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

BÁO CÁO CUỐI KỲ MÔN DỰ ÁN CÔNG NGHỆ THÔNG TIN

TÌM HIỂU VÀ ỨNG DỤNG ELASTICSEARCH CHO TÌM KIẾM TIN TỨC VỀ PHÁP LUẬT

Người hướng dẫn: **TS. LÊ ANH CƯỜNG**
Người thực hiện: **VINH LINH – 52000910**
VÕ THỊ QUẾ CHI – 52000741
Lớp : **20050301**
Khoá : **24**

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

BÁO CÁO CUỐI KỲ MÔN DỰ ÁN CÔNG NGHỆ THÔNG TIN

TÌM HIỂU VÀ ỨNG DỤNG ELASTICSEARCH CHO TÌM KIẾM TIN TỨC VỀ PHÁP LUẬT

Người hướng dẫn: **TS. LÊ ANH CƯỜNG**
Người thực hiện: **VINH LINH – 52000910**
VÕ THỊ QUẾ CHI – 52000741
Lớp : **20050301**
Khoá : **24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

LỜI CẢM ƠN

Để hoàn thành dồ án này chung em xin gửi đến thầy LÊ ANH CƯỜNG lời cảm ơn chân thành và sâu sắc nhât. Thầy đã tận tình hướng dẫn, giúp đỡ chúng em hoàn thành đề tài này. Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của sinh viê, bài báo cáo này không thể tránh khỏi được những thiếu sót. Chúng em rất mong nhận được sự giúp đỡ, đóng góp ý kiến từ thầy để chúng em nắm rõ hơn về kiến thức của môn áp dụng vào thực tế.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH
TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng em và được sự hướng dẫn của TS. LÊ ANH CƯỜNG ;. 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 đồ 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 đồ 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

Tác giả

(ký tên và ghi rõ họ tên)

Vinh Linh

Võ Thị Quế Chi

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Trình bày tóm tắt vấn đề nghiên cứu, các hướng tiếp cận, cách giải quyết vấn đề và một số kết quả đạt được, những phát hiện cơ bản trong vòng 1 -2 trang.

MỤC LỤC

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

f *Tần số của dòng điện và điện áp (Hz)*

p *Mật độ điện tích khối (C/m3)*

CÁC CHỮ VIẾT TẮT

CSTD	Công suất tác dụng
MF	Máy phát điện
BER	Tỷ lệ bít lỗi

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

DANH MỤC BẢNG

CHƯƠNG 1 – GIỚI THIỆU BÀI TOÁN

1.1 Tổng quan về bài toán

Trong thời đại số ngày nay, sự phổ biến của thông tin trên internet đã tạo nên một thách thức lớn đối với việc tìm kiếm và truy cập thông tin đáng tin cậy. Trong bối cảnh pháp luật, nhu cầu tìm kiếm thông tin pháp lý chính xác và nhanh chóng ngày càng trở nên quan trọng. Vì nhận thức về vấn đề này đã thúc đẩy đã thúc đẩy chúng em đến với tìm hiểu và ứng dụng Elasticsearch cho việc xây dựng một hệ thống tìm kiếm tin tức pháp luật bằng Chatbot.

Elasticsearch là một công cụ mạnh mẽ về tìm kiếm và phân tích dữ liệu, đã chứng minh sức mạnh của mình trong việc xử lý lượng thông tin lớn và đa dạng. Qua dự án này, chúng em sẽ khám phá cách Elasticsearch có thể được tích hợp vào hệ thống của chúng em để cải thiện hiệu suất tìm kiếm, đồng thời tạo ra một trải nghiệm người dùng thuận lợi và hiệu quả.

Dự án này không chỉ mang lại kiến thức về Elasticsearch mà còn góp phần giải quyết một vấn đề thiết thực trong lĩnh vực pháp luật và tin tức. Chúng em hy vọng kết quả của dự án này sẽ là một bước tiến tốt trong việc cải thiện khả năng tìm kiếm thông tin pháp lý trực tuyến, hỗ trợ người dùng trong việc nắm bắt nhanh chóng và chính xác những thay đổi trong lĩnh vực pháp luật đang diễn ra không ngừng.

1.2 Mục tiêu và phạm vi, phương pháp

1.2.1 Mục tiêu và phạm vi nghiên cứu

Mục tiêu của đề tài này là nghiên cứu và phát triển Elasticsearch xây dựng hệ thống tìm kiếm để cải thiện khả năng tìm kiếm và phân tích văn bản, đông thời áp dụng các công nghệ mới. Nâng cao trải nghiệm người dùng, tăng cường khả năng phân tích và trích xuất thông tin.

1.2.2 Phương pháp nghiên cứu

- Hiểu rõ về Elasticsearch: Nắm vững kiến thức về Elasticsearch, bao gồm cách nó hoạt động, cấu trúc dữ liệu, và các tính năng quan trọng. Đánh giá khả năng mở rộng và hiệu suất của Elasticsearch để đảm bảo sự phù hợp với nhu cầu tìm kiếm tin tức pháp luật.

- Tìm hiểu về dữ liệu pháp luật: Nghiên cứu sâu rộng về cấu trúc dữ liệu và đặc điểm của tin tức pháp luật để hiểu rõ nhu cầu tìm kiếm của người dùng. Xác định các đặc điểm đặc biệt của thông tin pháp luật, bao gồm thuật ngữ chuyên ngành và quan hệ giữa các khái niệm.

- Thiết kế mô hình tìm kiếm hiệu quả: Phát triển một mô hình tìm kiếm sử dụng Elasticsearch được tối ưu hóa cho việc truy vấn thông tin pháp luật. Tích hợp các tính năng

như lọc thông tin, sắp xếp theo độ ưu tiên, và gợi ý tìm kiếm để cải thiện trải nghiệm người dùng.

- Kiểm thử và tối ưu hóa hiệu suất: Tiến hành kiểm thử thực nghiệm để đánh giá hiệu suất của hệ thống tìm kiếm trong môi trường thực tế. Tối ưu hóa cấu hình Elasticsearch để đảm bảo độ nhanh chóng và đáng tin cậy trong việc truy vấn tin tức pháp luật.

- Thiết kế giao diện: Phát triển giao diện người dùng thân thiện và dễ sử dụng, tương thích với nhu cầu của người dùng cuối. Đảm bảo tính tương tác cao và trải nghiệm người dùng tích cực.

1.3 Nội dung công việc

- Kiến thức cơ bản về cách Elasticsearch hoạt động, kiến trúc và các tính năng.
- Triển khai Elasticsearch để xây dựng hệ thống tìm kiếm văn bản, gồm việc tạo và quản lý các chỉ mục, tìm kiếm và lọc dữ liệu
- Nghiên cứu và áp dụng các phương pháp tối ưu hóa hiệu suất
- Đánh giá hiệu suất và độ chính xác của hệ thống và thực hiện kiểm thử
- Dự kiến kết quả: Hiểu rõ về kiến trúc và tính năng của Elasticsearch. Xây dựng hệ thống tìm kiếm văn bản ổn định và hiệu quả sử dụng Elasticsearch.

CHƯƠNG 2 – CÁC KIẾN THỨC CƠ BẢN

2.1 Tổng quan về bài nghiên cứu

Tin tức hiện nay bao gồm nhiều lĩnh vực và chủ đề khác nhau, từ tin tức chính trị, kinh tế, xã hội đến giải trím thể thao, khoa học, công nghệ cho nên người dùng có nhu cầu khám phá và tiếp cận thông tin từ nhiều nguồn khác nhau. Với lượng thông tin lớn và đa dạng người dùng có thể gặp khó gặp trong việc tìm kiếm thông tin cụ thể và lọc nội dung chính xác, vì vậy chung ta mới duyệt qua các trang web, tìm cách dễ dang cho người dùng như nâng cao khả năng tìm kiếm đặc biệt là xây dựng một chatbot để phụ vụ.

2.1.1 Tìm hiểu về Elastic Search

Elasticsearch nói đơn giản là một cơ sở dữ liệu dùng để lưu trữ, tìm kiếm và query.

Elasticsearch là một công cụ mã nguồn mở tìm kiếm toàn văn bản và có khả năng phân tích mở rộng. Nó cho phép người dùng lưu trữ, tìm kiếm và phân tích khối lượng lớn dữ liệu một cách nhanh chóng và gần thời gian thực. Nó thường được sử dụng như các công cụ, công nghệ cơ bản mà quyền hạn các ứng dụng có tính năng tìm kiếm phức tạp và yêu cầu.

Elasticsearch là một công cụ dựa trên phần mềm Lucene. Nó cung cấp một bộ máy tìm kiếm dạng phân tán, có đầy đủ công cụ với một giao diện web HTTP có hỗ trợ dữ liệu JSON. Elasticsearch được phát triển bằng Java và được phát hành dạng mã nguồn mở theo giấy phép Apache.

- Các tính năng chính của elasticsearch:
  - Phân tích thời gian
  - Phân phối lưu trữ tập tin theo thời gian thực và từng lĩnh vực được lập chỉ mục.
  - Tài liệu định hướng, tất cả các đối tượng là tất cả các tài liệu.
  - Có tính sẵn sàng cao, dễ dàng mở rộng, hỗ trợ cluster, phân mảnh và nhân rộng (Shards và bản sao).
  - Giao diện thân thiện, hỗ trợ cho JSON.
  - Tìm Kiếm Toàn Văn Bản: Elasticsearch cung cấp một công cụ tìm kiếm mạnh mẽ cho văn bản đầy đủ và linh hoạt, bao gồm cả tìm kiếm phần văn bản, sắp xếp và lọc kết quả.
  - Phân Tán và Mở Rộng: Khả năng mở rộng của Elasticsearch cho phép nó mở rộng dữ liệu và số lượng người dùng mà không làm giảm hiệu suất.
  - Tìm Kiếm Đồng Nhất (Consistency): Elasticsearch hỗ trợ các mô hình tìm kiếm

đồng nhất thông qua các cấu hình nhất quán (consistency levels).

- Strong Consistency (Đồng Nhất Mạnh): Điều này đảm bảo rằng mọi đọc và ghi sẽ thấy dữ liệu ở một trạng thái đồng nhất trên toàn bộ hệ thống. Tính đồng nhất mạnh là đảm bảo rằng mọi replica sẽ đồng bộ với nhau một cách ngay lập tức.
- Eventual Consistency (Đồng Nhất Cuối Cùng): Điều này cho phép một số độ trễ giữa khi một giá trị được ghi và khi nó trở nên đồng nhất trên toàn bộ hệ thống. Eventual consistency không đảm bảo ngay lập tức, nhưng nó hứa rằng cuối cùng, tất cả các bản sao sẽ đồng bộ.
- Causal Consistency (Đồng Nhất Nhân Quả): Điều này giữ cho các sự kiện xảy ra có liên quan với nhau sẽ được duy trì trong một thứ tự nhất định trên tất cả các node. Nó đảm bảo rằng nếu một sự kiện A gây ra sự kiện B, thì tất cả các node sẽ thấy B sau A.)
- Chức Năng Query DSL (Domain-Specific Language): Elasticsearch sử dụng một ngôn ngữ truy vấn có cú pháp đặc biệt, được gọi là Query DSL, cho phép người dùng xác định các truy vấn phức tạp.
- Xử Lý Nâng Cao Dữ Liệu: Có khả năng xử lý dữ liệu phức tạp, bao gồm cả các loại dữ liệu khác nhau như văn bản, số, địa lý và hình ảnh.
- Giao Diện RESTful API: Elasticsearch cung cấp một giao diện API RESTful, làm cho việc tương tác và tích hợp với các ứng dụng web và dịch vụ trở nên dễ dàng.

➢ Các thành phần của Elasticsearch:

- Index: là một tập hợp các document có điểm tương tự nhau. Có 2 loại:
  + Forward Index: Đánh index theo nội dung, page -> words.
  + Inverted Index: Đánh index theo keyword: word -> pages.
- Document: một JSON object với một số dữ liệu. Đây là đơn vị dữ liệu cơ bản trong Elasticsearch.

+ JSON (JavaScript Object Notation) là một kiểu định dangj dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được.

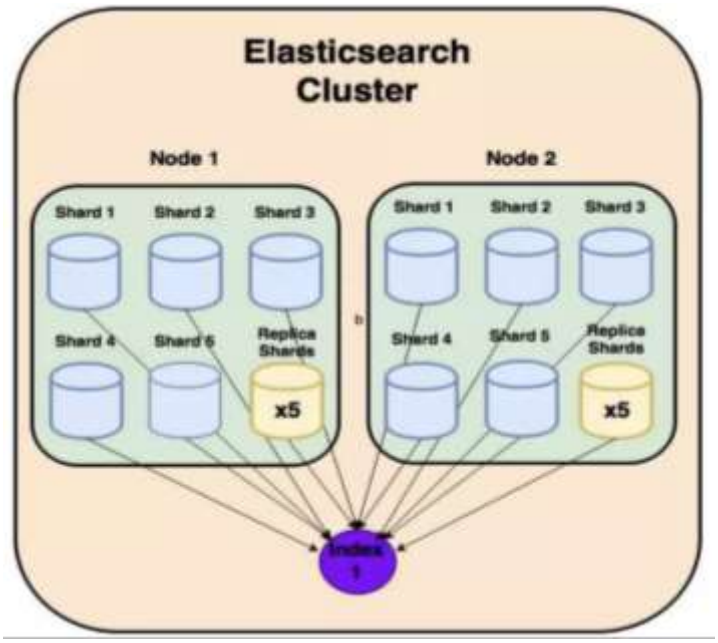
- Shard: Tập con các document của một index. Một index có thể có nhiều shard.

  Có 2 loại:

  + Primary Shard: Là nơi lưu trữ dữ liệu và đánh index. Sau khi đánh xong dữ liệu sẽ được vận chuyển tới các Replica Shard.

  + Replica Shard: Là nơi lưu trữ dữ liệu nhân bản của Primary Shard, bảo vệ duẽ liệu toàn vẹn khi có sự cố.

- Node: Nơi lưu trữ dữ liệu, tham gia vào việc đánh mục lục của cluster cũng như thực hiện việc tìm kiếm. Mỗi node được định danh bằng một unique name.
- Cluster: Tập hợp các node chứa tất cả các dữ liệu. Mỗi cluster được định danh bằng một unique name. Mỗi cluster có một node chính (master) được lựa chọn tự động và có thể thay thế khi gặp sự cố.
  - Cách lưu cấu trúc cơ bản của 1 cluster trong Elasticsearch:



Hình 1: Cấu trúc của 1 cluster trong Elasticsearch.

2.1.2 Natural Language Processing (NLP)

NLP là một lĩnh vực con của AI nó tập trung vào việc hiểu và xử lý ngôn ngữ tự nhiên của con người. NLP có khả năng xử lý nhiều ngôn ngữ khác nhau và có thể phân tích văn bản dựa trên cấu trúc ngữ pháp mà còn cần hiểu ngữ cảnh và nghĩa.

Sử dùng NLP để rút trích thông tin quan trọng như

2.1.3 Chatbot

Chatbot là một chương trình máy tính có thể mô phỏng cuộc trò chuyện hoặc trò chuyện với người dùng bằng ngôn ngữ tự nhiên thông qua ứng dụng nhắn tin, trang web hoặc ứng dụng di động và tương tác với người dùng theo thông tin đầu vào của họ và phải hoạt động 24/7.

Chatbots được phát triển và trở nên phổ biến nhờ việc sử dụng ngày càng nhiều các thiết bị thông minh và công nghệ IoT

- Các loại Chatbots

- Base-line chatbot là một chatbot dựa trên cơ sở dữ liệu và sử dụng logic if / then để tạo ra một luồng hội thoại và việc này mất rất nhiều thời gian để đảm bảo sự hiểu biết về câu hỏi và câu trả lời cần thiết
- AI chatbot phức tạp hơn so với dòng cơ bản nhưng nó có tính tương tác và cá nhân hóa cao hơn và cần được đào tạo về dữ liệu lớn để trở nên ấn tượng nếu vấn đề phù hợp với khả năng của chúng.
- Hybrid Model tiếp cận kết hợp chatbot Base-line và AI để làm cho nó trở nên thông minh hơn và hành vi của nó được mong đợi hơn bằng cách dựa vào cơ sở dữ liệu và thuật toán Ai để phối hợp với nhau.

Chatbot phân tích văn bản đã nhập và khớp văn bản với dữ liệu được xác định trước gọi là ý định được phân loại để quản lý cuộc trò chuyện. Lời nói của người dùng được gắn thẻ với một trong những ý định này, ngay cả khi những gì người dùng nói trải dài trên hai hoặc nhiều ý định. Hầu hết các chatbot sẽ có ý định đạt điểm cao nhất và đưa cuộc trò chuyện đi theo con đường đó.

Một số chatbot có thể tương tác và hiểu giọng nói của người dùng bằng cách sử dụng một bộ giao diện lập trình ứng dụng (api's) chuyển đổi giọng nói được ghi âm sang ngôn ngữ, sau đó chuyển đổi giọng nói đó thành các từ của ngôn ngữ đó rồi xử lý văn bản được chuyển đổi như đã đề cập ở trên.

- Giới thiệu về GPT 3.5 Turbo

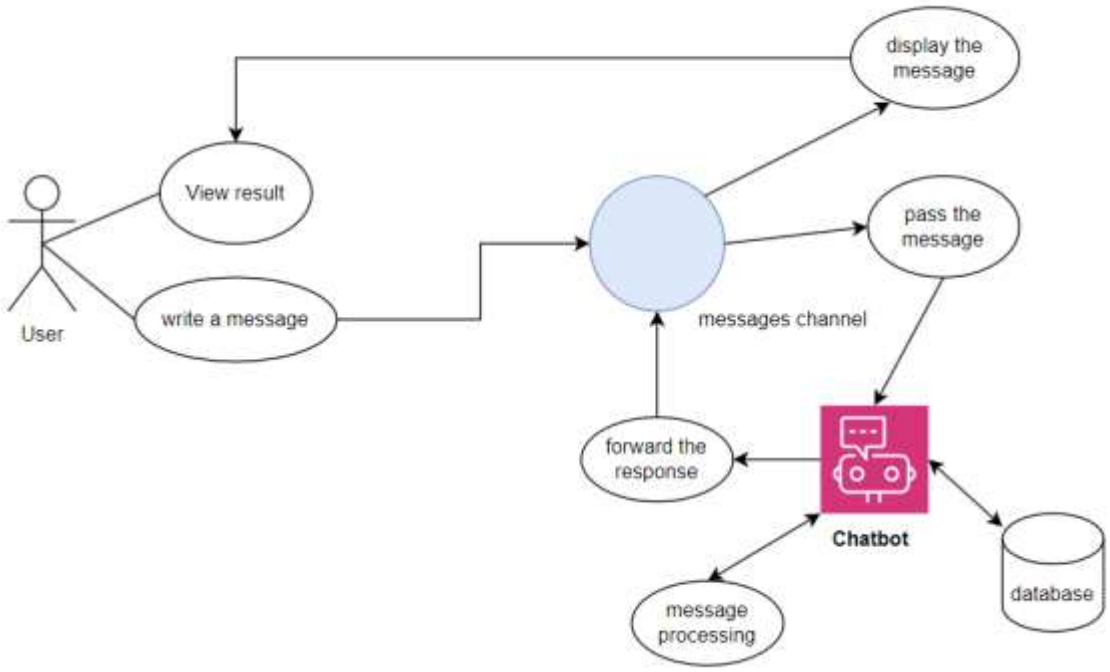
GPT 3.5 Turbo là phiên bản cải tiến của GPT-3, với mức giá rẻ hơn gấp 10 lần. Mô hình này có thể được sử dụng cho cả trò chuyện hội thoại và hoàn thiện văn bản giống như GPT-3. Trong phần tiếp theo, chúng ta sẽ khám phá cách xây dựng một ứng dụng chatbot sử dụng mô hình này cùng với Spell Kit và Verso Edge Functions.

- Xây dựng ứng dụng chatbot với GPT 3.5 Turbo
  - Đầu tiên, chúng ta sẽ sử dụng OpenAI Playground để làm quen với cách thức hoạt động của API mới này trước khi tích hợp vào các ứng dụng. Chúng ta có thể thấy trên màn hình là một phần nhập và phần hiển thị kết

quả. Phần nhập dùng để cung cấp thông tin về ngữ cảnh hoặc định danh cho mô hình. Phần kết quả hiển thị phản hồi từ AI dựa trên ngữ cảnh đã cung cấp.

- Tiếp theo, chúng ta sẽ xây dựng trang chủ và giao diện tin nhắn cho ứng dụng của chúng ta. Chúng ta sẽ sử dụng các thành phần UI của Tailwind CSS để tạo giao diện đơn giản và dễ sử dụng.
- Sau đó, chúng ta sẽ xử lý yêu cầu chat từ người dùng. Khi người dùng gửi tin nhắn, chúng ta sẽ lấy nội dung tin nhắn đó và gửi yêu cầu chat đến OpenAI API.
- Tiếp theo, chúng ta sẽ xử lý phản hồi từ OpenAI API để hiển thị kết quả trả về từ mô hình chatbot. Chúng ta sẽ theo dõi lịch sử chat để giữ ngữ cảnh và gửi toàn bộ lịch sử chat đó trong mỗi yêu cầu mới.
- Chúng ta cũng sẽ xử lý lỗi và hiển thị thông báo cho người dùng trong trường hợp có lỗi xảy ra. Điều này giúp cải thiện trải nghiệm người dùng và giúp họ hiểu rõ hơn về việc tương tác với chatbot.
- Cuối cùng, chúng ta sẽ triển khai ứng dụng chatbot lên Vercel, một nền tảng hosting phổ biến. Việc triển khai lên nền tảng này giúp ứng dụng của chúng ta hoạt động một cách nhanh chóng và ổn định.

- System của Chatbot



Giải thích mô hình:

- User: Người dùng có thể viết và gửi thông điệp và xem kết quả được hồi đáp từ hệ thống.
- Write a message: Hành động của người dùng khi họ tạo ra một thông điệp mới để gửi đi.
- View result: Người dùng sau khi gửi thông điệp sẽ chờ và xem kết quả được trả về từ hệ thống.
- messages channel: Đây là kênh thông điệp nơi mà thông điệp từ người dùng được truyền đi (pass the message) và sau đó hiển thị (display the

message). Kênh này có vai trò trung gian giữa người dùng và hệ thống xử lý thông điệp.

- Message processing: là phần của hệ thống nơi tin nhắn được xử lý. Có thể bao gồm việc phân tích, hiểu và chuẩn bị câu trả lời cho tin nhắn từ người dùng.
- Chatbot/Database: Cơ sở dữ liệu có thể được chatbot sử dụng để lấy thông tin cần thiết để trả lời người dùng. Chatbot ở đây là bộ phận sử dụng AI hoặc quy tắc đã được định trước để tạo ra phản hồi cho tin nhắn của người dùng.
- Forward the response: Sau khi tin nhắn đã được xử lý và câu trả lời đã sẵn sàng, câu trả lời này được chuyển tiếp lại thông qua kênh thông điệp để hiển thị cho người dùng.

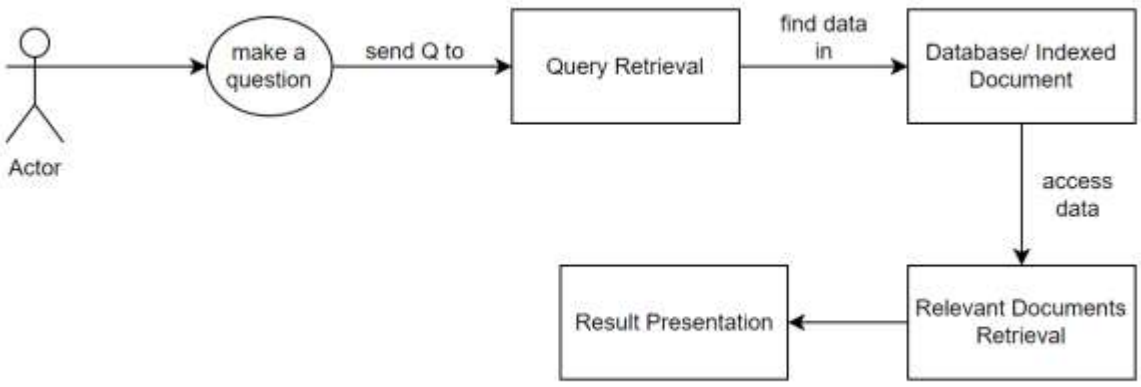
2.2 Mô Hình RAG (Retrieval-Augmented Generation)

Thế hệ tăng cường truy xuất (RAG) là một mô hình đổi mới trong lĩnh vực AI và xử lý ngôn ngữ tự nhiên. Nó đánh dấu một sự thay đổi đáng kể so với các mô hình ngôn ngữ thông thường bằng cách tích hợp việc truy xuất thông tin vào quá trình tạo ngôn ngữ. Cách tiếp cận kết hợp này nâng cao khả năng của các mô hình AI trong việc tạo ra các phản hồi không chỉ chính xác về mặt ngữ cảnh mà còn được truyền tải kiến thức cập nhật từ các nguồn dữ liệu bên ngoài.

- Truy xuất thông tin (Retrieval component): Tìm kiếm các tài liệu có liên quan đến đầu vào của người dùng trong một kho dữ liệu lớn. Thường sử dụng kỹ thuật nhúng vector (vector embedding) và cơ sở dữ liệu vector để truy vấn nhanh chóng và hiệu quả.
- Tạo sinh văn bản (Generation component): Dựa trên các tài liệu được truy xuất, mô hình sinh văn bản tạo ra đoạn văn bản mới phù hợp với yêu cầu của người

dùng. Các mô hình ngôn ngữ lớn (Large Language Models - LLM) như BERT, GPT-3 thường được sử dụng trong giai đoạn này.

- Flowchart mô hình RAG xử lý câu hỏi người dùng



Giải thích :

- Với "User Question", câu hỏi được đặt bởi người dùng.
- Câu hỏi được gửi đến phần "Query Retrieval" của mô hình RAG, nơi mà việc truy xuất các tài liệu từ cơ sở dữ liệu được thực hiện.
- "Query Retrieval" xử lý câu hỏi bằng cách tìm kiếm trong cơ sở dữ liệu hoặc tập dữ liệu được lập chỉ mục để tìm các đoạn văn bản hoặc tài liệu liên quan đến câu hỏi.
- Sau đó, các tài liệu liên quan nhất được truy xuất từ cơ sở dữ liệu hoặc tập dữ liệu đã được lập chỉ mục, được gọi là "Relevant Documents Retrieval".
- "Result Presentation": đưa ra kết quả cuối cùng

2.3 Mô hình

CHƯƠNG 3 PHƯƠNG PHÁP ĐỀ XUẤT

3.1 Overview

3.2 Mô hình

3.3 Mô hình xử lý dữ liệu

Chương 4 THỰC NGHIỆM

4.1 Xây dựng dữ liệu

4.1.1 Thu thập dữ liệu

- Giới Thiệu VIETNAMESE FAKE NEWS DATASET – VFND

VFND là bộ dataset về các tin tức giả bằng ngôn ngữ tiếng Việt được tập hợp trong khoảng thời gian từ 2017 đến 2019 (năm hoàn thành luận văn của các tác giả), các tin tức được đưa vào đây được phân loại thật giả dựa trên một số nguồn tin, tham chiếu chéo đến các nguồn tin được dẫn hoặc được phân loại bởi cộng đồng.

- ❖ Dữ liệu file CSV

Có 2 label để phân loại là: Fake (tin tức giả) giá trị 1 và Real (tin tức thật) giá trị 0. Các file và ý nghĩa têncủa chúng:

+ vn_news_226_tlfr.csv: Chứa 226 record dữ liệu bao gồm 2 trường Text và Label. Text tổng hợp từ các tin tức giả và thật từ Báo chí, tin tức báo chí bao gồm phần tiêu đề và nội dung.

+ vn_news_223_tdlfr.csv: Chứa 223 record dữ liệu các bài báo và domain name của các trang đã đăng các bài báo đó. tdlfr là [text, domain, label, fake,real]

- ❖ Dữ liệu file CSV và Json

Tập dữ liệu gồm: Tập dữ liệu được phân loại dựa trên 2 nhãn Fake và Real

trong đó với mỗi nhãn tương ứng với một thư muc, trong từng thư muc sẽ được phân loại thành 2 nhóm: Article_Contents: tập hợp các bài báo đã được phân

loại trong nhãn và Social_Contents: tập hợp các Facebook post đã được phân loại.

Cấu trúc của phần lưu trữ tin tức giả: {Tên file}: {Tiêu đề/Link} -

{Lý do phân loại}. Trong một số tin tức sẽ có thêm phầnrelevant_site"

VD : VFND_Ac_Fake_1 , VFDN_So_Fake_1

VD : VFND_Ac_Real_1, VFDN_So_Real_1

```
Fake_Real_Dataset
└──Fake
    └──Article_Contents
    └──Social_Contents
└──Real
    └──Article_Contents
    └──Social_Contents
```

- Dữ liệu ban đầu: Dữ liệu ban đầu là một bộ dữ liệu riêng lẻ, một bài báo là một file json, tương ứng sẽ có 127 bài báo fake (127 file json) và 131 bài báo real (131 file json).

- Xử lý lại dữ liệu: Gọp hết thành một file json và chuyển sang file csv.

Gọp các file json:

```
, 'VFND_Ac_Real_80.json', 'VFND_Ac_Real_81.json', 'VFND_Ac_Real_82.json', 'VFND_Ac_Real_83.json'
, 'VFND_Ac_Real_84.json', 'VFND_Ac_Real_85.json', 'VFND_Ac_Real_86.json', 'VFND_Ac_Real_87.json'
, 'VFND_Ac_Real_88.json', 'VFND_Ac_Real_89.json', 'VFND_Ac_Real_90.json', 'VFND_Ac_Real_91.json'
, 'VFND_Ac_Real_92.json', 'VFND_Ac_Real_93.json', 'VFND_Ac_Real_94.json', 'VFND_Ac_Real_95.json'
, 'VFND_Ac_Real_96.json', 'VFND_Ac_Real_97.json', 'VFND_Ac_Real_98.json', 'VFND_Ac_Real_99.json'
, 'VFND_Ac_Real_100.json', 'VFND_Ac_Real_101.json', 'VFND_Ac_Real_102.json', 'VFND_Ac_Real_103.json'
, 'VFND_Ac_Real_104.json', 'VFND_Ac_Real_105.json', 'VFND_Ac_Real_106.json', 'VFND_Ac_Real_107.json'
, 'VFND_Ac_Real_108.json', 'VFND_Ac_Real_109.json', 'VFND_Ac_Real_110.json', 'VFND_Ac_Real_111.json'
, 'VFND_Ac_Real_112.json', 'VFND_Ac_Real_113.json', 'VFND_Ac_Real_114.json', 'VFND_Ac_Real_115.json'
, 'VFND_Ac_Real_116.json', 'VFND_Ac_Real_117.json', 'VFND_Ac_Real_118.json', 'VFND_Ac_Real_119.json'
, 'VFND_Ac_Real_120.json', 'VFND_Ac_Real_121.json', 'VFND_Ac_Real_122.json', 'VFND_Ac_Real_123.json'
, 'VFND_Ac_Real_124.json', 'VFND_Ac_Real_125.json', 'VFND_Ac_Real_126.json', 'VFND_Ac_Real_127.json']

# Biến lưu trữ nội dung ghép
combined_data = []

# Đọc từng file JSON và thêm nội dung vào biến combined_data
for file_path in file_paths:
    with open(file_path, 'r', encoding='utf-8') as json_file:
        data = json.load(json_file)
        combined_data.append(data)

# Ghi nội dung ghép vào một file JSON mới
output_file_path = 'dataset.json'
with open(output_file_path, 'w', encoding='utf-8') as output_file:
    json.dump(combined_data, output_file, ensure_ascii=False, indent=4)
```

4.1.3 Tiền xử lý và làm sạch dữ liệu

- Đọc dữ liệu từ file Json:

```
In [1]: import json

        # Đường dẫn đến file JSON
        json_path = 'dataset_full.json'

        # Đọc dữ liệu từ file JSON
        with open(json_path, 'r', encoding='utf-8') as file:
            data_json = json.load(file)

        data_json[:2]
```

```
Out[1]: [{'authors': ['Lê Anh'],
          'date_download': '2018-11-28 14:44:51',
          'date_modify': None,
          'date_publish': '2018-05-06 18:27:52',
          'description': 'Ngày 6.5, Công an thị xã Phước Long, tỉnh Bình Phước đã tạm giữ bà Lê Thị Xuyến (46 tuổi, ngụ phường Long Phư
        ớc, thị xã Phước Long) để điều tra về việc bắt cóc trẻ em.',
          'filename': 'https%3A%2F%2Flaodong.vn%2Fphap-luat%2Fmot-phu-nu-bit-mat-bat-coc-chau-be-5-tuoi-bi-nguoi-dan-tom-gon-605391.ld
        o.json',
```

- Làm sạch dữ liệu: Loại bỏ các ký tự đặc biệt, số không quan trọng và các khoảng trắng thừa trong ‘text’ và ‘title’.
- Tiền xử lý văn bản: Chuyển đổi văn bản sang chữ thường, loại bỏ dấu câu, và loại bỏ khoảng trắng thừa.
- Dữ liệu đã được tiền xử lý thành công từ trường ‘text’ và ‘title’ sẽ được đưa vào các trường ‘text_processed’ và ‘title_processed’ chứa các văn bản đã được làm sạch và chuẩn hóa.

```
In [2]: import re

        # Hàm tiền xử lý văn bản
        def preprocess_text(text):
            text = str(text).lower()  # Chuyển văn bản thành chữ thường
            text = re.sub(r'\W', ' ', text)  # Loại bỏ các ký tự không phải từ
            text = re.sub(r'\s+', ' ', text)  # Loại bỏ khoảng trắng thừa
            return text.strip()

        # Áp dụng hàm tiền xử lý cho 'text' và 'title' trong mỗi bản ghi
        for item in data_json:
            item['text_processed'] = preprocess_text(item.get('text', ''))
            item['title_processed'] = preprocess_text(item.get('title', ''))

        # Kiểm tra lại dữ liệu sau khi đã tiền xử lý
        data_json[:2]
```

```
Out[2]: [{'authors': ['Lê Anh'],
          'date_download': '2018-11-28 14:44:51',
          'date_modify': None,
```

4.2 Mô hình thực nghiệm

4.3 Các tiêu chí, độ do đánh giá mô hình

4.4 Kết quả thực nghiệm

4.5 Demo hệ thống

Chương 5 KẾT LUẬN

TÀI LIỆU THAM KHẢO

[1]. “Chatbots - Artificial Solutions.” Chatbots | Conversational AI Platform for Enterprise – Teneo. Accessed 8 Nov. 2020

[2] “How Do Bots and Chatbots Work.”

https://www.cxtoday.com/contact-centre/how-do-bots-and-chatbots-work/

[3] Installation Elasticsearch: https://www.elastic.co/downloads/elasticsearch

[4] Installation Kibana: https://www.elastic.co/downloads/kibana

[5] “Get API Key from openAI chatgpt”, “GPT 3.5 Turbo”

https://openai.com/chatgpt

[6] “ChatGPT and Elasticsearch: OpenAI meets private data”

https://www.elastic.co/search-labs