận đầu vào từ trạng thái ẩn của chính nó hoặc trạng thái ẩn trước đó và mô hình này có thể ánh xạ từ các chuỗi đầu vào thành các chuỗi đầu ra với kích thước không giống nhau.



Hình 2. 2: Cách thức hoạt động của mô hình seq2seq

(Nguồn: (NLP [P3] - Seq2Seq Model - Hạt Nhân Của Google Translate, 2019))

Một số khó khăn khi sử dụng mô hình seq2seq là mô hình không thể xử lý được chuỗi quá dài và sẽ xử lý rất chậm khi huấn luyện mô hình vì bộ giải mã chạy hàm softmax coi một tập dữ liệu rất lớn cho từng từ được tạo ra.

2.3 Phân loại các mô hình đặt và trả lời câu hỏi

2.3.1 Phân loại hệ thống QA theo miền (Mervin, 2013)

Các hệ thống Trả lời Câu hỏi (QA) có thể được phân loại thành hai nhóm chính dựa trên phạm vi hoạt động. Nhóm thứ nhất là các hệ thống QA miền mở (Open domain) được thiết kế để tìm kiếm câu trả lời cho các câu hỏi bằng ngôn ngữ tự nhiên từ một nguồn dữ liệu lớn và đa dạng. Quá trình hoạt động của chúng bao gồm việc lọc dữ liệu, xử lý thông tin và trích xuất câu trả lời phù hợp. Các hệ thống này có khả năng xử lý nhiều loại câu hỏi khác nhau, dựa trên cơ sở tri thức chung và bối cảnh toàn cầu. Tuy nhiên, để đạt được hiệu quả cao, chúng thường yêu cầu một lượng dữ liệu đầu vào lớn. Nhóm thứ hai là các hệ thống QA miền đóng (Close domain), chuyên xử lý các câu hỏi trong một lĩnh vực cụ thể, được coi là nhiệm vụ đơn giản hơn so với nhóm trước.

Điều này bởi quá trình xử lý Ngôn ngữ Tự nhiên có thể tận dụng kiến thức trong một miền cụ thể đó. Tuy nhiên, trong một số trường hợp cụ thể, hệ thống QA miền đóng chỉ có thể đáp ứng một số câu hỏi hạn chế, bởi vì hệ thống chỉ dựa trên một nguồn thông tin hạn chế để cung cấp câu trả lời.

2.3.2 Phân loại mô hình với nhiều hướng

Có hai cách tiếp cận chính để xây dựng các hệ thống Trả lời Câu hỏi (QA). Phương pháp thứ nhất là *Mô hình Sinh*, trong đó hệ thống không phụ thuộc vào một tập dữ liệu được xác định trước, mà có khả năng tổng hợp câu trả lời từ đầu. Các mô hình này sử dụng các kỹ thuật tương tự như Dịch Máy, nhưng thay vì dịch từ ngôn ngữ này sang ngôn ngữ khác, chúng chuyển đổi từ câu hỏi đầu vào thành câu trả lời đầu ra. Phương pháp thứ hai là *Mô hình Trích* (Retrieval-based) (Minh, n.d.; Pandey & Sharma, 2023), trong đó hệ thống sử dụng một nguồn dữ liệu đã được xây dựng sẵn để tìm kiếm câu trả lời. Các thuật toán Truy Xuất Thông tin được sử dụng để lọc và chọn lọc câu trả lời phù hợp nhất dựa trên ngữ cảnh và nội dung của câu hỏi.

Kỹ thuật heuristic được sử dụng có thể đơn giản, như so sánh các biểu thức trùng khớp dựa trên luật, hoặc phức tạp hơn như sử dụng học máy để phân loại câu hỏi và câu trả lời. Tuy nhiên, các hệ thống này không tạo ra văn bản mới, mà chỉ sử dụng câu trả lời từ tập dữ liệu cố định đã có sẵn.

Mô hình trích chọn thông tin dựa vào một kho dữ liệu đã được định nghĩa trước. Mô hình này tập trung vào trích chọn thông tin mà không gặp các vấn đề về ngữ pháp. Tuy nhiên, nó không thể xử lý các trường hợp chưa được giám sát hoặc không có trong bộ luật huấn luyện. Do đó, mô hình này không thể ghi nhớ các thông tin ngữ cảnh trước đó đã được đề cập trong cuộc trò chuyện.

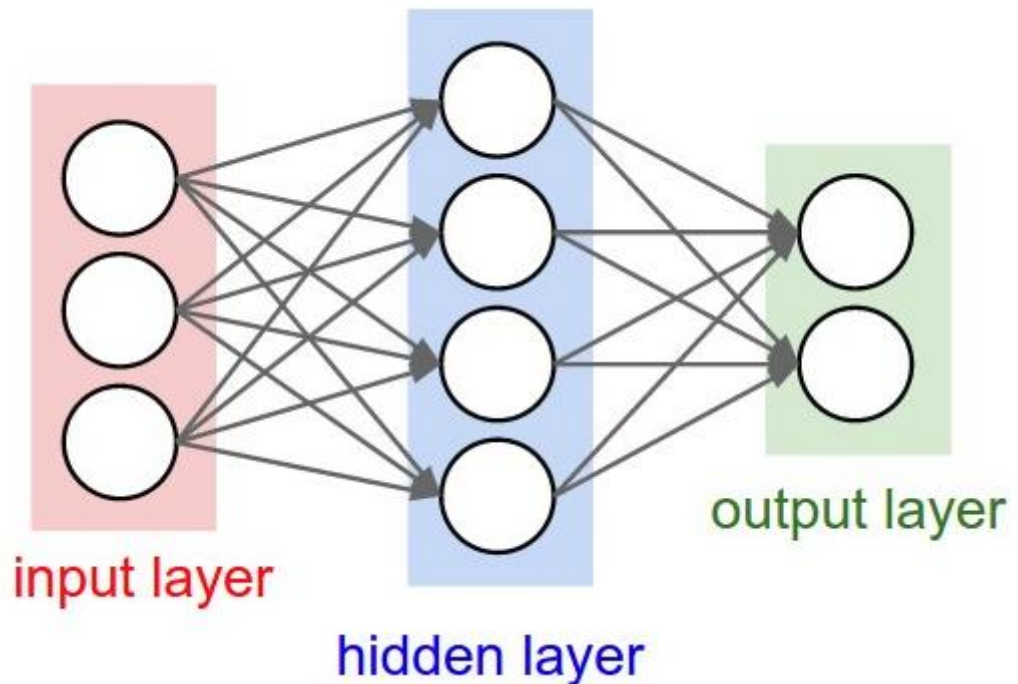
Mô hình sinh thông minh hơn mô hình trích chọn thông tin vì nó có khả năng ghi nhớ các thực thể hoặc ngữ cảnh trước đó đã được đề cập trong câu hỏi. Khi tương tác với mô hình này, ta có cảm giác như đang tương tác với một con người. Tuy vậy mô hình này huấn luyện nó rất khó khăn và yêu cầu nhiều dữ liệu, đồng thời dễ gặp lỗi về ngữ pháp, đặc biệt là đối với các câu dài.

2.4 Tổng quan về Neural Network

Neural network (NN) là một chuỗi các thuật toán được đưa ra nhằm tìm kiếm các mối quan hệ cơ bản trong tập hợp các dữ liệu. Cũng có thể tưởng tượng cách hoạt động của nó gần giống với việc bắt chước cách thức hoạt động của não bộ của con người. Ngoài ra, NN còn có thể thích ứng với việc thay đổi đầu vào. Chính vì thế nó

có thể đưa ra được kết quả một cách tốt nhất có thể mà không cần phải thay đổi các tiêu chí đầu ra.

Một Mạng Nơ-ron (Neural Network) là một mô hình toán học lấy cảm hứng từ cách thức hoạt động của hệ thần kinh sinh học. Nó được cấu thành từ nhiều đơn vị xử lý đơn giản (tương tự như các nơ-ron trong não) được sắp xếp thành các lớp kết nối với nhau. Mỗi Mạng Nơ-ron điển hình bao gồm ba loại lớp chính: Lớp Đầu vào (Input Layer) nhận dữ liệu từ bên ngoài, Lớp Đầu ra (Output Layer) sản sinh kết quả cuối cùng, và Lớp Ẩn (Hidden Layer) thực hiện các phép tính trung gian. Trong khi một Mạng Nơ-ron chỉ có một lớp đầu vào và một lớp đầu ra duy nhất, số lượng các lớp ẩn có thể thay đổi tùy theo độ phức tạp của mô hình.



Hình 2. 3: Kiến trúc Neural Network

Mỗi nút mạng là một sigmoid nơon nhưng chúng lại có hàm kích hoạt khác nhau. Nhưng trong thực tế thì chúng thường sẽ được sử dụng giống nhau để thuận lợi cho tính toán hơn. Ở mỗi tầng thì số lượng nút mạng có thể sẽ khác nhau vì còn phải phụ thuộc vào yêu cầu cách giải quyết và bài toán. Ngoài ra các nơon ở các tầng thường được liên kết đôi một với nhau để tạo thành mạng kết nối đầy đủ (full-

connected network). Lúc đó sẽ có thể tính được kích cỡ của mạng dựa vào số tầng và số neuron.

2.5 Recurrent Neural Network (RNN)

Recurrent neural network (RNN) là một trong những mô hình trong lĩnh vực trí tuệ nhân tạo (AI) và học máy (machine learning - ML) được thiết kế để xử lý dữ liệu có tính tuần tự hoặc chuỗi. Ngoài ra, RNN còn có thể nhớ các thông tin của quá khứ và sử dụng nó để ảnh hưởng đến quyết định được đưa ra tại thời điểm hiện tại. Chính vì điều này nên RNN sẽ phù hợp cho các nhiệm vụ liên quan đến chuỗi thời gian, NLP và các ứng dụng khác.

Một số thành phần chính của RNN:

- Nút đầu vào (input node) sẽ nhận dữ liệu đầu vào từ thời điểm t .
- Nút ẩn (hidden node) sẽ lưu trữ thông tin từ các thời điểm trước đó và sẽ chuyển nó tiếp cho các thời điểm sau. Đây cũng là nơi mà mô hình sẽ giữ lại “bộ nhớ” về các sự kiện quan trọng ở quá khứ.
- Nút đầu ra (output node) sẽ tạo ra dự đoán hoặc sẽ xuất ra kết quả từ các thông tin đã thu thập trước đó.

Trong mô hình RNN, chuỗi đầu vào được biểu diễn dưới dạng một chuỗi các vector $\{u_1, u_2, \dots, u_m\}$, và chuỗi các trạng thái ẩn được biểu diễn dưới dạng $\{v_1, v_2, \dots, v_m\}$. Mỗi đơn vị đầu vào tương ứng được kí hiệu là $\{z_1, z_2, \dots, z_m\}$. Lớp hồi quy trong mô hình này sử dụng hàm tái quy d , nhận đầu vào là vector u_x và trạng thái ẩn của trạng thái trước đó v_x , và sau đó tạo ra trạng thái ẩn mới như sau:

$$v_x = d(u_t, v_{t-1}) = \tanh(Pu_t + Qv_{t-1})$$

Các đầu ra sẽ được tính như sau:

$$z_x = \text{softmax}(Rv_t)$$

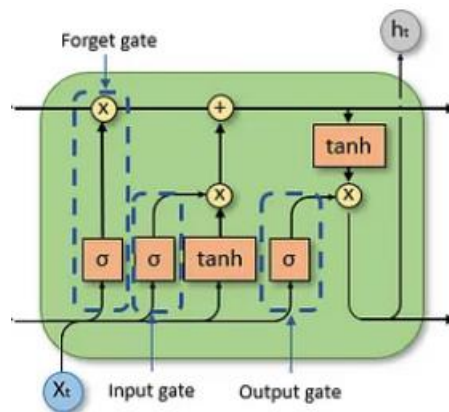
Trong mô hình RNN, chúng ta sử dụng các ma trận trọng số P, Q, R và hàm kích hoạt \tanh để biểu diễn hàm tiếp tuyến hợp phổ biến. Mô hình RNN sử dụng các chức năng phức tạp để hiểu và điều khiển luồng thông tin trong các lớp lặp lại, nhằm nắm bắt các phụ thuộc dài hạn.

Quan trọng nhất, RNN có khả năng chia sẻ trọng số giữa các bước thời gian, điều này giúp chúng nắm bắt thông tin từ quá khứ và duy trì trạng thái ẩn trong quá trình xử lý chuỗi dữ liệu. Mặc dù RNN mang lại nhiều lợi ích trong việc xử lý chuỗi, nhưng chúng cũng đối mặt với các vấn đề “vanishing gradient” khi huấn luyện trên các chuỗi dài. Do đó đã dẫn đến sự phát triển của các kiến trúc mô hình khác như Long Short-Term Memory và Gated Recurrent Unit chúng sẽ giúp khắc phục nhược điểm này và cải thiện hiệu suất cho mô hình RNN.

2.6 Long Short-Term Memory (LSTM) và Gated Recurrent Unit (GRU)

2.6.1 Long Short-Term Memory (Dancker, 2022; Fouzan, 2023)

Long Short-Term Memory (LSTM) là một loại kiến trúc mạng nơ-ron thuộc họ RNN, nó được thiết kế để giải quyết vấn đề “vanishing gradient” gặp phải trong mô hình RNN khi xử lý với chuỗi dữ liệu dài. LSTM giúp cho mô hình RNN có khả năng học và nhớ thông tin trong thời gian dài mà không gặp phải vấn đề mất mát gradient quá nhanh.



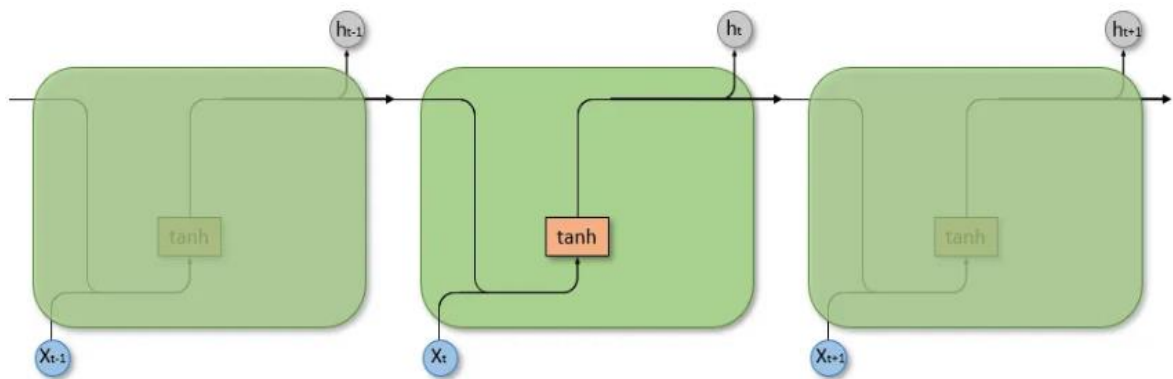
Hình 2. 4: Các thành phần chính của LSTM
(Nguồn (Dancker, 2022))

Các thành phần chính của LSTM (như hình 2.4) bao gồm:

- Cổng quên (Forget gate): Quyết định thông tin nào sẽ được giữ lại và thông tin nào sẽ bị quên từ trạng thái ô trước đó.

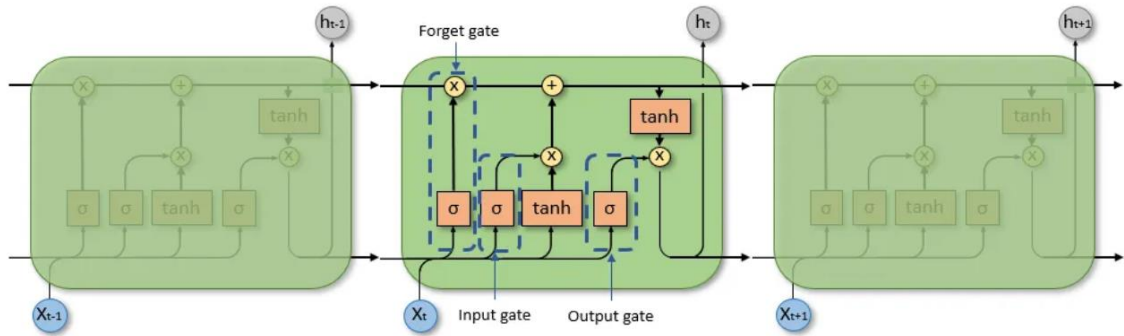
- Cổng đầu vào (Input gate): Quyết định thông tin mới nào nên được thêm vào trạng thái ô.
- Trạng thái ô (Cell state): Là một dạng của bộ nhớ, nó giúp cho mô hình giữ lại và truyền thông tin qua các thời điểm mà không bị ảnh hưởng quá nhiều bởi gradient.
- Cổng đầu ra (Output gate): Quyết định thông tin nào nên truyền đi từ trạng thái ô đến đầu ra của mô hình.

Mạng nơ-ron tuần hoàn được cấu thành từ chuỗi các mô-đun lặp lại, và đặc biệt trong RNN tiêu chuẩn, các mô-đun lặp lại này có cấu trúc đơn giản, ví dụ như chỉ sử dụng một lớp tanh.



Hình 2. 5: Các mô-đun lặp lại trong mạng RNN chỉ chứa một lớp
(Nguồn (Dancker, 2022))

Cấu trúc của các LSTM cũng tương tự như chuỗi được mô tả trong hình 2.5, tuy nhiên, nó khác biệt với một cấu trúc đặc biệt. Thay vì chỉ gồm một lớp như trong hình 2.5, LSTM lại có bốn lớp tương tác đặc biệt với nhau, như được minh họa chi tiết trong hình 2.6.



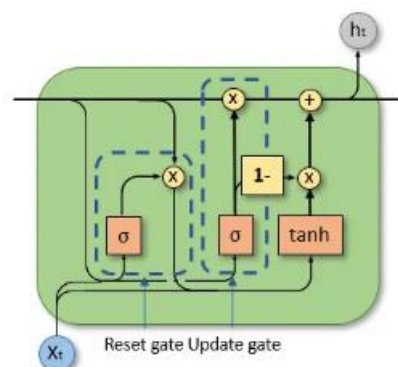
Hình 2. 6: Các mô-đun lặp lại trong một LSTM

(Nguồn(Dancker, 2022))

LSTM có khả năng học các mô hình phức tạp của chuỗi dữ liệu, nhưng vẫn có thể giữ được khả năng xử lý thông tin từ quá khứ. Điều này giúp cho LSTM trở thành một sự lựa chọn phổ biến trong nhiều ứng dụng như xử lý ngôn ngữ tự nhiên, dịch máy và dự đoán chuỗi thời gian. Nhờ vào cấu trúc đó nên LSTM đã giúp nâng cao khả năng mô hình và dự đoán chuỗi dữ liệu so với RNN truyền thống.

2.6.2 Gated Recurrent Unit (Dancker, 2022; Fouzan, 2023)

Gated Recurrent Unit (GRU) cũng là một kiến trúc mạng nơ-ron thuộc họ RNN, nó cũng giống LSTM, GRU được thiết kế để giải quyết vấn đề “vanishing gradient” trong mô hình RNN và đồng thời cung cấp một cách tiếp cận hiệu quả về xử lý chuỗi dữ liệu.



Hình 2. 7: Kiến trúc của GRU

(Nguồn(Dancker, 2022))

Một số thành phần chính của GRU:

- **Trạng thái ẩn (Hidden state):** là nơi lưu trữ thông tin từ quá khứ và giữ lại trạng thái ẩn để ảnh hưởng đến quyết định tại thời điểm hiện tại.
- **Cổng đầu vào (Update gate):** Quyết định mức độ thông tin mới nên được thêm vào trạng thái ẩn.
- **Cổng reset (Reset gate):** Quyết định mức độ thông tin nào từ trạng thái ẩn cũ nên được reset.

2.6.3 So sánh RNN, GRU và LSTM

Bảng 2. 1: So sánh RNN, GRU và LSTM

Đặc điểm	RNN	GRU	LSTM
Cấu trúc	Đơn giản, không có cổng	Đơn giản, có hai cổng	Phức tạp, nhiều cổng
Tốc độ	Nhanh	Nhanh	Chậm
Hiệu quả	Thấp	Cao	Cao
Khả năng ghi nhớ	Ngắn hạn	Ngắn hạn tốt, dài hạn không tốt	Dài hạn
Khả năng học tập	Dễ dàng	Dễ dàng	Khó khăn
Yêu cầu dữ liệu	Ít	Ít	Nhiều
Phù hợp với	Các ứng dụng đơn giản, ngắn hạn	Các ứng dụng cần tốc độ và hiệu quả	Các ứng dụng cần ghi nhớ các thông tin dài hạn

Lựa chọn sử dụng RNN, GRU hay LSTM sẽ phụ thuộc vào nhu cầu cụ thể của từng ứng dụng. Nếu cần một mạng nơ-ron đơn giản, hiệu quả và nhanh chóng thì RNN sẽ là một sự lựa chọn rất phù hợp. Nếu cần lựa chọn cân bằng giữa tốc độ, hiệu quả và khả năng ghi nhớ thì chọn GRU. LSTM sẽ là một mạng nơ-ron có khả năng ghi nhớ thông tin dài hạn tốt hơn.

2.7 Embedding và Keras Embedding Layer (*Embedding — Machine Learning Cho Dữ Liệu Dạng Bảng, n.d.; Team, n.d.*)

Embedding là một kỹ thuật chuyển đổi một vector có số chiều lớn và thưa thành một vector có số chiều nhỏ và dày đặc. Đây là một phương pháp hữu ích, đặc biệt khi xử lý các đặc trưng hạng mục với số lượng phân tử lớn. Thông thường, mỗi giá trị trong đặc trưng này được biểu diễn bằng một vector one-hot. Trong không gian embedding, các giá trị có ý nghĩa tương tự thường được gần nhau. Ví dụ đáng chú ý nhất là việc biểu diễn các từ trong một từ điển dưới dạng số. Khi từ điển có hàng triệu từ, biểu diễn từng từ bằng vector one-hot dẫn đến số chiều cực kỳ lớn. Hơn nữa, các từ này sẽ có khoảng cách đồng đều tới tất cả các từ khác, gây thiếu thông tin giá trị cho việc huấn luyện mô hình machine learning. Để xây dựng một mô hình xử lý ngôn ngữ tự nhiên hiệu quả, việc biểu diễn các từ vựng một cách phù hợp là rất quan trọng. Điều này đòi hỏi phải bắt được mối liên hệ giữa những từ có ý nghĩa tương đồng, chẳng hạn như cặp từ "vua" và "hoàng hậu", hoặc "chồng" và "vợ". Trong lĩnh vực học sâu, các thư viện như Keras cung cấp một lớp đặc biệt gọi là Embedding Layer, cho phép nhúng các từ vào một không gian véc-tơ đa chiều, giúp bảo toàn mối tương quan ngữ nghĩa giữa chúng. Kỹ thuật này đóng vai trò quan trọng trong việc xây dựng các mạng nơ-ron để xử lý dữ liệu dạng văn bản một cách hiệu quả. Đầu vào của layer này là các số nguyên đã được mã hóa, trong đó mỗi từ được biểu diễn bằng một số nguyên duy nhất.

Keras cũng cung cấp Tokenizer API để tiện lợi trong quá trình này. Lớp Embedding được khởi tạo ban đầu với các trọng số ngẫu nhiên, nhưng trong

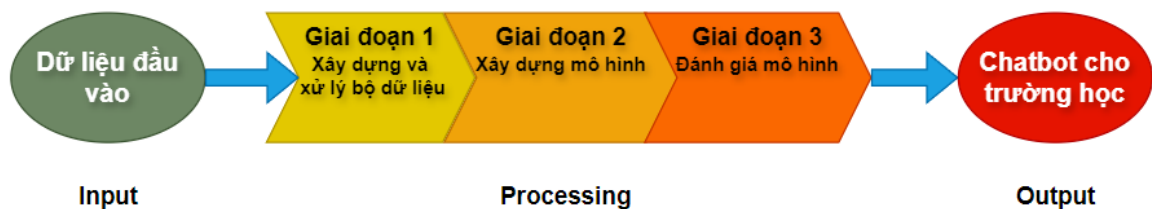
quá trình huấn luyện mô hình, nó sẽ học cách biểu diễn các từ vựng dưới dạng các véc-tơ nhúng phù hợp với ngữ cảnh và ý nghĩa của chúng trong tập dữ liệu đầu vào. Các thông số cơ bản của Embedding layer bao gồm:

- `input_dim`: Số lượng của từ điển trong dữ liệu đầu vào. Nếu kích thước của dữ liệu đầu vào là n , thì kích thước của từ điển là $n+1$.
- `output_dim`: Độ dài của vector tương ứng với mỗi từ.
- `input_length`: Độ dài của chuỗi đầu vào.

CHƯƠNG 3. XÂY DỰNG BỘ DỮ LIỆU VÀ MÔ HÌNH CHO BÀI TOÁN QAS

3.1 Đề xuất phương pháp cho bài toán

Trước khi huấn luyện mô hình, chúng ta cần chuẩn bị tập dữ liệu đầu vào bằng cách biến đổi các chuỗi văn bản thô thành định dạng phù hợp cho quá trình học máy. Sau đó, chúng ta sẽ khám phá phương pháp đề xuất để thực hiện mô hình cho bài toán này, sử dụng các thư viện như Tensorflow và Keras trong lĩnh vực Machine Learning. Cuối cùng, chúng ta sẽ đánh giá kết quả và thực hiện thử nghiệm thực tế để kiểm tra hiệu suất của mô hình.



Hình 3. 1: Thuật toán xử lý cho phương pháp đã đề xuất

3.2 Input: Dữ liệu về trường học

Trong quá trình huấn luyện mô hình, chúng tôi đã sử dụng dữ liệu từ trường Đại học Tôn Đức Thắng trong khoảng thời gian từ năm 2023 đến 2024 để làm dữ liệu huấn luyện. Đầu tiên, chúng tôi đã nghiên cứu các vấn đề phổ biến thường được hỏi khi tìm hiểu về trường Đại học Tôn Đức Thắng. Tiếp theo, chúng tôi đã thu thập thông tin từ các trang web chính thức của trường. Cuối cùng, chúng tôi đã xây dựng được một tập hợp câu hỏi và câu trả lời để sử dụng trong quá trình huấn luyện mô hình.

3.3 Processing

Gồm có 3 giai đoạn sau:

- Giai đoạn 1: Từ văn bản chưa qua xử lý đầu vào đã xây dựng dữ liệu với cấu trúc có 3 thành phần gồm: cauhoi là các câu hỏi mà người dùng

có thể sẽ đặt ra cho hệ thống, cautraloi sẽ là chứa các câu trả lời tương ứng với các câu hỏi ở cột cauhoi, tags sẽ chứa các nhãn của cuộc hội thoại. Từ ba thành phần trên sẽ tạo nên tệp dữ liệu cho việc huấn luyện mô hình. Kế tiếp sẽ tiến hành làm sạch dữ liệu cho bộ dữ liệu. Tiếp đến sẽ mã hóa dữ liệu bằng các phương thức được hỗ trợ bởi Keras của Tensorflow trước khi đưa vào huấn luyện.

- Giai đoạn 2: chúng ta sẽ xây dựng một mô hình dự đoán cho bài toán. Mô hình được chọn là LSTM, một kiểu mô hình trong lĩnh vực RNN (Recurrent Neural Network Kiến trúc LSTM sẽ đảm nhận vai trò xử lý và học tập thông tin từ các câu hỏi đầu vào trong giai đoạn đầu tiên, nhằm hỗ trợ việc dự đoán câu trả lời một cách chính xác. Để làm phẳng đầu ra của dữ liệu, chúng ta sử dụng lớp Flatten. Đồng thời, để xử lý chuỗi văn bản sau khi đi qua lớp LSTM, chúng ta sử dụng Embedding của Keras.
- Giai đoạn 3: Trong giai đoạn này, chúng ta sẽ đánh giá độ chính xác của mô hình. Để đánh giá mức độ chính xác, chúng ta sử dụng chỉ số accuracy (độ chính xác). Một giá trị accuracy cao cho thấy mô hình đạt được độ chính xác cao.

3.4 Output: Câu trả lời

Mô hình đã xây dựng sẽ được áp dụng vào việc phát triển chatbot. Sau đó, chúng ta sẽ tiến hành kiểm thử để đánh giá kết quả. Đáng kể là mô hình đã hoạt động rất hiệu quả, đưa ra các câu trả lời đúng hoặc gần đúng với các câu hỏi được đặt bởi người dùng.

CHƯƠNG 4. THỰC NGHIỆM

4.1 Dataset

4.1.1 Cơ sở của bộ dữ liệu

Mục tiêu là xây dựng hệ thống trả lời tự động về trường học, bộ dữ liệu sẽ xoay quanh thông tin tuyển sinh của trường đại học Tôn Đức thắng năm 2023-2024. Các vấn đề thường gặp khi mà các phụ huynh hoặc học sinh muốn tìm hiểu về trường, ngành học, tuyển sinh... Nhóm đã tìm hiểu các vấn đề này và đặt ra câu hỏi và câu trả lời để giải đáp cho các vấn đề liên quan.

4.1.2 Cấu trúc bộ dữ liệu

Các thông tin đã thu được ở trong quá trình tìm hiểu về lý thuyết ở trước đó. Ở bước này sẽ xây dựng bộ dữ liệu để huấn luyện cho mô hình. Bộ dữ liệu sẽ là file CSV chứa các câu hỏi và câu trả lời về một vấn đề. Mỗi dòng sẽ gồm có 3 thành phần:

- cauhoi : Là những thắc mắc mà người dùng đang có nhu cầu cần được giải đáp cho hệ thống.
- cautraloi: Là những câu trả lời tương ứng với các câu hỏi mà người dùng đã đặt.
- tags: Là nhãn đã được phân loại các câu trả lời tương ứng, dùng để ánh xạ với một câu trả lời cụ thể.

Nhờ cấu trúc trên, chúng tôi đã xây dựng hơn 2000 câu hỏi mà người dùng có thể đặt cho chatbot, cùng với các câu trả lời tương ứng cho mỗi câu hỏi đó. Để hiểu rõ hơn về cấu trúc dữ liệu đầu vào, bạn có thể tham khảo hình minh họa 4.1. Trong đó, mỗi câu hỏi đưa ra bởi người dùng sẽ được gán nhãn phân loại tương ứng.


```

                                cauhoi \
0                                bảo hộ lao động
1                                Ngành bảo hộ lao động là gì?
2                                Thông tin về ngành bảo hộ lao động?
3                                Giới thiệu về ngành bảo hộ lao động?
4                                Kỹ sư ngành bảo hộ lao động làm gì?
...
1738                                chỉ phí trung bình
1739                                chỉ phí trung bình 4 năm học
1740                                chỉ phí trung bình qua 4 năm
1741                                chỉ phí trung bình các ngành
1742                                chỉ phí trung bình ngành khoa học máy tính

                                cautraloi \
0                                Bảo hộ lao động là ngành học đón đầu xu hướng ...
1                                Bảo hộ lao động là ngành học đón đầu xu hướng ...
2                                Bảo hộ lao động là ngành học đón đầu xu hướng ...
3                                Bảo hộ lao động là ngành học đón đầu xu hướng ...
4                                Kỹ sư bảo hộ lao động có thể tự tin ứng tuyển ...
...
1738                                Mức học phí trung bình dự kiến năm 2024: https...
1739                                Mức học phí trung bình dự kiến năm 2024: https...
1740                                Mức học phí trung bình dự kiến năm 2024: https...
1741                                Mức học phí trung bình dự kiến năm 2024: https...
1742                                Mức học phí trung bình dự kiến năm 2024: https...

                                tags
0                                gioithieunganhbaoholaodong
1                                gioithieunganhbaoholaodong
2                                gioithieunganhbaoholaodong
3                                gioithieunganhbaoholaodong
4                                nganhbaoholaodonglamgi
...
1738                                hocphi2024
1739                                hocphi2024
1740                                hocphi2024
1741                                hocphi2024
1742                                hocphi2024

[2643 rows x 3 columns]

```

Hình 4. 1: Bộ dữ liệu

4.1.3 Tiền xử lý dữ liệu

Bước quan trọng trước khi huấn luyện mô hình: Tiền xử lý dữ liệu đóng vai trò then chốt trong việc đảm bảo độ chính xác cho bất kỳ bài toán học máy nào. Dữ liệu cần được xử lý, làm sạch và biến đổi để phù hợp với yêu cầu của mô hình.

Làm sạch dữ liệu đầu vào: Để đảm bảo dữ liệu đầu vào từ người dùng đáp ứng tiêu chuẩn, chúng ta thực hiện hai bước làm sạch sau:

1. Loại bỏ ký tự đặc biệt: Xóa các dấu câu và ký tự đặc biệt khỏi câu hỏi, bao gồm: !"#\$%&'()*+,-./:;<=>?@[^_`{|}~ Việc loại bỏ các ký tự này giúp đơn giản hóa dữ liệu và tập trung vào nội dung chính của câu hỏi.

2. Chuyển đổi sang chữ thường: Chuyển đổi tất cả chữ cái trong câu hỏi sang chữ thường. Việc này giúp loại bỏ sự khác biệt về cách viết hoa, viết thường, đồng thời giúp mô hình tập trung vào ngữ nghĩa của câu hỏi.

Sau đó, áp dụng thuật toán tokenizer (`Tf.Keras.Preprocessing.Text.Tokenizer` | `TensorFlow v2.15.0.Post1`, n.d.) - một nhánh con trong xử lý ngôn ngữ tự nhiên. Thuật toán này cho phép biến đổi một kho dữ liệu văn bản thành dạng véc-tơ. Cụ thể, mỗi văn bản sẽ được chuyển đổi thành một chuỗi các số nguyên, với mỗi số nguyên tương ứng với chỉ mục của mã thông báo trong từ điển. Ngoài ra, tokenizer cũng có thể biểu diễn văn bản thành một véc-tơ có hệ số cho mỗi mã thông báo. Hệ số này có thể là nhị phân, dựa trên số từ, dựa trên tf-idf, hoặc dựa trên các phương pháp khác.

- Tokenizer: là thuật toán được sử dụng để chia nhỏ văn bản thành các đơn vị nhỏ hơn gọi là mã thông báo. Các mã thông báo này có thể là từ, âm tiết, hoặc các đơn vị khác.
- Véc-tơ hóa: là quá trình chuyển đổi văn bản thành dạng véc-tơ, trong đó mỗi mã thông báo được biểu diễn bằng một số.
- Từ điển: là tập hợp các mã thông báo cùng với chỉ mục tương ứng của chúng.
- Hệ số: là giá trị được gán cho mỗi mã thông báo trong véc-tơ. Hệ số này có thể thể hiện nhiều thông tin khác nhau, chẳng hạn như tần suất xuất hiện của mã thông báo trong văn bản, hoặc mức độ quan trọng của mã thông báo đối với văn bản.

Từ bộ dữ liệu đã tạo sẵn, ta chọn lọc 5000 từ có tần suất xuất hiện cao nhất để sử dụng. Việc này giúp loại bỏ những từ ít phổ biến và giảm kích thước của bộ dữ liệu. Sau đó, ta cập nhật danh sách từ vựng bằng cách thêm các từ xuất hiện trong các câu hỏi của bộ dữ liệu. Tiếp theo, mỗi câu hỏi sẽ được chuyển đổi thành một chuỗi các số nguyên, với mỗi số nguyên đại diện cho một từ trong danh sách từ vựng. Do số lượng từ trong mỗi câu hỏi có thể khác nhau, ta sử dụng hàm `pad_sequences` (`Tf.Keras.Utils.Pad_sequences` | `TensorFlow v2.15.0.Post1`, n.d.) của Keras để tạo ra các chuỗi có độ dài bằng nhau. Cuối cùng, ta sử dụng bộ mã hóa `LabelEncoder`

(*Sklearn.Preprocessing.LabelEncoder*, n.d.) từ thư viện scikit-learn để mã hóa các nhãn (tags) thành các số nguyên.

- Lọc từ vựng: Giữ lại 5000 từ có tần suất xuất hiện cao nhất giúp loại bỏ những từ ít phổ biến, ít ảnh hưởng đến kết quả và giảm kích thước của bộ dữ liệu.
- Cập nhật từ vựng: Thêm các từ xuất hiện trong câu hỏi giúp đảm bảo rằng tất cả các từ trong câu hỏi đều được mã hóa chính xác.
- Véc-tơ hóa câu hỏi: Chuyển đổi câu hỏi thành chuỗi số nguyên giúp máy tính dễ dàng xử lý và phân tích.
- Cân bằng độ dài câu hỏi: Sử dụng `pad_sequences` giúp tạo ra các chuỗi có độ dài bằng nhau, tạo điều kiện cho việc huấn luyện mô hình.
- Mã hóa nhãn: Chuyển đổi nhãn thành số nguyên giúp đơn giản hóa quá trình xử lý.

```
input_shape = X_train.shape[1]
print(X_train.shape)
print(input_shape)

(2643, 36)
36

#vocab là các từ trong bộ dữ liệu và output_length là số lượng tag
vocabulary = len(tokenizer.word_index)
print("number of unique words: ", vocabulary)
output_length = le.classes_.shape[0]
print("output length: ", output_length)

number of unique words: 320
output length: 385
```

Hình 4. 2: Xử lý dữ liệu

Sau khi thực hiện các bước trên sẽ nhận được số chiều của bộ dữ liệu là 36, với 320 từ duy nhất và có 385 kết quả đầu ra.

4.2 Khởi tạo mô hình

Mô hình có đầu vào có số chiều đã được tính như đã đề cập ở trên. Để tiếp tục, chúng ta sẽ thêm một lớp Embedding để tạo ra các vector nhúng cho mỗi từ trong câu. Sau đó, các vector này sẽ được đưa vào một lớp recurrent với cổng LSTM. Để tránh overfitting, chúng ta sẽ thêm một lớp Dropout. Kế tiếp, đầu ra sẽ được đưa vào

một lớp Flatten để chuyển đổi đầu ra của lớp LSTM từ mảng nhiều chiều thành mảng một chiều. Cuối cùng, chúng ta sẽ sử dụng một lớp Dense để làm đầu ra của mô hình với hàm kích hoạt là softmax. Sau khi xây dựng mô hình thành công, chúng ta tiến hành biên dịch mô hình với các thiết lập về các thông số, bao gồm hàm mất mát là `sparse_categorical_crossentropy`, thuật toán tối ưu adam, và chỉ số độ chính xác (accuracy) để quan sát. Tiếp theo, chúng ta chạy quá trình huấn luyện mô hình với số epochs là 200. Kết quả của quá trình này có thể được thấy trong hình 4.3.

Mô hình được thiết kế với cấu trúc sau:

- Lớp đầu vào: Nhận dữ liệu đầu vào với số chiều đã được xác định trước đó. Lớp Embedding: Chuyển đổi mỗi từ trong câu thành véc-tơ nhúng.
- Lớp Recurrent (LSTM): Xử lý các véc-tơ nhúng theo trình tự thời gian.
- Lớp Dropout: Giảm nguy cơ overfitting.
- Lớp Flatten: Chuyển đổi đầu ra của lớp LSTM thành dạng một chiều.
- Lớp Dense: Tạo ra đầu ra cho mô hình với hàm kích hoạt softmax.

Sau khi xây dựng mô hình, ta tiến hành chạy mô hình với các thiết lập sau:

- Hàm mất mát: `sparse_categorical_crossentropy`
- Thuật toán tối ưu: Adam
- Chỉ số đánh giá: Accuracy

```

84/84 ————— 2s 29ms/step - accuracy: 0.4387 - loss: 2.6868
Epoch 4/200
84/84 ————— 2s 28ms/step - accuracy: 0.5400 - loss: 2.1896
Epoch 5/200
84/84 ————— 2s 28ms/step - accuracy: 0.5931 - loss: 1.8853
Epoch 6/200
84/84 ————— 2s 26ms/step - accuracy: 0.6509 - loss: 1.5552
Epoch 7/200
84/84 ————— 3s 26ms/step - accuracy: 0.6863 - loss: 1.3257
Epoch 8/200
84/84 ————— 2s 26ms/step - accuracy: 0.7482 - loss: 0.9979
Epoch 9/200
84/84 ————— 3s 27ms/step - accuracy: 0.7749 - loss: 0.8650
Epoch 10/200
84/84 ————— 2s 29ms/step - accuracy: 0.8243 - loss: 0.6831
Epoch 11/200
84/84 ————— 3s 27ms/step - accuracy: 0.8468 - loss: 0.6436
Epoch 12/200
84/84 ————— 2s 29ms/step - accuracy: 0.8788 - loss: 0.5191
Epoch 13/200
...
Epoch 199/200
84/84 ————— 6s 39ms/step - accuracy: 0.9988 - loss: 0.0017
Epoch 200/200
84/84 ————— 3s 39ms/step - accuracy: 0.9988 - loss: 0.0040

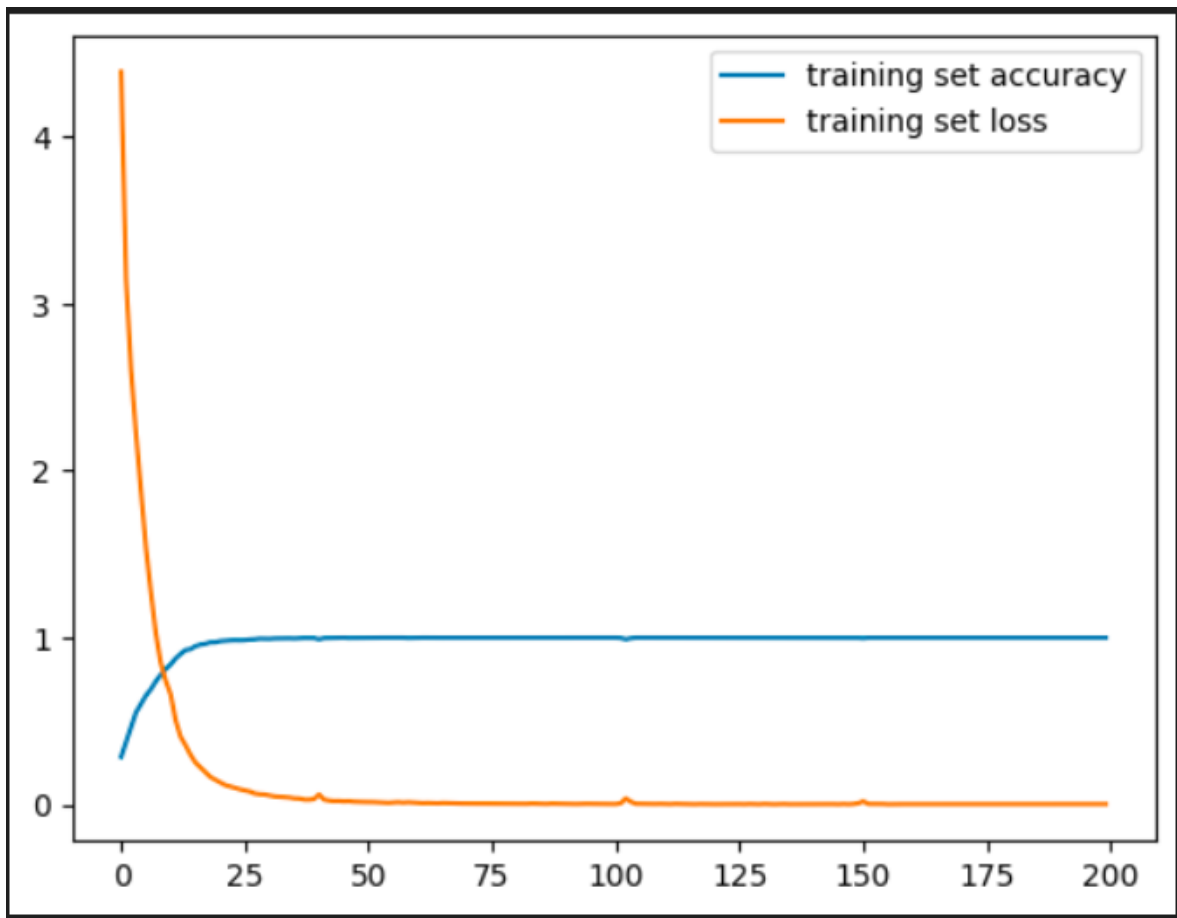
```

Hình 4. 3: Chạy mô hình

Mô hình được huấn luyện đạt kết quả ấn tượng với độ chính xác 99,88% và độ mất mát chỉ 0,0040. Với hiệu quả cao như vậy, mô hình có thể được ứng dụng vào chatbot để tự động trả lời các câu hỏi của người dùng về các thông tin tuyển sinh của trường học. Chatbot sử dụng mô hình để dự đoán câu trả lời phù hợp cho từng câu hỏi, giúp giải đáp các vấn đề một cách hiệu quả.

4.3 Đánh giá

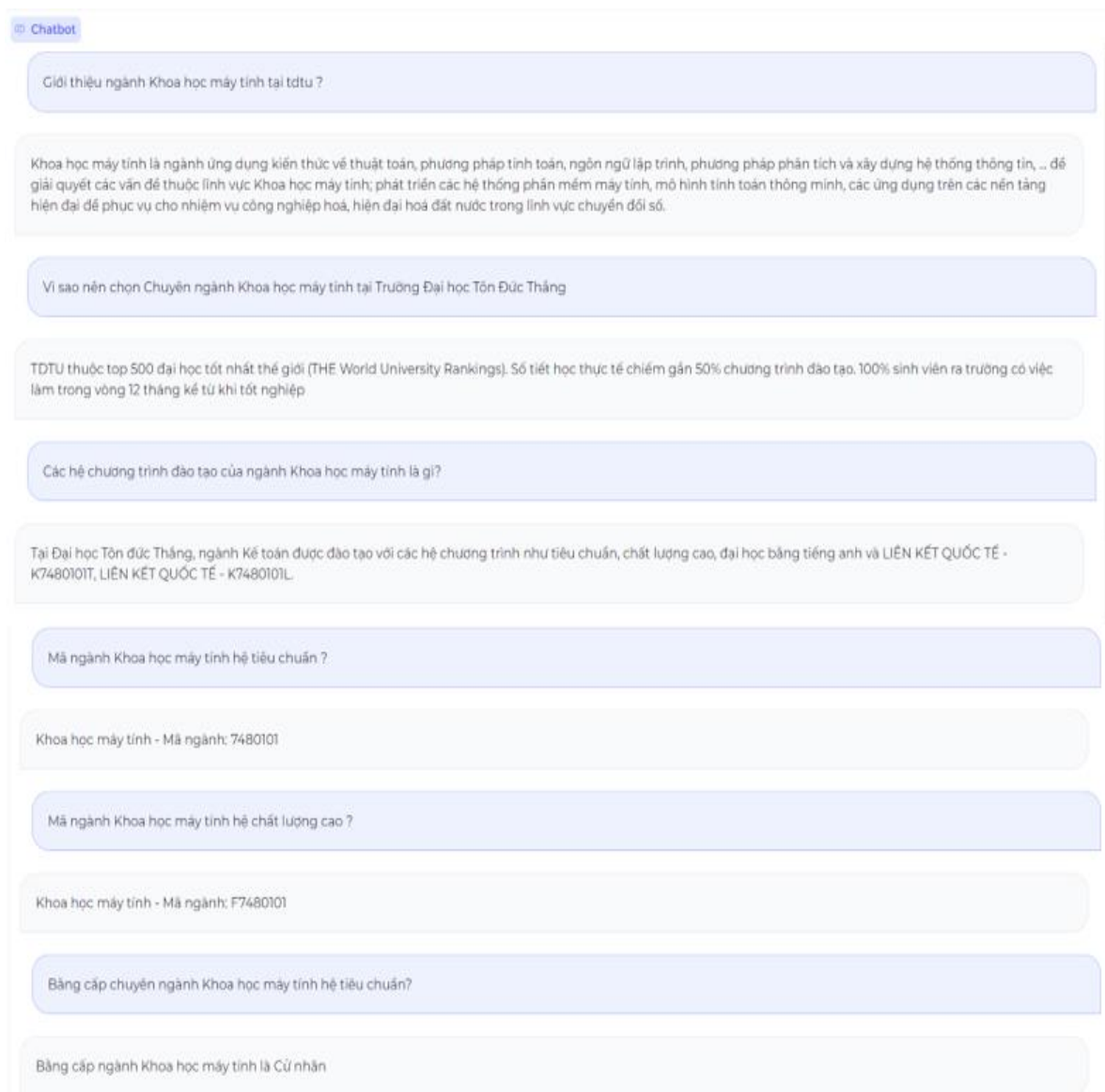
Để đánh giá tổng quan về mô hình đã được huấn luyện, chúng ta sẽ bắt đầu bằng việc xem xét kết quả đầu ra. Đánh giá sẽ dựa trên độ chính xác và biểu đồ trong hình 4.4 sẽ được sử dụng để hình dung kết quả huấn luyện của mô hình một cách trực quan.



Hình 4. 4: Biểu đồ thể hiện kết quả thực nghiệm mô hình

Biểu đồ trên cho thấy độ mất mát giảm dần và gần như là 0.0040 còn độ chính xác của mô hình thì ngược lại tăng dần và xấp xỉ 99.88%. Để xem kết quả này có chính xác hay không thì ứng dụng vào chatbot để xem kết quả mà hệ thống sẽ đưa ra như thế nào.

- Hỏi thông tin ngành khoa học máy tính



Hình 4. 5: Chatbot trả lời về thông tin ngành khoa học máy tính

- Hỏi về thông tin các chương trình đào tạo

Chatbot

Thông tin chương trình đào tạo ?

Tiêu chuẩn: Chương trình giảng dạy bằng tiếng Việt được xây dựng theo tiêu chuẩn kiểm định quốc tế như AUN-QA, FIBBA, ... Chất lượng cao: Chương trình giảng dạy bằng tiếng Việt và tiếng Anh, trong đó có hơn 50% môn học chuyên ngành bằng tiếng Anh. Đại học bằng tiếng Anh: Chương trình đào tạo chuẩn quốc tế, giảng dạy hoàn toàn bằng tiếng Anh (trừ một số môn lý luận chính trị, giáo dục thể chất, giáo dục quốc phòng). Liên kết quốc tế: Chương trình đào tạo tiên tiến từ các cơ sở giáo dục đại học uy tín trên thế giới.

Chương trình tiêu chuẩn là gì

Chương trình giáo dục đại học tiêu chuẩn của TDTU gồm có: (1) Chương trình đào tạo các cử nhân trình độ đại học chính quy, đạt chuẩn đầu ra bậc 6/8 theo khung trình độ quốc gia Việt Nam; (2) Chương trình đào tạo các ngành chuyên sâu đặc thù cấp bằng kỹ sư/kiến trúc sư/được sĩ, đạt chuẩn đầu ra bậc 7/8 khung trình độ quốc gia Việt Nam; Các chương trình đào tạo của TDTU đạt chuẩn đầu ra theo công bố của Trường và đáp ứng yêu cầu hội nhập trong khu vực và thế giới. Truy cập: <https://admission.tdtu.edu.vn/dai-hoc/chuong-trinh-tieu-chuan-day-hoc-bang-tieng-viet> để biết thêm chi tiết.

Chương trình chất lượng cao là gì ?

Là chương trình chất lượng cao giảng dạy bằng tiếng Việt, trong đó có một số học phần chuyên ngành dạy bằng tiếng Anh. Còn được gọi tắt là Chương trình giáo dục chất lượng cao. Chương trình được xây dựng với phương pháp giáo dục đặc biệt, đào tạo người học đạt chuẩn đầu ra cao hơn chương trình tiêu chuẩn về kỹ năng tiếng Anh, kỹ năng mềm, kỹ năng chuyên môn, các chứng chỉ nghề quốc gia /quốc tế. Khi sinh viên theo học chương trình chất lượng cao, bên cạnh những tiêu chuẩn mà sinh viên trường Đại học Tôn Đức Thắng nhận được, một số đặc trưng của sinh viên chương trình chất lượng cao được Nhà trường tổ chức và đào tạo. Truy cập <https://admission.tdtu.edu.vn/dai-hoc/chuong-trinh-chat-luong-cao> để biết thêm chi tiết.

Chuẩn tiếng anh đầu vào ?

Sinh viên trúng tuyển nhập học vào TDTU sẽ nộp bằng điểm/chứng chỉ tiếng Anh quốc tế hoặc thi kiểm tra đánh giá năng lực tiếng Anh đầu vào để xếp lớp học theo năng lực.

Chuẩn tiếng anh đầu vào của Đại học bằng tiếng Anh ?

Thí sinh đăng ký xét tuyển phải nộp chứng chỉ tiếng Anh quốc tế tương đương IELTS 5.0 trở lên hoặc tham gia kỳ thi đánh giá năng lực tiếng Anh do TDTU tổ chức để xét điều kiện xét tuyển. Nếu chưa đạt điều kiện tiếng Anh đầu vào, SV có thể tham gia chương trình dự bị tiếng Anh trước khi vào học chính thức.

Hình 4. 6: Chatbot trả lời về các thông tin của các chương trình đào tạo

- Hỏi về thông tin các phòng ban

Chatbot

Thông tin ký túc xá

Bạn có thể vào trang thông tin chi tiết thông tin kèm hình ảnh về ký túc xá các cơ sở tại Đại học Tôn Đức Thắng: <https://admission.tdtu.edu.vn/hoc-tai-tdtu/ky-tuc-xa>

Phòng ban hỗ trợ sinh viên ?

Phòng Công tác HSSV (A0003), Phòng Tài chính (A0007), Phòng Khảo thí và Kiểm định chất lượng (A0302), Phòng Đại học (A0005), Văn phòng Tư vấn và hỗ trợ (E0001), Đoàn Thanh niên và Hội sinh viên(A0004), TDT Creative Language Center (E0012), Trung tâm Công nghệ thông tin ứng dụng (A0506), Trung tâm Hợp tác doanh nghiệp & Cựu sinh viên (C003).

Phòng công tác HSSV hỗ trợ việc gì cho sinh viên ?

Cấp thẻ SV, các loại giấy chứng nhận, bằng điểm rèn luyện... Tiếp nhận và xử lý: gia hạn học phí, SV khó khăn, miễn giảm học phí. Xét các loại học bổng, miễn giảm học phí...

Thông tin của phòng tài chính là gì ?

Phòng Tài chính (A0007) Tel: (028) 37755070 Email: phongktc@tdtu.edu.vn

Thông tin của phòng khảo thí và kiểm định chất lượng là gì ?

Phòng Khảo thí và Kiểm định chất lượng (A0302) Tel: (028) 37755042 Email: qtpa@tdtu.edu.vn

Phòng Đoàn thanh niên và Hội sinh viên hỗ trợ gì ?

Nộp, rút sổ Đoàn, tiếp nhận và xử lý các vấn đề công tác Đoàn, Hội, Đảng, Quản lý các hoạt động của Đoàn thanh niên, Hội Sinh Viên.

Cần tư vấn về tiếng anh thì liên hệ đâu ?

TDT Creative Language Center (E0012) Tel: (028) 37755064 Email: clc@tdtu.edu.vn Website: <https://clc.tdtu.edu.vn>

Hình 4. 7: Chatbot trả lời về các thông tin của phòng ban

4.4 Nhận xét thực nghiệm

Dựa trên kết quả chạy mô hình và thực nghiệm đã được thực hiện, có thể thấy rằng mô hình xây dựng đạt được độ chính xác cao và có khả năng cung cấp các tư vấn chính xác dựa trên câu hỏi của người dùng. Mô hình đã học được nhiều trường hợp hơn từ bộ dữ liệu huấn luyện. Tuy nhiên, do bộ dữ liệu còn khá nhỏ, cần bổ sung

nhiều hơn và nghiên cứu thêm để cung cấp cho mô hình khả năng đưa ra câu trả lời chính xác hơn, đáp ứng tốt hơn nhu cầu của người dùng.

4.5 Xây dựng API

Flask là một microframework web Python được sử dụng để xây dựng các ứng dụng web và API một cách đơn giản và hiệu quả. Flask là một lựa chọn phổ biến cho việc xây dựng API vì nó nhẹ, dễ sử dụng và có thể mở rộng. Dưới đây là một số tính năng chính của Flask:

- Khởi tạo nhanh chóng: Flask có thể được khởi tạo và chạy chỉ với một vài dòng mã.
- Dễ sử dụng: Flask cung cấp API đơn giản và dễ hiểu, giúp bạn dễ dàng xây dựng các ứng dụng web và API.
- Có thể mở rộng: Flask có thể được mở rộng với nhiều thư viện và tiện ích mở rộng để hỗ trợ các chức năng khác nhau.
- Nhẹ: Flask là một microframework, có nghĩa là nó có kích thước nhỏ và sử dụng ít tài nguyên hệ thống.

Ứng dụng thư viện Flask trong python, ta có thể dễ dàng xây dựng một Web API. Trong dự án lần này, nhóm chỉ tập trung viết hàm POST để khi tại giao diện chatbot, người dùng gửi yêu cầu (cụ thể là câu hỏi), hệ thống sẽ gửi request đến server thông qua phương thức Post. Lúc này, hệ thống sẽ xử lý cũng như trả về câu trả lời tương ứng với câu hỏi thông qua mô hình LSTM đã được đào tạo sẵn.

```

@app.route("/", methods=["POST"])
def _answer_question():
    data = {"success": False}
    if request.form.get("question"):
        # Lấy câu hỏi từ người dùng gửi lên
        ques = request.form.get("question")
        prediction_input = utils._check_input(ques)
        output = model.predict(prediction_input)
        output = output.argmax()
        data['answer'] = utils._answer(output)
        data['success'] = True
        utils._record(ques, data['answer'])
    return json.dumps(data, ensure_ascii=False)

if __name__ == "__main__":
    print("App run!")
    # Load model
    model = utils._load_model()
    app.run(debug=False, host="127.0.0.1", threaded=False)

```

Đối với xây dựng API bao gồm 3 file chính: serve.py, utils.py, hyper.py. Với serve.py sẽ xử lý các yêu cầu gửi đến và trả về dữ liệu cần thiết. Còn file utils.py sẽ chứa các hàm xử lý như _load_model(), _check_input(), _answer(), _record(). Cuối cùng là file hyper.py là nơi lưu trữ các tham số quan trọng để vận hành các tác vụ trên.

4.6 Xây dựng giao diện web chatbot

Gradio là một thư viện Python mã nguồn mở giúp bạn dễ dàng xây dựng các giao diện web tương tác cho các mô hình học máy. Thư viện này cung cấp nhiều thành phần giao diện người dùng (UI) sẵn có, giúp bạn có thể tạo chatbot một cách nhanh chóng và dễ dàng. Áp dụng thư viện Gradio trong py, ta có thể dễ dàng xây dựng một giao diện dành cho chatbot rất thân thiện với người dùng.

Bằng cách khởi tạo một hàm để xử lý dữ liệu câu hỏi nhập từ người dùng, sau đó gửi một yêu cầu đến Web Api đã tạo ở trên, lúc này chỉ chờ nhận kết quả trả về và hiển thị lên giao diện cho người dùng quan sát được câu trả lời.

```
def qa(message, history):
    data = {
        "question": message,
    }
    response = requests.post(url, data=data)
    answer = response.json()['answer']
    return answer

chat = gr.ChatInterface(
    qa,
    chatbot=gr.Chatbot(height=400),
    textbox=gr.Textbox(placeholder="Thắc mắc của bạn là gì ?", container=False, scale=7),
    title="SchoolAssistant",
    description="Hệ thống tư vấn tự động trong trường học",
    theme="soft",
    examples=["Hello", "Tư vấn tuyển sinh", "Khám phá TDTU", "Có học bổng 100% học phí không?"],
    cache_examples=True,
    retry_btn=None,
    undo_btn=None,
    submit_btn="Gửi",
    clear_btn="Clear",
)
chat.launch()
```

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

Mô hình trả lời tự động đã được xây dựng bằng việc sử dụng bộ dữ liệu và các kỹ thuật hỗ trợ từ Tensorflow. Mô hình này đạt được độ chính xác gần 99,88% và chỉ mất mát 0,0040, cho thấy tiềm năng ứng dụng thực tế trong việc tự động hóa việc cung cấp tư vấn cho người dùng. Tuy nhiên, bộ dữ liệu hiện tại vẫn còn hạn chế và cần được mở rộng trong tương lai. Lĩnh vực xây dựng mô hình học máy để tạo ra câu trả lời cho hệ thống hỏi đáp đã thu hút sự quan tâm của nhiều nhà nghiên cứu trong thời gian gần đây. Mục tiêu chung là phát triển các mô hình hiệu quả và chính xác. Nổi bật trong xu hướng này là ứng dụng trí tuệ nhân tạo (AI) để xây dựng chatbot trả lời tự động cho các lĩnh vực như trường học. Kết quả thu được từ quá trình học tập và nghiên cứu là một mô hình tiềm năng với độ chính xác cao, mở ra hướng đi mới cho việc tự động hóa tư vấn trong nhiều lĩnh vực.

5.2 Hạn chế và hướng phát triển

5.2.1 Hạn chế

Trong quá trình làm dự án công nghệ thông tin thì vẫn còn các hạn chế sau:

- Khả năng xử lý ngôn ngữ tiếng Việt còn hạn chế.
- Bộ dữ liệu hiện tại tương đối nhỏ, cần được bổ sung thêm.

5.2.2 Hướng phát triển

Hướng phát triển tiếp theo của nghiên cứu sẽ là:

- Tìm hiểu và áp dụng các phương pháp xử lý dữ liệu tối ưu, nhằm cải thiện hiệu suất mô hình, giảm độ mất mát và nâng cao độ chính xác.
- Mở rộng và nâng cao chất lượng bộ dữ liệu, hướng đến ứng dụng mô hình vào thực tế.

TÀI LIỆU THAM KHẢO

- Dancker, J. (2022, December 26). A Brief Introduction to Recurrent Neural Networks. *Towards Data Science*. <https://towardsdatascience.com/a-brief-introduction-to-recurrent-neural-networks-638f64a61ff4>
- Do, P., & Phan, T. (2021). BERT+vnKG: Using Deep Learning and Knowledge Graph to Improve Vietnamese Question Answering System. *International Journal of Advanced Computer Science and Applications*, 11. <https://doi.org/10.14569/IJACSA.2020.0110761>
- Dušek, O., & Jurcicek, F. (2016). Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 45–51. <https://doi.org/10.18653/v1/P16-2008>
- Embedding—Machine Learning cho dữ liệu dạng bảng*. (n.d.). Retrieved March 13, 2024, from https://machinelearningcoban.com/tabml_book/ch_embedding/embedding.html
- Fouzan, M. (2023, October 22). Understanding LSTM, GRU, and RNN Architectures. *Medium*. <https://medium.com/@mfouzan144/understanding-lstm-gru-and-rnn-architectures-e0b3a0c1d741>

- Fulmal, V., & Al, E. (2021). The Implementation of Question Answer System Using Deep Learning. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(1S), Article 1S.
- Le, N.-T., Le, T.-B., Truong, S.-D., & Le, H.-B. (2020). Building Filters for Vietnamese Chatbot Responses. *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 1–6.
<https://doi.org/10.1109/RIVF48685.2020.9140770>
- Mervin, R. (2013). *An Overview of Question Answering System. 1.*
- Minh T. H. C. (n.d.). *ỨNG DỤNG MÁY HỌC TRONG TẠO SINH CÂU TRẢ LỜI CHO HỆ THỐNG HỎI - ĐÁP.*
- Nguyen, D., Nguyen, D. Q., & Pham, S. (2009). A Vietnamese Question Answering System. 26–32. <https://doi.org/10.1109/KSE.2009.42>
- Nguyen, T., & Shcherbakov, M. (2018). A Neural Network based Vietnamese Chatbot. *2018 International Conference on System Modeling & Advancement in Research Trends (SMART)*, 147–149.
<https://doi.org/10.1109/SYSMART.2018.8746962>
- Nguyen, T. T., Le, A. D., Hoang, H. T., & Nguyen, T. (2021). NEU-chatbot: Chatbot for admission of National Economics University. *Computers and Education: Artificial Intelligence*, 2, 100036.
<https://doi.org/10.1016/j.caeai.2021.100036>

NLP [P3]—Seq2Seq Model—Hạt nhân của Google Translate. (2019, May 18).

Nero's Blog. <https://nerophung.github.io/2019/05/18/nlp-p3>

Palangi, H., Smolensky, P., He, X., & Deng, li. (2017). Deep Learning of

Grammatically-Interpretable Representations Through Question-Answering.

Proceedings of the AAAI Conference on Artificial Intelligence, 32.

<https://doi.org/10.1609/aaai.v32i1.12004>

Ramaiyan, P., & Krishnamurthy, L. (2020). Evaluating the Performance of

Recurrent Neural Network based Question Answering System with Easy and

Complex bAbI QA Tasks. *International Journal of Advanced Science and*

Technology, 29, 1389–1402.

Shyam, T., E, S., B, V., M, V., & M, P. (2020). *Code Mixed Question Answering*

Challenge using Deep Learning Methods. 1331–1337.

<https://doi.org/10.1109/ICCES48766.2020.9137971>

Sklearn.preprocessing.LabelEncoder. (n.d.). Scikit-Learn. Retrieved March 14,

2024, from [https://scikit-](https://scikit-learn/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html)

[learn/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html](https://scikit-learn/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html)

Team, K. (n.d.). *Keras documentation: Embedding layer*. Retrieved March 13,

2024, from https://keras.io/api/layers/core_layers/embedding/

Tf.keras.preprocessing.text.Tokenizer / TensorFlow v2.15.0.post1. (n.d.).

TensorFlow. Retrieved March 14, 2024, from

https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer

Tf.keras.utils.pad_sequences / *TensorFlow v2.15.0.post1*. (n.d.). TensorFlow.

Retrieved March 14, 2024, from

https://www.tensorflow.org/api_docs/python/tf/keras/utils/pad_sequences

Tran, O. T., & Luong, T. C. (2020). Understanding what the users say in chatbots:

A case study for the Vietnamese language. *Engineering Applications of Artificial Intelligence*, 87, 103322.

<https://doi.org/10.1016/j.engappai.2019.103322>

trituenhantao.io. (2019, December 4). Seq2Seq—Hiểu về mô hình Encoder-

Decoder. *Trí tuệ nhân tạo*. <https://trituenhantao.io/kien-thuc/hieu-ve-mo-hinh-encoder-decoder-seq2seq/>

Udani, M., Shrivastava, A., Shukla, V., & Rao, A. (2013). Question Answering System Based on Artificial Intelligence for Restricted Domain. *International Journal of Engineering Research & Technology*, 2(12).

<https://doi.org/10.17577/IJERTV2IS120050>

Xie, Z., Zeng, Z., Zhou, G., & He, T. (2016). Knowledge Base Question Answering Based on Deep Learning Models. In C.-Y. Lin, N. Xue, D. Zhao, X. Huang, & Y. Feng (Eds.), *Natural Language Understanding and Intelligent Applications* (pp. 300–311). Springer International Publishing.

https://doi.org/10.1007/978-3-319-50496-4_25

Yu, W., Wu, L., Deng, Y., Mahindru, R., Zeng, Q., Guven, S., & Jiang, M. (2020).

A Technical Question Answering System with Transfer Learning. In Q. Liu & D. Schlangen (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 92–99). Association for Computational Linguistics.

<https://doi.org/10.18653/v1/2020.emnlp-demos.13>

Zhong, B., He, W., Huang, Z., Love, P. E. D., Tang, J., & Luo, H. (2020). A

building regulation question answering system: A deep learning methodology. *Advanced Engineering Informatics*, 46, 101195.

<https://doi.org/10.1016/j.aei.2020.101195>