TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TRẦN VĂN THẢO - 52000805**

**NGUYỄN VÕ CÔNG HUY - 52000765**

**XÂY DỰNG HỆ THỐNG TƯ VẤN TUYỂN SINH CHO SINH VIÊN**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN**

**TÍNH TOÁN THÔNG MINH**

**VÀ HỆ THỐNG THÔNG TIN**

**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**



**TRẦN VĂN THẢO - 52000805**

**NGUYỄN VÕ CÔNG HUY - 52000765**

**XÂY DỰNG HỆ THỐNG TƯ VẤN TUYỂN SINH CHO SINH VIÊN**

**DỰ ÁN CÔNG NGHỆ THÔNG TIN**

**TÍNH TOÁN THÔNG MINH**

**VÀ HỆ THỐNG THÔNG TIN**

Người hướng dẫn

**TS. TRẦN LƯƠNG QUỐC ĐẠI**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**LỜI CẢM ƠN**

Lời nói đầu tiên, chúng em xin gửi lời cảm ơn chân thành nhất, đặc biệt nhất đến ***TS Trần Lương Quốc Đại*** đã hỗ trợ và truyền đạt những kiến thức quý báo cũng như đã tạo mọi điều kiện thuận lợi nhất để chúng em được tiếp cận và hoàn thành môn Dự án công nghệ thông tin.

Với thời gian và trình độ còn hạn chế, bài báo cáo không thể tránh khỏi những thiếu sót. Kính mong thầy sẽ chỉ bảo cũng như là đóng góp ý kiến để bài báo cáo của chúng em ngày càng dần hoàn thiện hơn. Đó sẽ là hành trang quý giá để chúng em có thể hoàn thiện về kĩ năng và kiến thức liên quan đến những vấn đề sau này.

Lời cuối cùng, chúng em xin kính chúc thầy thật nhiều sức khỏe, thành công, hạnh phúc và luôn giữ mãi sự nhiệt huyết để có thể giúp thêm thật nhiều thế hệ sinh viên Trường Đại Học Tôn Đức Thắng có nhiều sự tự tin và vững kiến thức với môn Dự án công nghệ thông tin. Ngọn lửa của môn học sẽ ngày càng được lan tỏa rộng rãi nhiều hơn!

*TP. Hồ Chí Minh, ngày 03 tháng 03 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*N.V.C Huy*

*Nguyễn Võ Công Huy*

*T.V Thảo*

*Trần Văn Thảo*

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng 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 03 tháng 03 năm 2024*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*N.V.C Huy*

*Nguyễn Võ Công Huy*

*T.V Thảo*

*Trần Văn Thảo*

**XÂY DỰNG HỆ THỐNG TƯ VẤN TUYỂN SINH CHO SINH VIÊN**

**TÓM TẮT**

Hệ thống tư vấn tuyển sinh sinh viên là một đề tài nghiên cứu nhằm tạo ra một hệ thống hỗ trợ tư vấn, giải đáp các thắc mắc cho các thí sinh khi xét tuyển vào các trường đại học ở Việt Nam. Đề tài này đã xây dựng một website hoàn chỉnh dựa trên MERN Stack, cung cấp thông tin tuyển sinh, các khóa học cho các thí sinh có mong muốn học trước ngành Công nghệ thông tin và hỗ trợ tư vấn cho các thí sinh trong quá trình xét tuyển. Hệ thống này giúp tối ưu hóa việc tư vấn, nâng cao chất lượng đào tạo, và hạn chế tình trạng bỏ học giúp các thí sinh có thể đưa ra quyết định chính xác nhất cho tương lai của mình.

Ngoài ra, xu hướng sử dụng chatbot đang được ưa chuộng trên thế giới. Việc áp dụng chatbot vào tư vấn tuyển sinh giúp tối ưu hiệu quả vận hành, tóm tắt và cung cấp thông tin nhanh chóng, đầy đủ nhất cho thí sinh.

**STUDENT ADMISSION ADVISORY SYSTEM**

**ABSTRACT**

Student Admission Advisory System is a research topic that aims to create a system of advisory support, answering questions for applicants when applying to universities in Vietnam. This topic has built a complete website based on MERN Stack, providing admission information, courses for candidates who wish to pre-study in Information Technology and advising candidates during the admissions process. This system optimizes counselling, improves the quality of training, and limits dropouts, enabling candidates to make the most accurate decisions for their future.

Besides, the trend of using chatbots is popular around the world. The use of chatbots in the admissions consultancy helps to optimize operational efficiency, summarize and provide the candidates with the fastest, most complete information.

**MỤC LỤC**

[DANH MỤC HÌNH VẼ vii](#_Toc161069506)

[DANH MỤC BẢNG BIỂU viii](#_Toc161069507)

[DANH MỤC CÁC CHỮ VIẾT TẮT ix](#_Toc161069508)

[CHƯƠNG 1. MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI 1](#_Toc161069509)

[1.1 Lý do chọn đề tài 1](#_Toc161069510)

[1.2 Mục tiêu thực hiện đề tài 1](#_Toc161069511)

[1.3 Các tác nhân của hệ thống 1](#_Toc161069512)

[1.4 Các usecase 1](#_Toc161069513)

[1.5 Lược đồ quan hệ thực thể - ERD 1](#_Toc161069514)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 2](#_Toc161069515)

[2.1 Hệ thống hỏi đáp mở (ODQA) 2](#_Toc161069516)

[2.1.1 Hệ thống hỏi đáp mở là gì? 2](#_Toc161069517)

[2.1.2 Các loại hệ thống hỏi đáp mở trong ODQA 2](#_Toc161069518)

[2.1.3 Các mô hình hỏi đáp mở 3](#_Toc161069519)

[2.1.4 … 8](#_Toc161069520)

[2.2 Module Website 10](#_Toc161069521)

[2.2.1 MongoDB và Mongoose 10](#_Toc161069522)

[2.2.2 ExpressJS 11](#_Toc161069523)

[2.2.3 ReactJS 12](#_Toc161069524)

[2.2.4 NodeJS 13](#_Toc161069525)

[CHƯƠNG 3. DỮ LIỆU 14](#_Toc161069526)

[3.1 …Mai làm 14](#_Toc161069527)

[CHƯƠNG 4. MÔ HÌNH ĐỀ XUẤT 14](#_Toc161069528)

[4.1 …Mô hình + thực nghiệm của mỗi mô hình 14](#_Toc161069529)

[CHƯƠNG 5. KẾT QUẢ ĐẠT ĐƯỢC 14](#_Toc161069530)

[5.1 …Mấy cái UI 14](#_Toc161069531)

[CHƯƠNG 6. KẾT LUẬN 14](#_Toc161069532)

[6.1 Kết luận 14](#_Toc161069533)

[6.2 Hướng phát triển 14](#_Toc161069534)

[TÀI LIỆU THAM KHẢO 15](#_Toc161069535)

# DANH MỤC HÌNH VẼ

[Hình 2.1: Tổng quan về ba loại mô hình hỏi đáp mở 3](#_Toc161060446)

[Hình 2.2: Quy trình tìm kiếm câu trả lời cho câu hỏi của Retriever-Reader 4](#_Toc161060447)

[Hình 2.3: Sơ đồ quy trình hệ thống thông tin hoạt động 5](#_Toc161060448)

[Hình 2.4: Quy trình tìm kiếm câu trả lời cho câu hỏi của Retriever-Generator 7](#_Toc161060449)

[Hình 2.5: Cấu trúc project sử dụng Express 11](#_Toc161060450)

# DANH MỤC BẢNG BIỂU

[Bảng 4.1: Thống kê kiểu thực thể trong tập VLSP 2016 6](#_Toc160218996)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| BERT | Bidirectional Encoder Representations from Transformers |
| GEC | Grammatical Error Correction |
| MLM | Masked Language Model |
| NLP | Natural Language Processing |
| NSP | Next Sentence Prediction |

# MỞ ĐẦU VÀ TỔNG QUAN ĐỀ TÀI

## Lý do chọn đề tài

Một trong những lý do quan trọng để chọn đề tài này là đáp ứng nhu cầu ngày càng tăng về thông tin trong quá trình tuyển sinh. Hệ thống tuyển sinh truyền thống thường gặp khó khăn trong việc cung cấp thông tin nhanh chóng và chính xác đối với các học sinh đang tìm hiểu về quy trình tuyển sinh. Sự chậm trễ trong phản hồi có thể tạo ra rủi ro về việc học sinh không nhận được thông tin đầy đủ và kịp thời, ảnh hưởng đến quyết định của họ trong việc chọn lựa trường học.

Hệ thống Q&A có khả năng ngay lập tức cung cấp thông tin chi tiết và rõ ràng, giúp học sinh giải đáp mọi thắc mắc một cách hiệu quả. Bằng cách này, đề tài không chỉ giải quyết vấn đề bất cập trong quá trình tuyển sinh mà còn đáp ứng nhu cầu ngày càng cao của học sinh đối với thông tin nhanh chóng và dễ tiếp cận.

## Mục tiêu thực hiện đề tài

Mục tiêu chính của nghiên cứu là phát triển một hệ thống Q&A trong quá trình tuyển sinh, nhằm giảm thời gian phản hồi, đa dạng hóa khả năng trả lời, tự động cập nhật thông tin, và thực nghiệm ứng dụng trong môi trường thực tế. Mục tiêu này nhằm tối ưu hóa trải nghiệm thông tin cho học sinh, giảm công sức của tổ chức tuyển sinh, và tạo cơ sở cho phát triển tương lai trong ứng dụng trí tuệ nhân tạo trong giáo dục.

## Các tác nhân của hệ thống

…

## Các usecase

…

## Lược đồ quan hệ thực thể - ERD

…

# CƠ SỞ LÝ THUYẾT

## Hệ thống hỏi đáp mở (ODQA)

### Hệ thống hỏi đáp mở là gì?

Hệ thống hỏi đáp mở (Open Domain Question Answering) là một loại nhiệm vụ ngôn ngữ tự nhiên, yêu cầu mô hình tạo ra câu trả lời cho các câu hỏi đối với dữ liệu thực tế bằng ngôn ngữ tự nhiên. Câu trả lời đúng là khách quan nên việc đánh giá hiệu quả của mô hình rất đơn giản.

Ví dụ:

Câu hỏi: Khi nào bắt đầu thời gian đăng ký xét tuyển đại học năm nay?

Câu trả lời: Thời gian đăng ký xét tuyển đại học năm nay bắt đầu từ ngày 15 tháng 6 đến hết ngày 30 tháng 6.

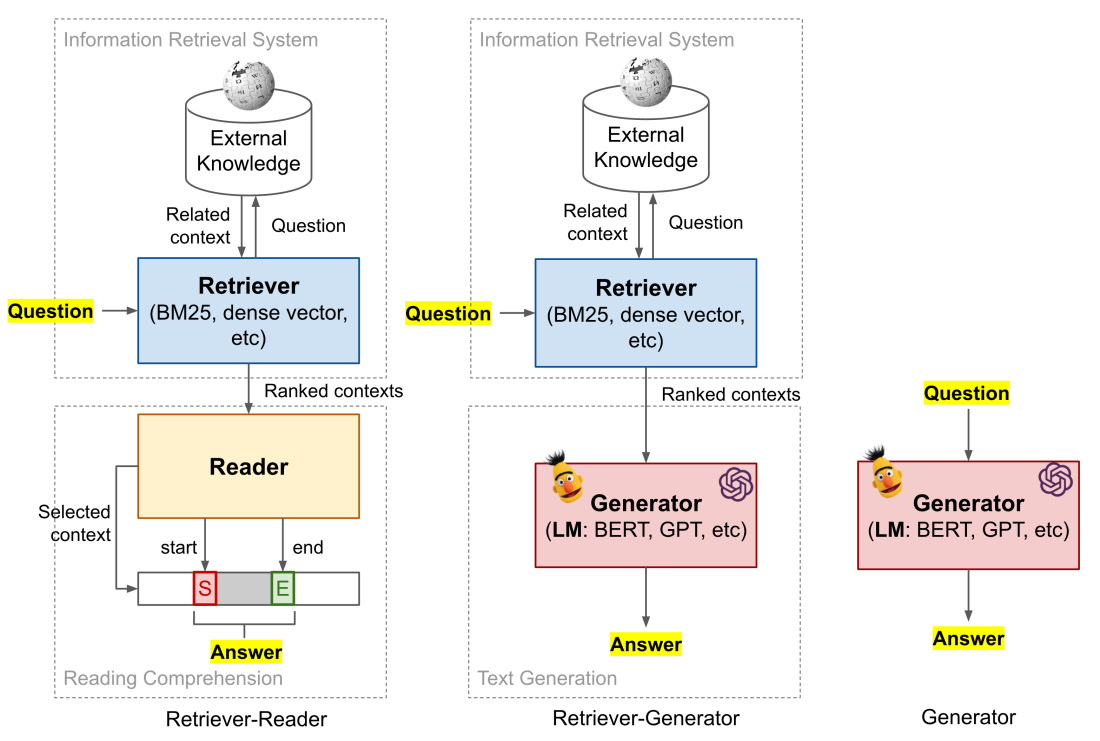
Phần "open-domain" đề cập đến sự thiếu hụt của ngữ cảnh liên quan đối với bất kỳ câu hỏi thực tế nào được đặt ra một cách tùy ý. Trong ví dụ trên, mô hình chỉ nhận câu hỏi làm đầu vào mà không có bất kỳ thông tin nào về "tại sao quyết định đăng ký xét tuyển lại có thời gian như vậy" được cung cấp. Trong trường hợp cả câu hỏi và ngữ cảnh đều được cung cấp, nhiệm vụ này được biết đến là Đọc hiểu (Reading Comprehension - RC).

Một mô hình ODQA có thể hoạt động có hoặc không có quyền truy cập vào nguồn tri thức bên ngoài (ví dụ: trang web của trường đại học), và điều này có thể được gọi là open-book hoặc closed-book.

### Các loại hệ thống hỏi đáp mở trong ODQA

Khi xem xét các loại hỏi đáp trong open-domain, thường được phân loại theo 3 loại có độ khó tăng dần:

* Mô hình có khả năng ghi nhớ đúng và trả lời với câu trả lời của một câu hỏi mà nó đã thấy trong quá trình huấn luyện.
* Mô hình có khả năng trả lời các câu hỏi mới trong quá trình kiểm thử và chọn câu trả lời từ tập các câu trả lời mà nó đã thấy trong quá trình huấn luyện.
* Mô hình có khả năng trả lời các câu hỏi mới mà có câu trả lời không xuất hiện trong tập dữ liệu huấn luyện.



Hình 2.1: Tổng quan về ba loại mô hình hỏi đáp mở

### Các mô hình hỏi đáp mở

#### Retriever – Reader

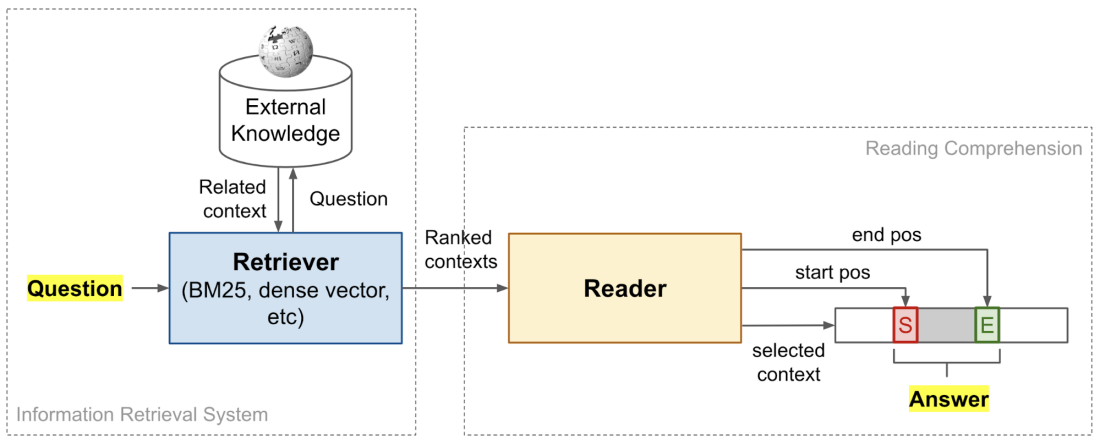
1. Tổng quan về quy trình thực hiện

Dựa trên một câu hỏi thông tin cụ thể, nếu một mô hình ngôn ngữ không có ngữ cảnh hoặc không đủ lớn để ghi nhớ ngữ cảnh có sẵn trong bộ dữ liệu huấn luyện, khả năng đoán đúng câu trả lời là khá thấp. Trong kỳ thi cho phép xem tài liệu, sinh viên được phép tham khảo các nguồn tài liệu bên ngoài như ghi chú và sách khi trả lời các câu hỏi kiểm tra. Tương tự, hệ thống Hỏi-Đáp mở (ODQA) có thể được kết hợp với một cơ sở kiến thức phong phú để xác định các tài liệu liên quan làm bằng chứng cho câu trả lời.

Chúng ta có thể phân tích quy trình này thành hai giai đoạn:

* Tìm kiếm ngữ cảnh liên quan trong một kho dữ liệu kiến thức bên ngoài (Retriever):
  + Ở giai đoạn này, hệ thống cần xác định các tài liệu, văn bản hoặc nguồn thông tin ngoại vi chứa thông tin liên quan đến câu hỏi.
  + Quá trình này có thể thực hiện thông qua một hệ thống hỏi đáp mở (ODQA) được kết hợp với một kho dữ liệu lớn chứa kiến thức đa dạng.
* Xử lý ngữ cảnh đã tìm được để trích xuất câu trả lời (Reader):
  + Sau khi có được ngữ cảnh liên quan, hệ thống cần phân tích và xử lý thông tin để trích xuất câu trả lời đáp ứng câu hỏi.
  + Các phương pháp xử lý ngôn ngữ tự nhiên và trích xuất thông tin có thể được áp dụng để hiểu và tìm ra câu trả lời trong ngữ cảnh tìm kiếm.

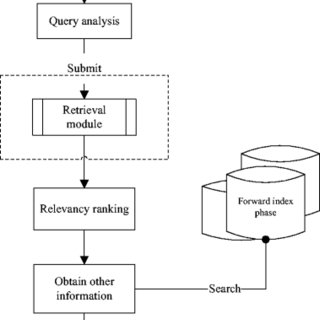
Để hiểu tổng quan về quy trình tìm kiếm và cho ra câu trả lời chúng ta có thể quan sát hình 2.2



Hình 2.2: Quy trình tìm kiếm câu trả lời cho câu hỏi của Retriever-Reader

1. Hệ thống thu thập thông tin (Retriever)

Hệ thống thu thập thông tin (Retriever) trong ngữ cảnh truy vấn thông tin (Information Retrieval) là một phần quan trọng của mô hình hệ thống truy vấn thông tin. Nhiệm vụ chính của Retriever là tìm kiếm và trả về các tài liệu phù hợp nhất với truy vấn từ người dùng.



Hình 2.3: Sơ đồ quy trình hệ thống thông tin hoạt động

Có hai phương pháp chính để triển khai hệ thống thu thập thông tin (retriever) trong hệ thống trả lời câu hỏi mở (ODQA):

* Hệ thống thu thập thông tin (IR) dựa trên các đặc trưng TF-IDF cổ điển không dựa trên học máy (Classic IR): Phương pháp này tập trung vào việc sử dụng hệ thống thu thập thông tin (IR) truyền thống, dựa trên đặc trưng TF-IDF để xác định sự tương đồng giữa câu hỏi và các văn bản nguồn. Các truy vấn được biểu diễn bằng các vector dựa trên tần suất xuất hiện của các từ khóa và các văn bản được sắp xếp theo độ tương đồng.
* Hệ thống thu thập thông tin (IR) dựa trên các vector dense của văn bản được tạo ra bởi các mạng nơ-ron (neural IR): Phương pháp này sử dụng các mô hình mạng nơ-ron để tạo ra các biểu diễn vector dense của văn bản, thường thông qua các mô hình như BERT, RoBERTa, hay các biến thể khác của Transformer. Các biểu diễn này mang lại khả năng hiểu ngữ nghĩa sâu rộng và tương đồng ngữ cảnh giữa câu hỏi và nguồn thông tin.

1. Quá trình đọc và trả lời (Reader)

Mô hình Reader là một phần quan trọng của hệ thống hỏi đáp mở, chịu trách nhiệm đọc và hiểu nội dung của văn bản để tìm kiếm thông tin cần thiết để trả lời các câu hỏi từ người dùng. Mục tiêu chính của mô hình là hiểu và rút trích thông tin ý nghĩa từ văn bản nguồn để tạo ra câu trả lời chính xác.

Kiến trúc chung của mô hình Reader gồm các phần sau:

* Lớp Embedding: Chuyển đổi từ ngôn ngữ tự nhiên thành các vector số thực để máy tính có thể hiểu được ý nghĩa của từng từ.
* Kiến Trúc Tự Chú Ý (Self-Attention): Các lớp tự chú ý giúp mô hình tập trung vào các phần quan trọng của văn bản, xử lý thông tin từ cấp từ đến cấp câu.
* Trích Xuất Thông Tin: Sau khi đọc và hiểu, mô hình trích xuất thông tin quan trọng để sử dụng trong quá trình tạo câu trả lời.

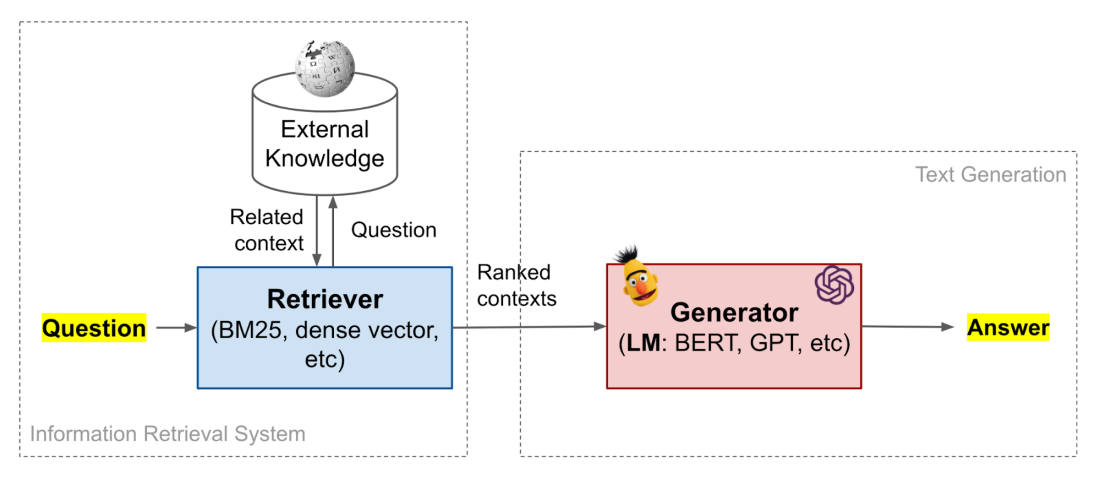
Các mô hình thường sử dụng trong vai trò Reader: Bi directional LSTM, BERT,...

#### Retriever – Generative

1. Tổng quan về quy trình thực hiện

So với phương pháp thu thập thông tin và đọc hiểu (retriever-reader), phương pháp thu thập thông tin và tạo văn bản (retriever-generator) cũng gồm 2 giai đoạn, nhưng giai đoạn thứ hai được thiết kế để tạo ra văn bản tự do trực tiếp để trả lời câu hỏi thay vì trích xuất vị trí bắt đầu/kết thúc trong một đoạn văn được thu thập. Một số nghiên cứu cũng gọi phương pháp này là "Generative Question Answering" (Trả lời câu hỏi bằng cách tạo văn bản).

Để hiểu tổng quan về quy trình tìm kiếm và cho ra câu trả lời của retriever - generator chúng ta có thể quan sát hình 2.4



Hình 2.4: Quy trình tìm kiếm câu trả lời cho câu hỏi của Retriever-Generator

1. Quá trình tạo câu trả lời (Generator)

Không giống như mô hình Reader, mô hình Generator trong hệ thống hỏi đáp mở là một thành phần quan trọng giúp tạo ra câu trả lời chính xác và có ý nghĩa cho các câu hỏi mà hệ thống không có sẵn câu trả lời trong nguồn dữ liệu cố định.

Kiến trúc chung của mô hình Generator gồm các phần sau:

* Mạng Nơ-ron Tự Sinh (Seq2Seq): Sử dụng mô hình Seq2Seq, trong đó có một mạng nơ-ron Encoder chịu trách nhiệm biểu diễn câu hỏi đầu vào thành một vector ngữ cảnh. Mạng nơ-ron Decoder sẽ sinh ra câu trả lời dựa trên vector ngữ cảnh đã biểu diễn.
* Attention Mechanism: Các mô hình thường sử dụng cơ chế chú ý để tập trung vào các phần quan trọng của câu hỏi và nguồn dữ liệu khi tạo ra câu trả lời. Cơ chế này giúp mô hình tập trung vào các từ quan trọng và mối quan hệ trong ngữ cảnh.
* Transfer Learning và Pre-training: Sử dụng kỹ thuật pre-training để học các biểu diễn ngôn ngữ tự nhiên từ dữ liệu lớn trước khi fine-tuning cho nhiệm vụ ODQA cụ thể. Mô hình có thể sử dụng các kiến thức từ các nguồn lớn như Wikipedia để cải thiện khả năng sinh câu trả lời.

Các mô hình thường sử dụng trong vai trò Generator: T5, GPT, BERT,...

#### Generative

… Nào bro làm thì fill vào

### …

#### Mô hình TF-IDF

1. Giới thiệu mô hình TF-IDF

Mô hình TF-IDF (Term Frequency-Inverse Document Frequency), là một trong những phương pháp truy vấn thông tin cổ điển nhất được sử dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên và truy vấn thông tin.

1. Phương pháp tìm kiếm thông tin
2. Dưới đây là mô tả chi tiết về cách mô hình TF-IDF hoạt động tìm kiếm thông tin:

Bước 1: Thu thập dữ liệu

* Dữ liệu đầu vào của mô hình là tập văn bản (ví dụ: tài liệu, bài viết, văn bản) được gọi là "tập tài liệu".
* Mỗi tài liệu trong tập được biểu diễn bởi một bộ từ vựng, tức là tập hợp tất cả các từ xuất hiện trong tài liệu đó.

Bước 2: Tạo chỉ số (Indexing)

* Mỗi từ trong tập hợp từ vựng được gán một số thứ tự (index).
* Dùng index này để tạo ra một ma trận, với mỗi dòng biểu diễn một tài liệu và mỗi cột biểu diễn một từ trong bộ từ vựng.
* Mỗi ô trong ma trận chứa giá trị TF-IDF của từ đó trong tài liệu tương ứng.

Bước 3: Tính giá trị Term Frequency (TF)

* Term Frequency đo tần suất xuất hiện của mỗi từ trong một tài liệu. Công thức tính TF cho một từ w trong tài liệu d là:

\[TF(w, d) = \frac{\text{số lần xuất hiện của w trong d}}{\text{Tổng số từ trong d}}\]

Bước 4: Tính giá trị Inverse Document Frequency (IDF)

* Inverse Document Frequency đo mức độ quan trọng của một từ trong toàn bộ tập tài liệu. Công thức tính IDF cho một từ w trong toàn bộ tập tài liệu là:

\[IDF(w) = \log\left(\frac{\text{Tổng số tài liệu}}{\text{Số tài liệu chứa từ w + 1}}\right)\] (+1 ở mẫu để tránh trường hợp chia cho 0 nếu từ đó không xuất hiện trong bất kỳ tài liệu nào)

Bước 5: Tính giá trị TF-IDF

* Giá trị TF-IDF của mỗi từ trong mỗi tài liệu được tính bằng cách nhân giá trị TF với giá trị IDF:

\[TF-IDF(w, d) = TF(w, d) \times IDF(w)\]

Bước 6: Truy vấn và tính điểm tương tự

* Khi có một truy vấn, cũng được biểu diễn dưới dạng vector TF-IDF.
* Tính điểm tương tự giữa truy vấn và mỗi tài liệu bằng cách sử dụng các phương pháp đo lường sự tương đồng như cosine similarity.
* Các tài liệu được sắp xếp theo điểm tương tự và trả về kết quả.

Mô hình TF-IDF là một mô hình đơn giản nhưng hiệu quả trong nhiều trường hợp truy vấn thông tin và tìm kiếm văn bản. Tuy nhiên, nó không quan tâm đến cấu trúc ngữ nghĩa của câu hoặc văn bản, điều này là một hạn chế khi so sánh về mặt ngữ nghĩa.

#### Mô hình Bi directional LSTM

1. Giới thiệu mô hình Bi directional LSTM

…

1. Quá trình đọc và trích xuất thông tin

…

#### Mô hình PhoBERT (Generative)

1. Giới thiệu mô hình PhoBERT

…

1. Tạo câu trả lời tự nhiên

…

## Module Website

### MongoDB và Mongoose

#### Tổng quan về MongoDB:

MongoDB là một database hướng tài liệu (document) hay còn được gọi một dạng NoSQL database. Vì thế, MongoDB sẽ tránh cấu trúc table-based của relational database để thích ứng với các tài liệu như JSON có một schema rất linh hoạt gọi là BSON. MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau. Các dữ liệu được lưu trữ trong document kiểu JSON nên tốc độ truy vấn sẽ rất nhanh.

#### Ưu và nhược điểm của MongoDB:

1. Ưu điểm

* Dữ liệu lữu trữ phi cấu trúc, không có tính ràng buộc, hiệu suất lớn, dễ dàng mở rộng và lưu trữ.
* Dữ liệu được ghi đệm (cache) lên RAM, hạn chế truy cập vào phần cứng nên tốc độ đọc và ghi cao.

1. Nhược điểm

* Không được sử dụng cho các mô hình giao dịch cho các giao dịch yêu cầu độ chính xác cao do không có tính ràng buộc.
* Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy hoạt động yêu cầu một bộ nhớ RAM lớn.
* Khả năng mất dữ liệu khá cao do mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ở cứng ngay lập tức.

#### Tổng quan về Mongoose:

Mongoose là một thư viện mô hình hóa đối tượng (Object Data Model - ODM) cho MongoDB và Node.js. Nó quản lý mối quan hệ giữa dữ liệu, cung cấp sự xác nhận giản đồ và được sử dụng để dịch giữa các đối tượng trong mã và biểu diễn các đối tượng trong MongoDB.

#### Ưu và nhược điểm của Mongoose:

1. Ưu điểm

* Mongoose đã tích hợp khả năng xác thực dữ liệu, cho phép kiểm soát dữ liệu được thêm vào hoặc cập nhật vào database.
* Tiền định nghĩa model cho mỗi collection.
* Cho phép áp dụng tính ràng buộc lên mỗi document trong collection.

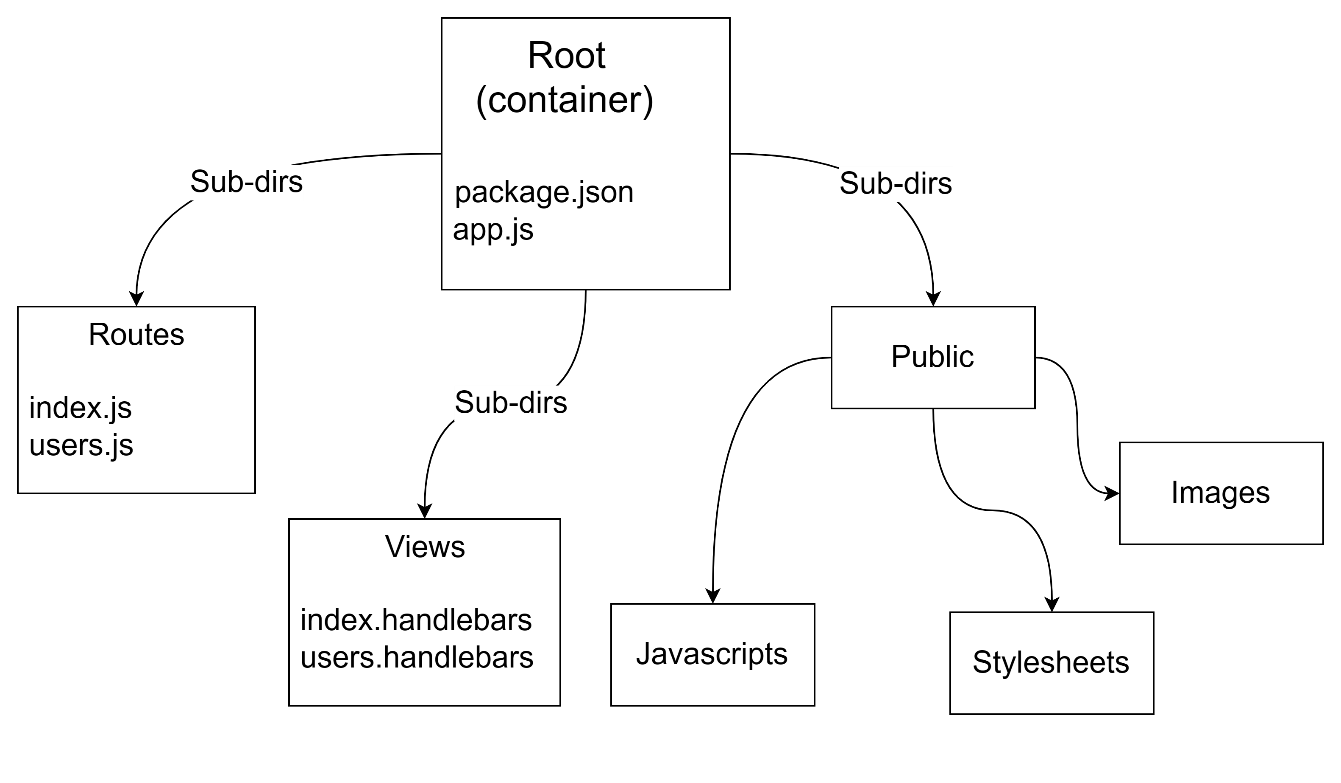
1. Nhược điểm

* Chậm hơn so với việc sử dụng trực tiếp MongoDB.
* Mongoose thêm một lớp trừu tượng (abstraction layer), điều này làm tăng độ phức tạp (complexity) của ứng dụng.

### ExpressJS

#### Tổng quan:

ExpressJS là một web framework tối giản và linh hoạt cho NodeJS, cung cấp các tính năng mạnh mẽ để phát triển các ứng dụng web và di động.



Hình 2.5: Cấu trúc project sử dụng Express

#### Ưu và nhược điểm:

1. Ưu điểm

* Nhẹ, tối giản, dễ học và sử dụng.
* Hỗ trợ middleware.
* Dễ dàng quản lý các định tuyến (routes).
* Hỗ trợ đa dạng các template engine (handlebars, jade, ejs,..v.v)
* Hỗ trợ đa dạng các database (MySQL, MongoDB)

1. Nhược điểm

* Cấu trúc (structure) và quy ước (convention) chưa thật sự chặt chẽ.
* Chưa cung cấp được kiểm lỗi thời gian thực, dẫn đến dễ phát sinh lỗi.
* Middleware gây sự lạ lẫm cho người mới.

### ReactJS

#### Tổng quan:

ReactJS một thư viện JavaScript mã nguồn mở được thiết kế bởi Facebook để tạo ra những ứng dụng web một cách nhanh chóng và hiệu quả. Mục đích cốt lõi của ReactJS không chỉ khiến cho trang web phải thật mượt mà còn phải nhanh, khả năng mở rộng cao và đơn giản.

Sức mạnh của nó xuất phát từ việc tập trung vào các thành phần (component) riêng lẻ. Chính vì vậy, thay vì làm việc trên toàn bộ ứng dụng web, ReactJS cho phép một developer có thể phá vỡ giao diện người dùng phức tạp thành các component nhỏ hơn.

#### Ưu và nhược điểm:

1. Ưu điểm

* Dễ học, dễ sử dụng.
* Tạo web động (dynamic) dễ dàng hơn.
* Các tái sử dụng component.
* Cải thiện hiệu suất (performance) do virtual DOM.

1. Nhược điểm

* Documentation (tài liệu) còn ít.
* Sự khó khăn với người mới do ReactJS sử dụng JSX (điều này sự kết hợp giữa JSX và HTML).

### NodeJS

#### Tổng quan:

NodeJS là môi trường thời gian chạy (runtime environment) JavaScript đa nền tảng và mã nguồn mở. NodeJS cho phép các lập trình viên tạo cả ứng dụng front-end và back-end bằng JavaScript.

#### Ưu và nhược điểm:

1. Ưu điểm

* Mã nguồn mở.
* Hỗ trợ đa nền tảng.
* Dễ dàng mở rộng.
* Tốc độ thực thi nhanh.
* Truyền dữ liệu nhanh.
* Không có bộ nhớ đệm.
* Cộng đồng, tài liệu lớn.
* Tiết kiệm thời gian, chi phí, công sức.

1. Nhược điểm

* Chỉ có đơn luồng (single-threaded).
* Có khả năng gây bottleneck cho CPU.
* Các vấn đề phức tạp với bất đồng bộ (asynchronous).
* Phụ thuộc nhiều vào các bên thứ 3 (các thư viện bên ngoài).

# DỮ LIỆU

## …Mai làm

# MÔ HÌNH ĐỀ XUẤT

## …Mô hình + thực nghiệm của mỗi mô hình

# KẾT QUẢ ĐẠT ĐƯỢC

## …Mấy cái UI

# KẾT LUẬN

## Kết luận

…

## Hướng phát triển

….

# TÀI LIỆU THAM KHẢO

Tiếng Việt

Ban biên tập Topdev (2018), https://topdev.vn/blog/mongodb-la-gi/

Hoang Hung (2018), https://viblo.asia/p/mongoose-cho-mongodb-nodejs-Qbq5QWvJZD8

Tuong Uyen (2023), https://itviec.com/blog/nodejs-la-gi/

Tiếng Anh

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural Computation*, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). *Attention Is All You Need* (arXiv:1706.03762). arXiv. https://doi.org/10.48550/arXiv.1706.03762

Zach\_ayy (2023), https://www.geeksforgeeks.org/what-are-the-advantages-of-using-mongoose-module/

Express (2024), https://expressjs.com/

Kevin Kononenko (2017), https://www.freecodecamp.org/news/going-out-to-eat-and-understanding-the-basics-of-express-js-f034a029fb66

DataFlair contributor (2023), https://data-flair.training/blogs/expressjs-advantages-and-disadvantages/

EPAM Anywhere Editorial (2024), https://anywhere.epam.com/en/blog/node-js-pros-and-cons