ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

BÁO CÁO BÀI TẬP LỚN

MÔN HỌC: KỸ THUẬT VÀ CÔNG NGHỆ DỮ LIỆU LỚN

ELASTIC CHATBOT RAG APP

Sinh viên: Đàm Văn Hiển - 22022664
Nguyễn Xuân Hiệp - 22022591
Nguyễn Thế An - 22022649
Bùi Duy Hải - 22022575

Giảng viên hướng dẫn: TS Trần Hồng Việt
ThS Ngô Minh Hương

HÀ NỘI - 12/2024

LỜI MỞ ĐẦU

Lời đầu tiên chúng em xin được gửi lời cảm ơn chân thành và sâu sắc nhất tới 2 Giảng viên: TS Trần Hồng Việt và ThS Ngô Minh Hương đã cung cấp những kiến thức bổ ích phục vụ trực tiếp cho quá trình viết bài báo cáo này.

Trong quá trình làm báo cáo này, do trình độ và kinh nghiệm thực tiễn còn hạn chế, chúng em không thể tránh khỏi những thiếu sót. Chúng em rất mong được sự chỉ dẫn và góp ý từ Cô nhằm khắc phục những khuyết điểm này.

Mục lục

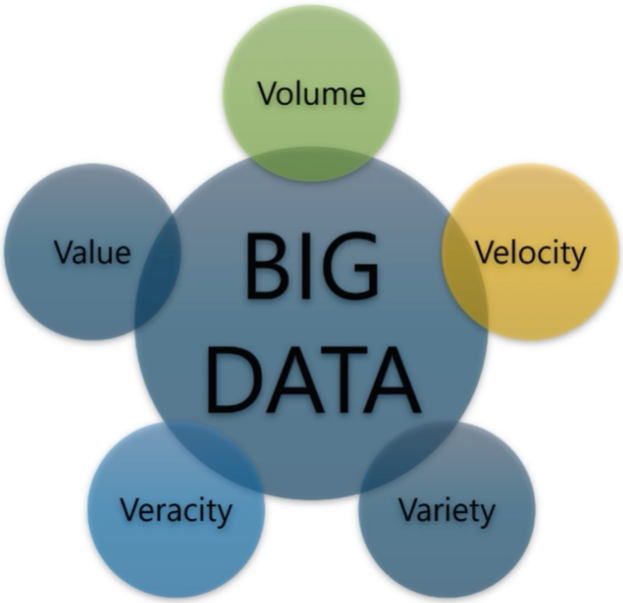
BẢNG PHÂN CHIA CÔNG VIỆC

Công việc	Hiệp	Hiển	An	Hải
Tìm hiểu ElasticSearch		✓		✓
Tìm hiểu RAG	✓			✓
Thêm dữ liệu vào hệ thống				✓
Làm slide			✓	✓
Viết report	✓	✓		
Viết pipeline		✓		
Demo	✓			

1 Tổng quan về ElasticSearch

1.1 Giới thiệu về dữ liệu lớn

Theo Wikipedia: Dữ liệu lớn là một thuật ngữ chỉ bộ dữ liệu lớn hoặc phức tạp mà các phương pháp truyền thống không đủ các ứng dụng để xử lý dữ liệu này. Theo Gartner: Dữ liệu lớn là những nguồn thông tin có đặc điểm chung khối lượng lớn, tốc độ nhanh và dữ liệu định dạng dưới nhiều hình thức khác nhau,do đó muốn khai thác được phải đòi hỏi phải có hình thức mới để đưa ra quyết định khám phá và tối ưu hóa quy trình. vực khác nhau.



Các đặc trưng cơ bản của dữ liệu lớn (5V)

- **Volume (Khối lượng)**: Lượng dữ liệu khổng lồ được tạo ra từ nhiều nguồn khác nhau, như mạng xã hội, cảm biến IoT, giao dịch thương mại, email, ảnh, video, v.v. Dữ liệu lớn thường có kích thước từ terabyte đến petabyte hoặc hơn.
- **Velocity (Tốc độ)**: có thể hiểu theo 2 khía cạnh, tốc độ sinh ra, xử lý và truyền tải dữ liệu rất nhanh hoặc các dữ liệu yêu cầu xử lí thời gian thực (real-time).
- **Variety (Đa dạng)**: Dữ liệu có cấu trúc (cơ sở dữ liệu quan hệ), dữ liệu phi cấu trúc (văn bản, ảnh, video, âm thanh), dữ liệu bán cấu trúc(JSON, XML)

- **Veracity (Tính xác thực)**: Dữ liệu lớn có thể chứa thông tin không chính xác, nhiễu, hoặc thiếu độ tin cậy, xử lý dữ liệu và đảm bảo chất lượng là một thách thức lớn.
- **Value (Giá trị)**: đây là đặc trưng quan trọng nhất của dữ liệu lớn. Việc trích xuất thông tin hữu ích từ dữ liệu lớn có thể hỗ trợ ra quyết định, tối ưu hóa quy trình kinh doanh, và mang lại lợi ích kinh tế hoặc xã hội.

1.2 Giới thiệu về ElasticSearch

1.2.1 ElasticSearch là gì?

Elasticsearch là một công cụ tìm kiếm và phân tích dữ liệu phân tán, mã nguồn mở, được xây dựng trên nền tảng Apache Lucene. Đây là một thành phần cốt lõi của Elastic Stack (gồm Elasticsearch, Logstash, Kibana và Beats) và thường được sử dụng để tìm kiếm, phân tích và trực quan hóa dữ liệu trong thời gian thực.

Elasticsearch được sử dụng bởi nhiều công ty lớn trên toàn cầu nhờ khả năng tìm kiếm và phân tích dữ liệu nhanh chóng, mạnh mẽ. Các công ty như Netflix, LinkedIn, Uber và Slack dùng Elasticsearch để giám sát hệ thống, tìm kiếm dữ liệu và phân tích log thời gian thực. Trong lĩnh vực thương mại điện tử, eBay, Shopify, và Walmart tận dụng Elasticsearch để tối ưu hóa tìm kiếm và cải thiện trải nghiệm khách hàng.

423 systems in ranking, December 2024

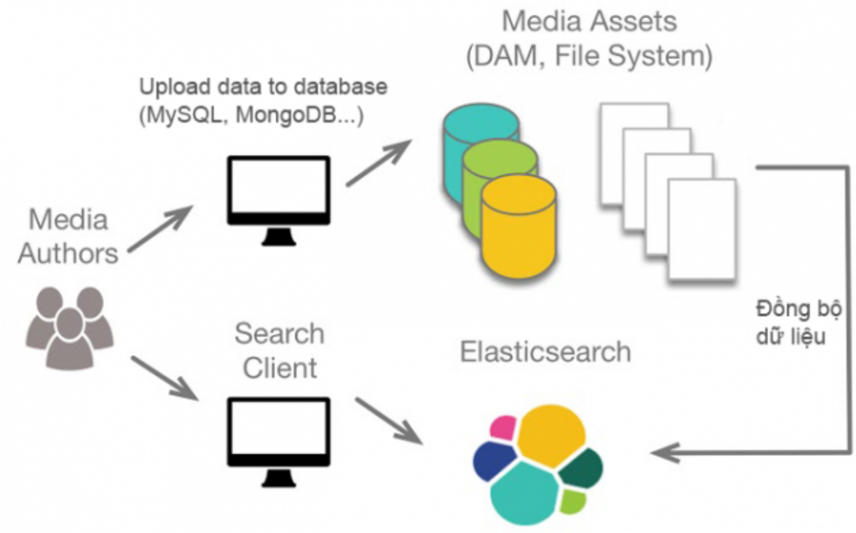
Rank			DBMS	Database Model	Score		
Dec 2024	Nov 2024	Dec 2023			Dec 2024	Nov 2024	Dec 2023
1.	1.	1.	Oracle	Relational, Multi-model	1263.79	-53.22	+6.38
2.	2.	2.	MySQL	Relational, Multi-model	1003.76	-14.04	-122.88
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	805.69	+5.88	-98.14
4.	4.	4.	PostgreSQL	Relational, Multi-model	666.37	+12.04	+15.47
5.	5.	5.	MongoDB	Document, Multi-model	400.39	-0.54	-18.76
6.	6.	6.	Redis	Key-value, Multi-model	150.27	+1.63	-8.08
7.	7.	↑ 10.	Snowflake	Relational	147.36	+4.87	+27.48
8.	8.	↓ 7.	Elasticsearch	Multi-model	132.32	+0.68	-5.43
9.	9.	↓ 8.	IBM Db2	Relational, Multi-model	122.78	+1.04	-11.81
10.	10.	↑ 11.	SQLite	Relational	101.72	+2.24	-16.23

Dựa vào bảng xếp hạng trong hình, Elasticsearch hiện đang đứng thứ 8 trong số 423 hệ thống cơ sở dữ liệu được xếp hạng vào tháng 12/2024. Điều này cho thấy Elasticsearch vẫn là một công cụ phổ biến và được sử dụng rộng rãi, đặc biệt là trong các hệ thống yêu cầu tìm kiếm và phân tích dữ liệu nhanh chóng.

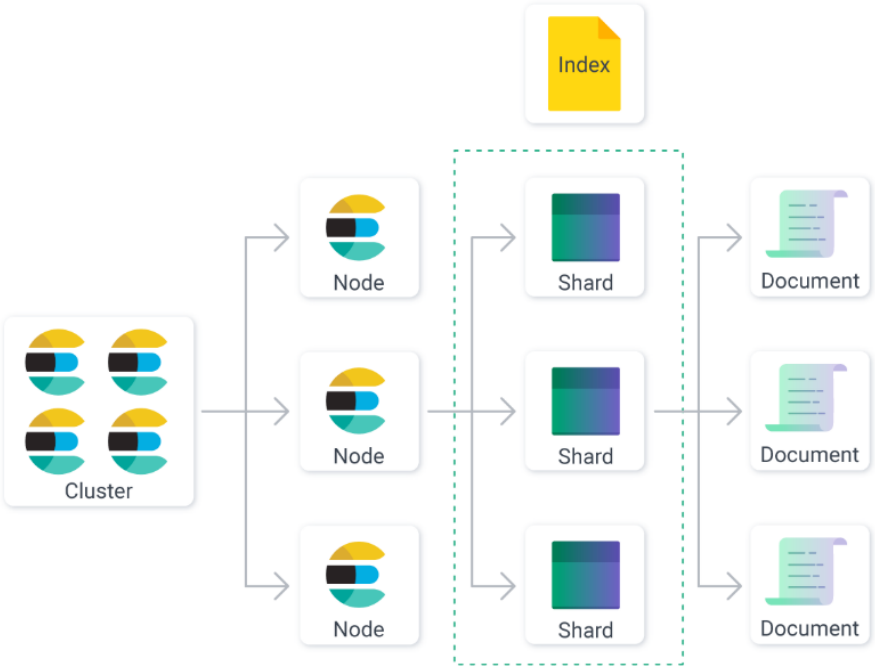
1.2.2 Đặc điểm nổi bật của ElasticSearch

- **Tìm kiếm mạnh mẽ:** Elasticsearch hỗ trợ tìm kiếm toàn văn bản (full-text search) với khả năng xử lý dữ liệu phi cấu trúc rất tốt. Nó có thể xếp hạng các kết quả tìm kiếm dựa trên mức độ liên quan.
- **Phân tán (Distributed):** Elasticsearch được thiết kế để chạy trên nhiều máy chủ, chia dữ liệu thành các shard và tạo bản sao để đảm bảo tính sẵn sàng cao và khả năng mở rộng.
- **Thời gian thực (Near Real-Time):** Cập nhật dữ liệu và tìm kiếm hầu như diễn ra ngay lập tức, giúp hỗ trợ các hệ thống cần phân tích nhanh.
- **Dễ mở rộng:** Bạn có thể thêm hoặc xóa các node (máy chủ) mà không làm gián đoạn hoạt động của hệ thống.
- **APIRestful:** Elasticsearch sử dụng giao diện API RESTful để tương tác với người dùng, giúp tích hợp dễ dàng với các ngôn ngữ lập trình khác nhau.
- **Khả năng phân tích dữ liệu:** Ngoài tìm kiếm, Elasticsearch có thể thực hiện các phép tính thống kê phức tạp, giúp phân tích dữ liệu một cách linh hoạt.

1.2.3 Cách hoạt động của ElasticSearch



Elasticsearch là 1 server riêng biệt để "phục vụ" việc tìm kiếm dữ liệu. ES sẽ chạy một cổng (dưới local default là 9200). ES không mạnh trong các thao tác CRUD, nên thường sẽ dùng song song với 1 DB chính (SQL, MySQL, MongoDB . . .)



- **Dữ liệu được lưu trong các chỉ mục (Index):** Một chỉ mục chứa nhiều tài liệu (Document), và mỗi tài liệu được mô tả bằng một số trường (Field). Dữ liệu được lập chỉ mục để tối ưu hóa tốc độ tìm kiếm.
- **Sharding và Replication:** Một chỉ mục có thể được chia thành nhiều shard để tăng hiệu suất. Mỗi shard có thể có bản sao (replica) để đảm bảo dữ liệu không bị mất khi hệ thống gặp sự cố.
- **Công cụ tìm kiếm toàn văn bản (Full-text search):** Elasticsearch sử dụng bộ máy Lucene để phân tích văn bản, tìm kiếm và xếp hạng kết quả theo mức độ liên quan.

1.2.4 Ứng dụng ElasticSearch

- Hệ thống tìm kiếm trên website (như trang thương mại điện tử, tin tức).
- Phân tích log: Dùng để phân tích dữ liệu log và phát hiện các vấn đề trong hệ thống (kết hợp với Logstash và Kibana).

- Giám sát dữ liệu: Phát hiện bất thường trong thời gian thực.
- Hệ thống đề xuất: Gợi ý sản phẩm/dịch vụ dựa trên dữ liệu tìm kiếm của người dùng.

2 Tổng quan về Retrieval-Augmented Generation (RAG)

2.1 Kiến trúc RAG

2.1.1 RAG là gì?

Retrieval-Augmented Generation (RAG) là một kiến trúc AI kết hợp giữa khả năng truy xuất thông tin từ bên ngoài (Retrieval) và khả năng sinh ngôn ngữ tự nhiên (Generation). Mô hình này được thiết kế để giải quyết các vấn đề mà mô hình ngôn ngữ truyền thống (LLMs) gặp phải, bằng cách bổ sung khả năng tiếp cận dữ liệu từ các nguồn tri thức lớn hoặc thời gian thực.

2.1.2 Các thành phần trong kiến trúc RAG

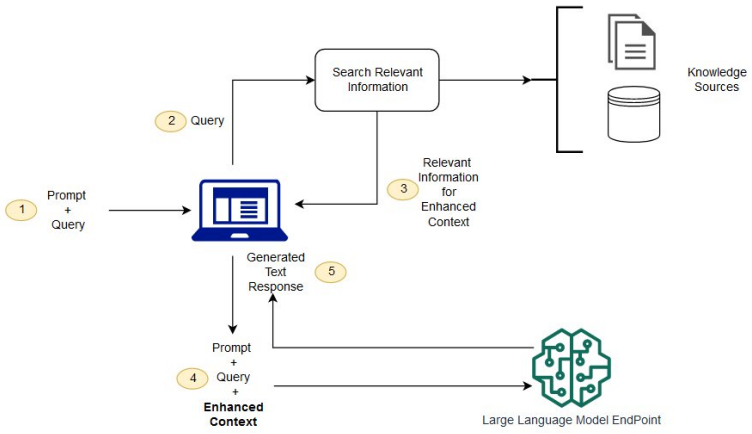
RAG là một mô hình kết hợp giữa truy xuất dữ liệu và sinh câu trả lời. Các thành phần trong kiến trúc này hoạt động kết hợp với nhau để cải thiện độ chính xác và tính liên quan của các câu trả lời mà mô hình tạo ra. Sau đây là ba thành phần chính trong kiến trúc RAG:

1. **Trình thu thập dữ liệu (Retriever)**: Đây là thành phần chịu trách nhiệm tìm kiếm và thu thập thông tin liên quan từ cơ sở dữ liệu hoặc các nguồn tài liệu bên ngoài dựa trên truy vấn đầu vào của người dùng.
2. **Trình tạo nội dung (Generator)**: Thành phần này sẽ tạo ra câu trả lời cuối cùng dựa trên thông tin thu thập được từ trình thu thập dữ liệu và truy vấn đầu vào của người dùng.Thường là một mô hình ngôn ngữ lớn như GPT-3, có khả năng tổng hợp và tạo ra văn bản mạch lạc và liên quan.

2.1.3 RAG hoạt động như thế nào?

Khi không sử dụng RAG, LLM chỉ dựa vào thông tin mà nó đã được huấn luyện trước đó để phản hồi các truy vấn từ người dùng. Cụ thể, LLM sẽ nhận đầu vào từ người dùng, sau đó tạo ra câu trả lời dựa trên những kiến thức sẵn có trong tập dữ liệu huấn luyện. Tuy nhiên, cách tiếp cận này có hạn chế lớn, vì LLM không thể cập nhật thông tin mới hoặc xử lý các dữ liệu đặc thù của tổ chức hay doanh nghiệp, dẫn đến câu trả lời có thể không phù hợp hoặc thiếu chính xác trong một số trường hợp.

Khi sử dụng RAG, hệ thống được cải tiến bằng cách tích hợp một thành phần truy xuất thông tin (**information retrieval component**). Trước tiên, thành phần này sử dụng truy vấn của người dùng để tìm kiếm và lấy dữ liệu liên quan từ các nguồn bên ngoài. Sau đó, dữ liệu truy xuất được kết hợp với truy vấn gốc của người dùng và gửi cho LLM. Nhờ đó, LLM có thể sử dụng cả thông tin mới và kiến thức đã được huấn luyện để tạo ra câu trả lời phù hợp hơn với nhu cầu thực tế của người dùng.



- **Bước 1 - Tạo dữ liệu bên ngoài**: Dữ liệu bên ngoài là những thông tin nằm ngoài tập dữ liệu ban đầu mà mô hình ngôn ngữ lớn được huấn luyện, được lấy từ nhiều nguồn khác nhau như API, cơ sở dữ liệu, kho tài liệu. Sau đó được chuyển đổi thành các biểu diễn số học bằng các mô hình nhúng ngôn ngữ và lưu trữ trong cơ sở dữ liệu vector, tạo thành thư viện kiến thức mà các mô hình AI tạo sinh có thể sử dụng.

- **Bước 2 - Lấy thông tin liên quan**: Trong bước này, câu hỏi của người dùng được chuyển đổi thành một biểu diễn vector và tìm kiếm sự liên quan với cơ sở dữ liệu vector. Mức độ liên quan được tính toán và xác định thông qua các phép toán vector và biểu diễn toán học.

- **Bước 3 - Tăng cường câu hỏi (prompt) của LLM**: Sau khi truy xuất thông tin liên quan, bước tiếp theo là kết hợp dữ liệu này với truy vấn gốc của người dùng. Đây được gọi là tăng cường prompt của LLM. Các kỹ thuật xây dựng prompt được áp dụng để đảm bảo rằng LLM nhận được một đầu vào có cấu trúc tốt, bao gồm cả truy vấn của người dùng và dữ liệu ngoài liên quan. Prompt đã được tăng cường này cho phép LLM tạo ra những câu trả lời chính xác.

2.2 Tại sao sử dụng RAG?

2.2.1 Vấn đề của mô hình ngôn ngữ lớn truyền thống

Trong khi các mô hình ngôn ngữ lớn (LLMs) đã đạt được nhiều thành tựu ấn tượng trong các tác vụ xử lý ngôn ngữ tự nhiên, chúng vẫn gặp phải một số vấn đề lớn, đặc biệt khi áp dụng vào các tình huống đòi hỏi sự cập nhật kiến thức liên tục và khả năng đưa ra câu trả lời chính xác dựa trên thông tin thời gian thực

- **Không thể cập nhật kiến thức theo thời gian**: Các LLMs như GPT hoặc BERT chỉ học và tạo ra câu trả lời dựa trên dữ liệu mà chúng được huấn luyện trước đó. Khi một mô hình không thể cập nhật kiến thức sau khi được huấn luyện, nó sẽ không thể phản ánh các sự kiện, thông tin hay xu hướng mới.

- **Hiện tượng "hallucination"**: Hiện tượng "hallucination"xảy ra khi LLMs tạo ra thông tin sai lệch hoặc không có thực. Điều này là do các mô hình ngôn ngữ không luôn dựa trên sự thật mà thay vào đó là các xác suất dựa trên dữ liệu mà chúng đã học.

- **Tính không chuyên sâu**: LLMs thường thiếu khả năng trả lời các câu hỏi yêu cầu kiến thức chi tiết, chuyên sâu hoặc những tài liệu cụ thể mà chúng chưa được huấn luyện. Nếu câu hỏi yêu cầu thông tin từ một lĩnh vực rất chuyên môn hoặc một nguồn tài liệu đặc thù mà mô hình chưa tiếp cận, mô hình sẽ không thể cung cấp câu trả lời chính xác.

- **Chi phí tái huấn luyện cao**: Việc huấn luyện lại một LLM để cập nhật kiến thức hoặc thông tin mới là rất tốn kém và không khả thi trong nhiều trường hợp, đặc biệt là khi cần phải có phản hồi nhanh chóng và liên tục. Để cập nhật

mô hình, cần phải thu thập dữ liệu mới và huấn luyện lại mô hình, điều này yêu cầu tài nguyên tính toán và thời gian rất lớn.

2.2.2 Ý nghĩa khi sử dụng RAG

RAG mang lại một cách tiếp cận hoàn toàn mới để tích hợp dữ liệu bên ngoài vào quá trình tạo câu trả lời.

- **Cải thiện độ chính xác**: Nhờ việc truy xuất thông tin từ cơ sở dữ liệu ngoài, RAG cung cấp các câu trả lời chính xác hơn dựa trên dữ liệu thực tế và được xác thực.
- **Hiểu ngữ cảnh tốt hơn**: RAG không chỉ dựa vào dữ liệu đã huấn luyện mà còn truy xuất và kết hợp thông tin từ các nguồn kiến thức bên ngoài, giúp mô hình hiểu rõ hơn các câu hỏi và tạo ra các câu trả lời chính xác hơn, phù hợp với ngữ cảnh.
- **Giảm thiểu thiên vị và thông tin sai lệch**: Vì RAG dựa vào các nguồn kiến thức đã được xác minh, nó giúp giảm thiểu nguy cơ sai lệch và lan truyền thông tin sai trong quá trình sinh tạo câu trả lời.
- **Đa dụng**: RAG có thể được áp dụng trong nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên, như trả lời câu hỏi, chatbot, hay tạo nội dung.

2.2.3 Lợi ích

- **Câu trả lời cập nhật và chính xác**: RAG giúp đảm bảo các câu trả lời dựa trên nguồn dữ liệu bên ngoài hiện tại, từ đó giảm thiểu rủi ro cung cấp thông tin lỗi thời hoặc không chính xác.
- **Giảm sai sót và "hallucination"**: Khi các câu trả lời được gắn với kiến thức bên ngoài có liên quan, RAG giúp giảm khả năng mô hình tạo ra thông tin không chính xác hoặc hư cấu, được gọi là hiện tượng "hallucination"trong các mô hình ngôn ngữ.

- **Câu trả lời đặc thù và phù hợp với lĩnh vực**: RAG cho phép mô hình tạo ra câu trả lời phù hợp với ngữ cảnh và đặc thù của một tổ chức hoặc lĩnh vực, giúp nâng cao chất lượng của các câu trả lời.

- **Hiệu quả và tiết kiệm chi phí**: RAG là một phương pháp đơn giản và tiết kiệm chi phí để tùy chỉnh các mô hình ngôn ngữ lớn (LLMs) với dữ liệu đặc thù của lĩnh vực, vì nó không yêu cầu tùy chỉnh hoặc huấn luyện lại mô hình một cách phức tạp.

2.2.4 Hạn chế

- **Phức tạp**: RAG kết hợp hai thành phần chính là quá trình truy xuất (retrieval) và sinh dữ liệu (generation). Việc này làm cho mô hình trở nên phức tạp hơn vì cần phải tối ưu hóa và điều chỉnh cả hai phần sao cho chúng hoạt động hài hòa với nhau. Điều này có thể đòi hỏi nhiều thời gian và công sức để đảm bảo mô hình không chỉ truy xuất dữ liệu chính xác mà còn sinh ra phản hồi chất lượng.

- **Độ trễ**: Trong quá trình RAG, bước truy xuất dữ liệu có thể làm tăng độ trễ (latency), đặc biệt là khi số lượng dữ liệu cần phải truy xuất rất lớn. Điều này có thể gây khó khăn khi muốn triển khai mô hình trong các ứng dụng yêu cầu phản hồi nhanh và thời gian thực, ví dụ như chatbots hoặc trợ lý ảo trong các dịch vụ khách hàng.

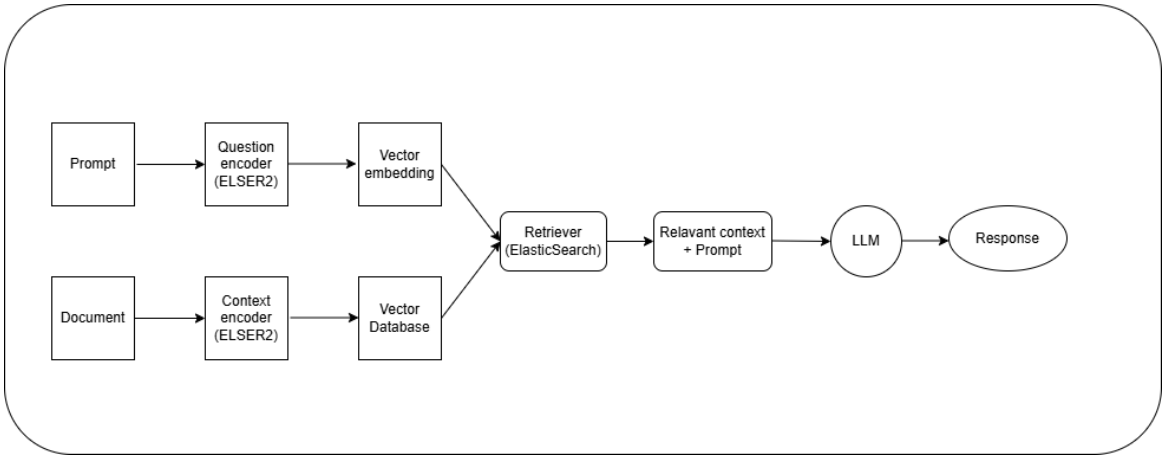
- **Chất lượng truy xuất**: Hiệu quả của RAG rất phụ thuộc vào chất lượng của các tài liệu được truy xuất từ cơ sở dữ liệu. Nếu hệ thống không thể truy xuất được những tài liệu có liên quan hoặc tài liệu không đủ chính xác, kết quả sinh ra sẽ không chính xác và làm giảm hiệu quả của mô hình. Ví dụ, nếu mô hình truy xuất sai dữ liệu, câu trả lời sinh ra có thể không liên quan đến câu hỏi của người dùng.

- **Thiên vị và công bằng**: RAG có thể kế thừa các thiên vị có trong dữ liệu huấn luyện ban đầu và tài liệu được truy xuất. Nếu dữ liệu huấn luyện hay tài liệu trong cơ sở dữ liệu có sự thiên vị (ví dụ: thiên vị văn hóa, giới tính hoặc chủng tộc), thì mô hình có thể tạo ra kết quả không công bằng hoặc phản ánh những định kiến sai lệch. Việc này yêu cầu phải có sự giám sát và nỗ lực liên tục để

đảm bảo rằng dữ liệu được xử lý công bằng và không làm tăng thiên vị trong các câu trả lời.

3 Ứng dụng ElasticSearch và RAG để xây dựng Chatbot

3.1 PipeLine



Hình trên mô tả pipeline của hệ thống Elastic RAG Chatbot, sau đây là mô tả chi tiết của từng thành phần:

- Prompt từ người dùng: Người dùng nhập Prompt (câu hỏi hoặc truy vấn), prompt sẽ được xử lý và gửi vào hệ thống.
- Question Encoder (ELSER2): mã hóa câu input của người dùng thành vector embedding
- Context Encoder: mã hóa các tài liệu thành các vector rồi lưu vào Vector Database (thống qua ELasticSearch)
- Retriever: sử dụng ElasticSearch, Vector embedding của câu hỏi từ người dùng được so sánh với các vector của tài liệu trong Vector Database. Retriever sử dụng phương pháp k-Nearest Neighbors (kNN) để tìm các tài liệu liên quan nhất rồi trả về danh sách các tài liệu có liên quan.
- LLM: kết hợp ngữ cảnh (relevant context) và câu hỏi gốc thành một prompt mới. Prompt mới thông qua API được gửi đến các LLM để sinh ra câu trả lời cuối cùng.

3.2 Demo

Bước 1: cài đặt môi trường ảo, kích hoạt môi trường và cài đặt các gói phụ thuộc của python

```
# Create a virtual environment
python -m venv .venv

# Activate the virtual environment
source .venv/bin/activate

# Install Python dependencies
pip install -r requirements.txt
```

Bước 2: Chạy lệnh yarn với mục đích là cài đặt tất cả các gói được liệt kê trong file package.json của thư mục frontend.

```
# Install Node dependencies
cd frontend && yarn && cd ..
```

Bước 3: Chuẩn bị dữ liệu nằm ở file .json và lưu trữ vào Elasticsearch để hỗ trợ tìm kiếm và phân tích sau này.

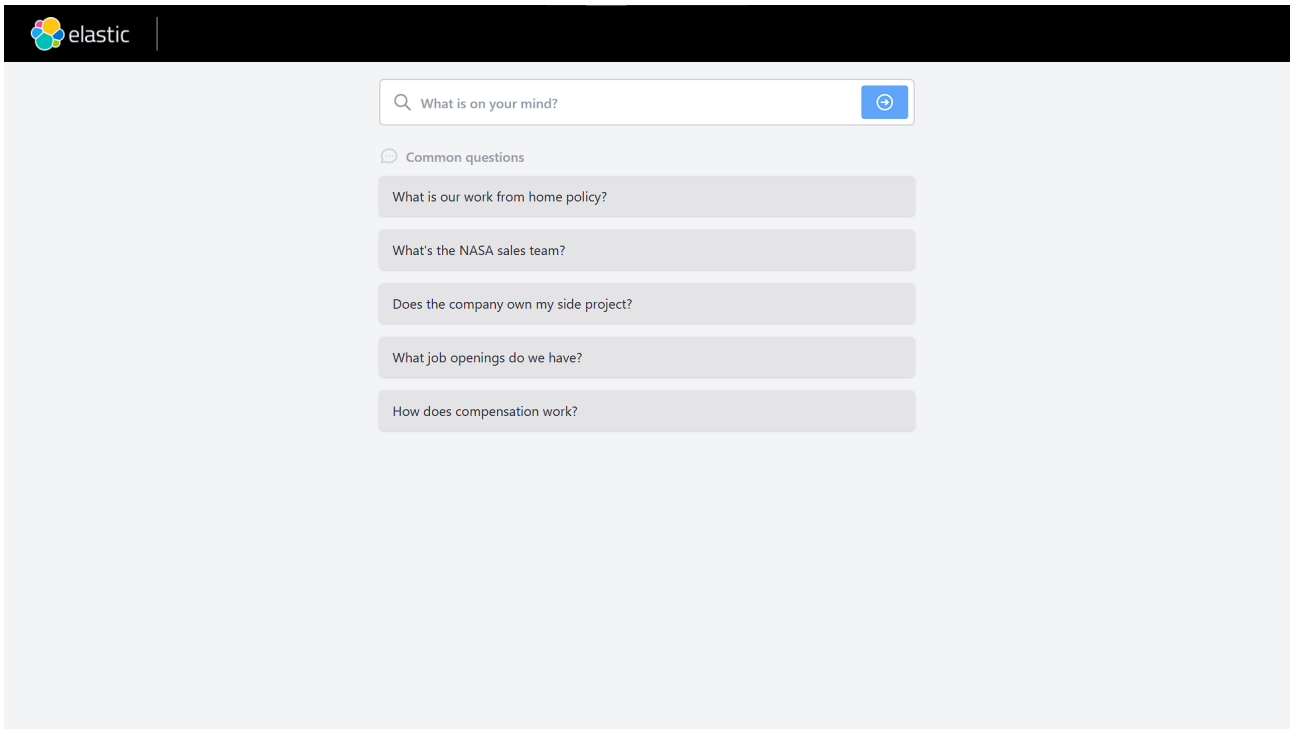
```
flask create-index
```

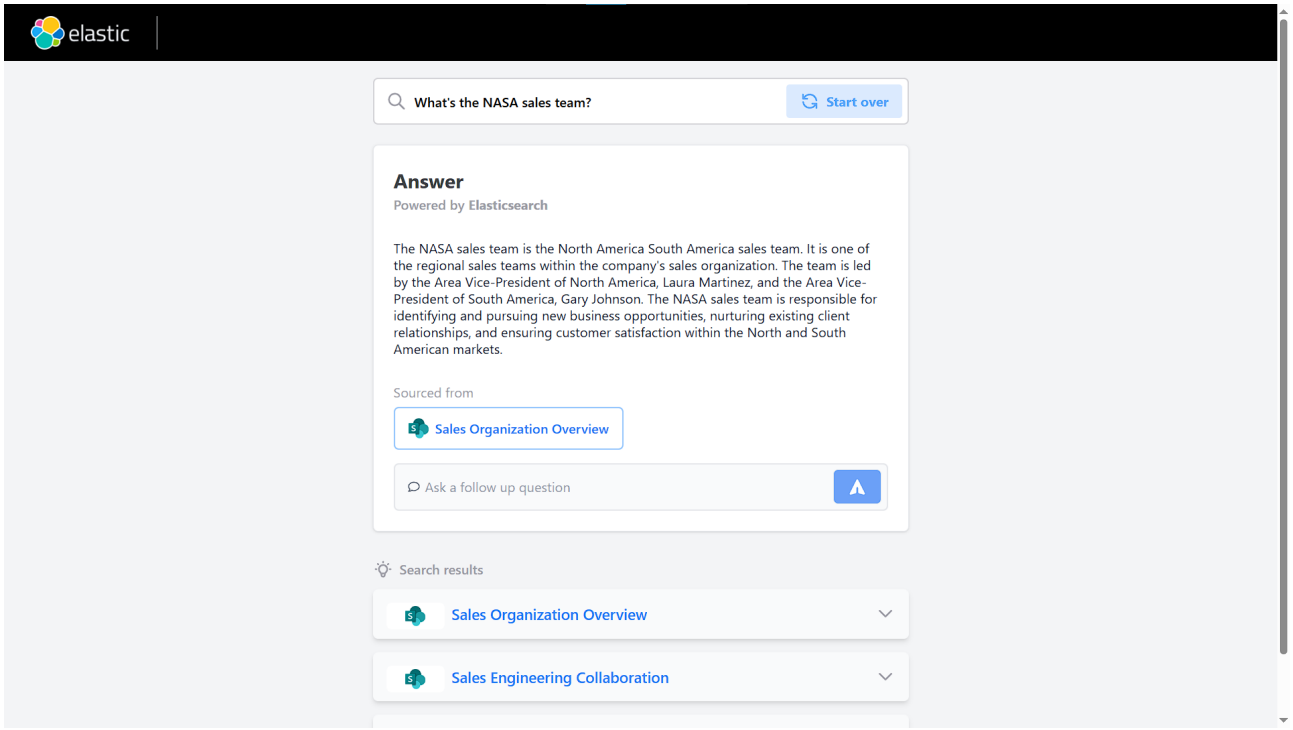
Bước 4: Khởi chạy API backend bằng Flask và frontend ứng dụng bằng React (yarn start) để người dùng truy cập giao diện tại http://localhost:3000 với tính năng tự động cập nhật khi có thay đổi.

```
# Launch API app
flask run

# In a separate terminal launch frontend app
cd frontend && yarn start
```

Sau khi hoàn thành các bước trên, truy cập vào cổng http://localhost:3000 để sử dụng giao diện kết nối với Elasticsearch, nơi bạn có thể đặt câu hỏi và nhận câu trả lời dựa trên dữ liệu đã được index bởi bạn.


elastic
What is on your mind?
Common questions
What is our work from home policy?
What's the NASA sales team?
Does the company own my side project?
What job openings do we have?
How does compensation work?


elastic
What's the NASA sales team?
Start over
Answer
Powered by Elasticsearch
The NASA sales team is the North America South America sales team. It is one of the regional sales teams within the company's sales organization. The team is led by the Area Vice-President of North America, Laura Martinez, and the Area Vice-President of South America, Gary Johnson. The NASA sales team is responsible for identifying and pursuing new business opportunities, nurturing existing client relationships, and ensuring customer satisfaction within the North and South American markets.
Sourced from
Sales Organization Overview
Ask a follow up question
Search results
Sales Organization Overview
Sales Engineering Collaboration

4 Kết luận và hướng phát triển

4.1 Kết luận

Sau quá trình thực hiện đề tài, nhóm đã đạt được một số kết quả sau:

- Hiểu tổng quan về kiến trúc RAG, Elastic Search và vai trò của từng thành phần.
- Triển khai thành công pipeline:
  - Question Encoder và Context Encoder sử dụng mô hình ELSER2.
  - Lưu trữ vector embedding vào Elasticsearch để hỗ trợ tìm kiếm kNN.
  - Tích hợp Large Language Model (GPT-based) để sinh câu trả lời.
- Xây dựng thành công chương trình demo chatbot:
  - Nhập câu hỏi, thực hiện tìm kiếm tài liệu liên quan từ Elasticsearch.
  - Kết hợp ngữ cảnh và câu hỏi, gửi tới LLM để sinh câu trả lời.
- Đánh giá độ chính xác và tốc độ xử lý của chương trình.

Mặc dù đạt được nhiều kết quả tích cực, chương trình vẫn còn một số hạn chế:

- Chưa tối ưu hóa hoàn toàn pipeline tìm kiếm và sinh câu trả lời.
- Chưa hỗ trợ nhiều ngôn ngữ (hiện tại chỉ dùng tiếng Anh).

4.2 Hướng phát triển

Vì thời gian thực hiện có hạn, chúng em chưa thể tối ưu ứng dụng này. Trong tương lai, chúng em có dự định phát triển RAG chatbot app với các hướng cụ thể sau:

- Mở rộng hỗ trợ dữ liệu lớn (Big Data):Hiện tại hệ thống chỉ thử nghiệm trên tập dữ liệu nhỏ, nhóm sẽ mở rộng phạm vi áp dụng cho tập dữ liệu lớn hơn. Sử dụng các công cụ tối ưu truy vấn như Elasticsearch Cluster, hoặc tích hợp các hệ thống phân tán như Apache Spark để xử lý dữ liệu lớn hiệu quả hơn.

- Tối ưu hóa truy vấn trong Elasticsearch bằng cách: Kết hợp giữa keyword search và vector search để cải thiện độ chính xác và tốc độ truy vấn. Sử dụng chỉ số ANN (Approximate Nearest Neighbors) để tìm kiếm nhanh hơn trong không gian embedding. Cải thiện mô hình ngôn ngữ (LLM):
- Tăng cường trải nghiệm người dùng:tiếp tục phát triển giao diện người dùng thân thiện (frontend) để người dùng dễ dàng nhập câu hỏi và nhận phản hồi.

Trong quá trình hoàn thành bài tập lớn, nhóm em đã cố gắng tìm hiểu và tham khảo các tài liệu liên quan. Tuy nhiên, thời gian có hạn nên chúng em sẽ không tránh khỏi những thiếu sót, rất mong nhận được sự đóng góp ý kiến của thầy cô và các bạn để báo cáo và kỹ năng của chúng em ngày được hoàn thiện hơn. Em xin chân thành cảm ơn!

Tài liệu tham khảo

- https://github.com/elastic/elasticsearch-labs/tree/main/example-apps/chatbot-rag-app
- https://topdev.vn/blog/elasticsearch-la-gi/
- https://www.elastic.co/
- https://medium.com/enterprise-rag/a-first-intro-to-complex-rag-retrieval-augmented-generation-a8624d70090f
- https://www.geeksforgeeks.org/what-is-retrieval-augmented-generation-rag/
- https://www.youtube.com/watch?v=T-D1OfcDW1Mt=18s