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

BÁO CÁO GIỮA 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 GIỮA 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 2014

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 – MỞ ĐẦU

1.1 Mô tả đề tài

Đề tài dự án này là tìm hiểu và ứng dụng nền tảng Elasticsearch cho tìm kiếm văn bản, tức là chúng em sẽ sử dụng công cụ Elasticsearch để phúc vụ cho website tin tức về pháp luật giúp xử lý cho việc tìm kiếm càng dễ dàng hơn.

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.

Phương pháp nghiên cứu

- Tìm hiểu và nghiên cứu về Elasticsearch
- Xây dựng một hệ thống sử dụng Elastic Search để thử nghiệm khả năng tìm kiếm và xử lý dữ liệu.
- Phát triển giải thuật tìm kiếm.

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.

1.2 Giới thiệu bài toán

Bài toán Elasticsearch đặt ra ý nghĩa quan trọng trong lĩnh vực tìm kiếm và quản lý dữ liệu, đặc biệt là trong các hệ thống thư viện, website thông tin, hay các ứng dụng web khác.

- Tìm Kiếm Hiệu Quả: Elasticsearch giúp cải thiện hiệu suất tìm kiếm văn bản, cho phép người dùng nhanh chóng và chính xác tìm thấy thông tin mong muốn trên các trang web, hệ thống thư viện, hoặc các ứng dụng.
- Quản Lý Thư Viện và Tài Nguyên: Trong ngữ cảnh của thư viện và các trang web tài nguyên, Elasticsearch giúp tổ chức và quản lý các tài liệu, sách, bài viết, và thông tin khác một cách hiệu quả.
- Tìm Kiếm Tích Hợp: Bài toán Elasticsearch giúp tích hợp khả năng tìm kiếm mạnh mẽ vào các ứng dụng và trang web, tạo ra trải nghiệm người dùng tốt hơn.
- Phân Loại và Lọc Dữ Liệu: Elasticsearch không chỉ giúp tìm kiếm mà còn hỗ trợ phân loại và lọc dữ liệu theo nhiều tiêu chí khác nhau, từ tác giả đến thể loại hay ngày xuất bản.
- Đồng Bộ Dữ Liệu Phân Tán: Trong môi trường phân tán, Elasticsearch giúp đồng bộ dữ liệu trên nhiều node và replica, đảm bảo tính nhất quán và đồng nhất trong hệ thống.
- Tối Ưu Hóa Hiệu Suất Hệ Thống: Elasticsearch cung cấp cơ chế tối ưu hóa hiệu suất để xử lý lượng dữ liệu lớn mà vẫn duy trì khả năng tìm kiếm nhanh chóng.
- Hỗ Trợ Nghiên Cứu và Học Thuật: Các dự án và nghiên cứu liên quan đến Elasticsearch đóng góp vào lĩnh vực nghiên cứu về hệ thống thông tin, tìm kiếm thông tin, và quản lý dữ liệu.

- Tiện Ích Cho Cộng Đồng Người Dùng: Trong môi trường thư viện và website thông tin, Elasticsearch tạo ra một tiện ích lớn cho người đọc và người sử dụng thông tin, giúp họ dễ dàng tìm kiếm và tiếp cận tài liệu.
- Nó thuộc một bà toán hiện đại bởi vì nó là ứng dụng công nghệ hiện đại và mạnh mẽ nhất trong lĩnh vực tìm kiếm và xử lý dữ liệu lớn.
- Về phân tán và linh hoạt nó là một thách thức phức tạp về việc phân tán và quản lý dữ liệu văn bản trên mạng, đồng thời đảm bảo khả năng tìm kiếm nhanh chóng và hiệu quả. Elasticsearch, với kiến trúc phân tán và linh hoạt, là lựa chọn phù hợp.
- Có thể coi đây là một bài toán kết hợp giữa cổ điển và hiện đại. Bài toán tối ưu hóa tìm kiếm thông tin trong các nguồn dữ liệu đã tồn tại từ trước (ví dụ: thư viện sách) là một vấn đề cổ điển. Tuy nhiên, cách tiếp cận và sử dụng công nghệ Elasticsearch để giải quyết vấn đề là một chiều mới và hiện đại của bài toán.

❖ Tính Năng của Elasticsearch:

+ 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).

+ (trong 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.

1.3 Giới thiệu nghiên cứu liên quan

1.3.1 Tìm hiểu tiếp cận tổng quan giải quyết bài toán

- 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.
- Mô hình Retrieval-Augmented Generation (RAG) là một bước tiến mới trong lĩnh vực xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP), đặc biệt là trong khía cạnh tạo sinh văn bản (text generation).

- Truy xuất thông tin (Retrieval): 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): 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.
- Quá trình hoạt động của RAG với dạng cơ sở dữ liệu vector
  + Lưu trữ dữ liệu : Khi tài liệu được nhập vào, một mô hình nhúng sẽ biến văn bản thành vectơ nhiều chiều. Mỗi vectơ thể hiện ý nghĩa ngữ nghĩa của tài liệu dưới dạng số. Các vectơ này sau đó được lưu trữ trong cơ sở dữ liệu vectơ.
  + Lập chỉ mục : Để tạo điều kiện truy xuất nhanh, cơ sở dữ liệu xây dựng chỉ mục trên các vectơ này bằng các thuật toán phù hợp với không gian nhiều chiều, chẳng hạn như Chỉ mục tệp đảo ngược (IVF) hoặc Thế giới nhỏ điều hướng phân cấp (HNSW). Việc lựa chọn loại chỉ mục sẽ cân bằng giữa tốc độ và độ chính xác của tìm kiếm.
  + Tìm kiếm tương tự : Khi một truy vấn được thực hiện, nó cũng được chuyển đổi thành một vectơ bằng cách sử dụng cùng một mô hình nhúng. Cơ sở dữ liệu vectơ sau đó sử dụng chỉ mục để nhanh chóng tìm thấy các vectơ giống nhất với vectơ truy vấn. Độ tương tự được xác định bằng các số liệu khoảng cách như khoảng cách Euclide hoặc độ tương tự cosin.

1.3.2 Tìm hiểu các nghiên cứu khác giải quyết bài toán

- GitHub:

+ Sử dụng Từ Khóa Phù Hợp: Sử dụng các từ khóa như "Elasticsearch library," "Elasticsearch example," hoặc "Elasticsearch integration" để tìm kiếm dự án có liên quan.
+ Lọc Kết Quả Tìm Kiếm: Sử dụng các tùy chọn lọc trên GitHub để hạn chế kết quả theo ngôn ngữ lập trình, số sao, và ngày cập nhật.
+ Kiểm Tra README và Documentation: Đọc README và tài liệu của dự án để đảm bảo rằng nó đáp ứng yêu cầu và mục tiêu của bạn.
+ Kiểm Tra Thảo Luận và Issues: Kiểm tra thảo luận và vấn đề mở để đánh giá sự ổn định và hỗ trợ của dự án.

– Hugging Face:
+ Sử Dụng Hugging Face Search Bar: Sử dụng thanh tìm kiếm trên trang chủ của Hugging Face để tìm kiếm các mô hình hoặc ứng dụng liên quan đến Elasticsearch.
+ Xem Kho Mô Hình (Model Hub): Kiểm tra trong "Model Hub" để xem nếu có mô hình nào đã được huấn luyện hoặc tích hợp với Elasticsearch.
+ Đọc Blogs và Hướng Dẫn: Đọc các bài viết trong blog của Hugging Face để tìm hiểu về các kịch bản sử dụng Elasticsearch trong NLP và machine learning.

– Medium:
+ Sử Dụng Từ Khóa Tìm Kiếm: Sử dụng từ khóa như "Elasticsearch integration," "text search," hoặc "Elasticsearch use case" để tìm bài viết có liên quan.
+ Sử Dụng Chuyên Mục và Tags: Tìm kiếm trong các chuyên mục hoặc sử dụng các tags để chính xác hóa kết quả.
+ Đánh Giá Bài Viết: Kiểm tra xem bài viết có đánh giá tích cực từ cộng đồng không và có được cập nhật gần đây hay không.

+ Thảo Luận và Phản Hồi: Xem các phản hồi và thảo luận dưới bài viết để có cái nhìn đánh giá từ độc giả và chia sẻ thông tin bổ ích.

CHƯƠNG 2 – Giới Thiệu Tổng Quan

2.1 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): 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): 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.

2.2 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:

1. 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 Facebook và Báo chí, tin

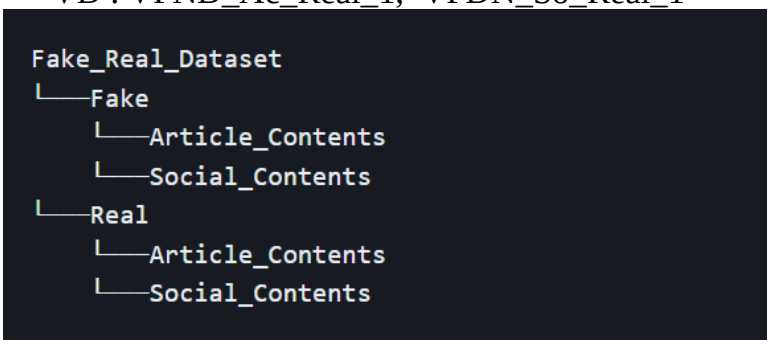
tức báo chí sẽ bao gồm phần tiêu đề và nội dung. tlfr là [text, label, fake, real]

2. 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 phần relevant_site"
- VD : VFND_Ac_Fake_1 , VFDN_So_Fake_1
- VD : VFND_Ac_Real_1, VFDN_So_Real_1



- 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_76.json', 'VFND_Ac_Real_77.json', 'VFND_Ac_Real_78.json', 'VFND_Ac_Real_79.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)
```

- Chuyển sang pdf:

In [25]:
```
json_file = 'dataset.json'
csv_file = 'dataset_full.csv'

# Đọc nội dung từ file JSON
with open(json_file_path, 'r', encoding='utf-8') as json_file:
    data = json.load(json_file)

# Chọn trường các trường
desired_fields = [
    "authors","date_download","date_modify","date_publish","description","filename","image_url","language","localpath","source_dc
]

# Sắp xếp mỗi phần tử trong danh sách JSON theo các trường đã chọn
sorted_data = [{field: item[field] for field in desired_fields} for item in data]

# Chuyển đổi dữ liệu JSON đã sắp xếp thành DataFrame của pandas
df = pd.json_normalize(sorted_data)

# Ghi DataFrame vào file CSV
df.to_csv(csv_file_path, index=False, encoding='utf-8-sig')
```

2.3 Truy vấn dữ liệu trên Elasticsearch

- **API Insert, thêm dữ liệu vào Elasticsearch**

```
PUT /ac

POST /ac/fake/0
{
  "authors": [],
  "date_download": "2022-05-30 01:52:10",
  "date_modify": null,
  "date_publish": "2018-05-24 00:00:00",
  "description": "Tình phụ tử: Cách yêu thương của cha giành cho con gái luôn vô
    cùng đặc biệt và ấm áp.",
  "filename": "https%3A%2F%2F2sao.vn%2Float-anh-sieu-yeu-chung-minh-con-gai-la-nguoi
    -tinh-kiep-truoc-cua-cha-i-154763.html.json",
  "image_url": "https://2sao.vietnamnetjsc.vn/images/2018/05/23/21/26/con-la-nguoi
    -tinh-kiep-truoc-cua-bo-oki.jpg",
  "language": "vi",
  "localpath": null,
  "maintext": "Thế nhưng, dù không nói ra thì cha vĩnh viễn là người đàn ông bao
    dung, yêu thương và mang đến cho bạn những điều tốt đẹp nhất.",
  "source_domain": "2sao.vn",
  "title": "Con gái chính là người tình kiếp trước của cha \n Cách yêu thương của
    cha dành cho con gái thật vô cùng đặc biệt và ấm áp.\n\nCâu nói \"con gái là
    người tình kiếp trước của cha\" để chỉ mối quan hệ thân thiết, gắn bó giữa cha
```

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "ac"
}
```

- **API Hiển thị thông tin dữ liệu vn_news_223_tdflr trên Kibana**

```
GET /vn_news_223_tdlfr/_search
{
  "query": {
  "match_all": {}
  }
}
```

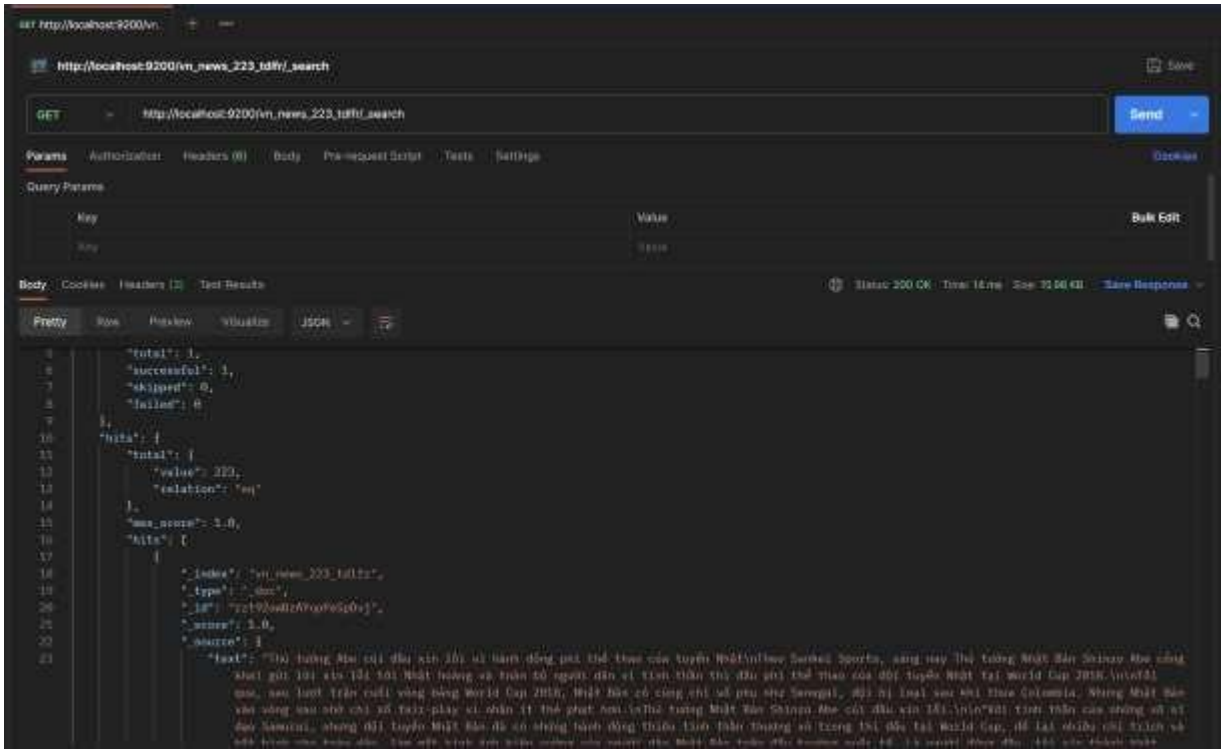
- Kết quả trên Kibana

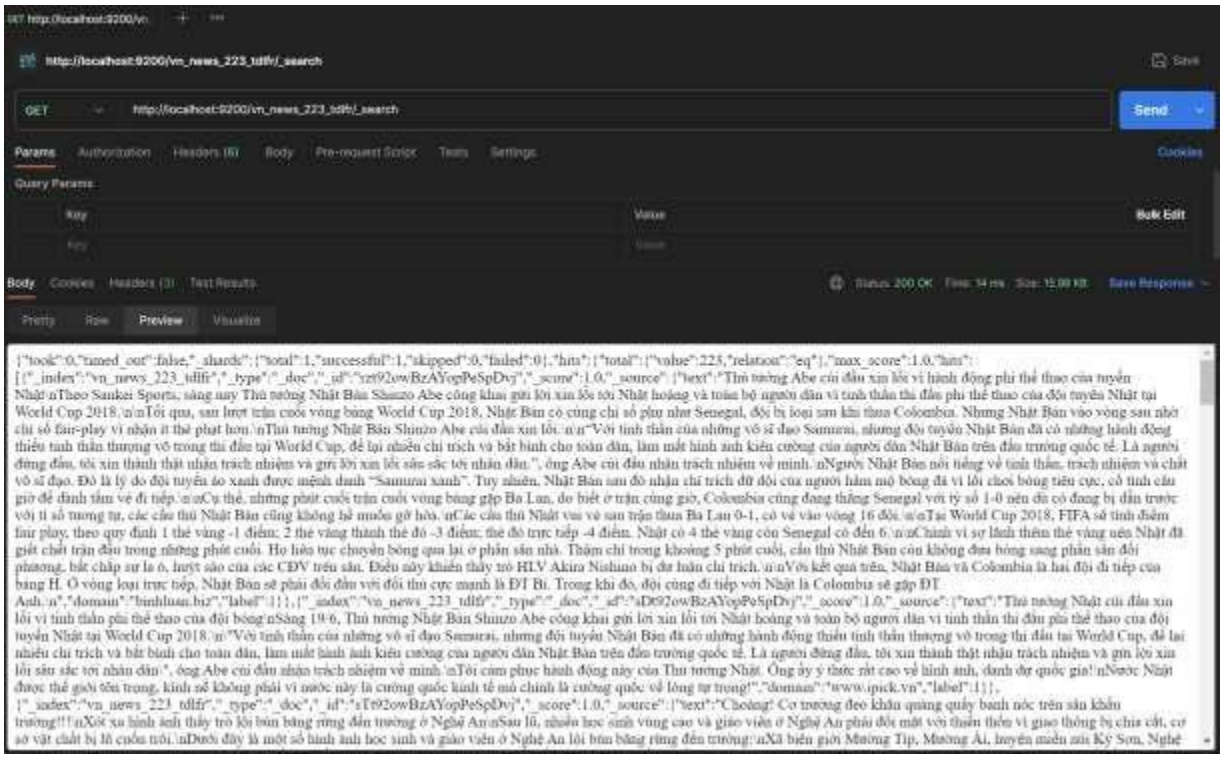
```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 223,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "vn_news_223_tdlfr",
        "_type" : "_doc",
        "_id" : "rat92owBzAYopPeSpDvj",
        "_score" : 1.0,
        "_source" : {
          "text" : """Thủ tướng Abe cúi đầu xin lỗi vì hành động phi thể thao của tuyển Nhật
Theo Sankei Sports, sáng nay Thủ tướng Nhật Bản Shinzo Abe công khai gửi lời xin lỗi tới Nhật hoàng và toàn bộ người dân vì tinh thần thi đấu phi thể thao của đội tuyển Nhật tại World Cup 2018.

Tối qua, sau lượt trận cuối vòng bảng World Cup 2018, Nhật Bản có cùng chỉ số phụ như Senegal, đội bị loại sau khi thua Colombia. Nhưng Nhật Bản vào vòng sau nhờ chỉ số fair-play vì nhận ít thẻ phạt hơn.
Thủ tướng Nhật Bản Shinzo Abe cúi đầu xin lỗi:

"Với tinh thần của những võ sĩ đạo Samurai, nhưng đội tuyển Nhật Bản đã có những hành động thiếu tinh thần thượng võ trong thi đấu tại World Cup, để lại nhiều chỉ trích và bất bình cho toàn dân, làm mất hình ảnh kiên cường của người dân Nhật Bản trên đấu trường quốc tế. Là người đứng đầu, tôi xin thành thật nhận trách nhiệm và gửi lời xin lỗi sâu sắc tới nhân dân.", ông Abe cúi đầu nhận trách nhiệm về mình.
Người Nhật Bản nổi tiếng về tinh thần, trách nhiệm và chất võ sĩ đạo. Đó là lý do đội tuyển áo xanh được mệnh danh "Samurai xanh". Tuy nhiên, Nhật Bản sau đó nhận chỉ trích dữ dội của người hâm mộ bóng đá vì lối chơi bóng tiêu cực, cố tình câu giờ để dành tấm vé đi tiếp.
```


- Kết quả trên Elasticsearch http://localhost:9200/vn_news_223_tdlfr/_search

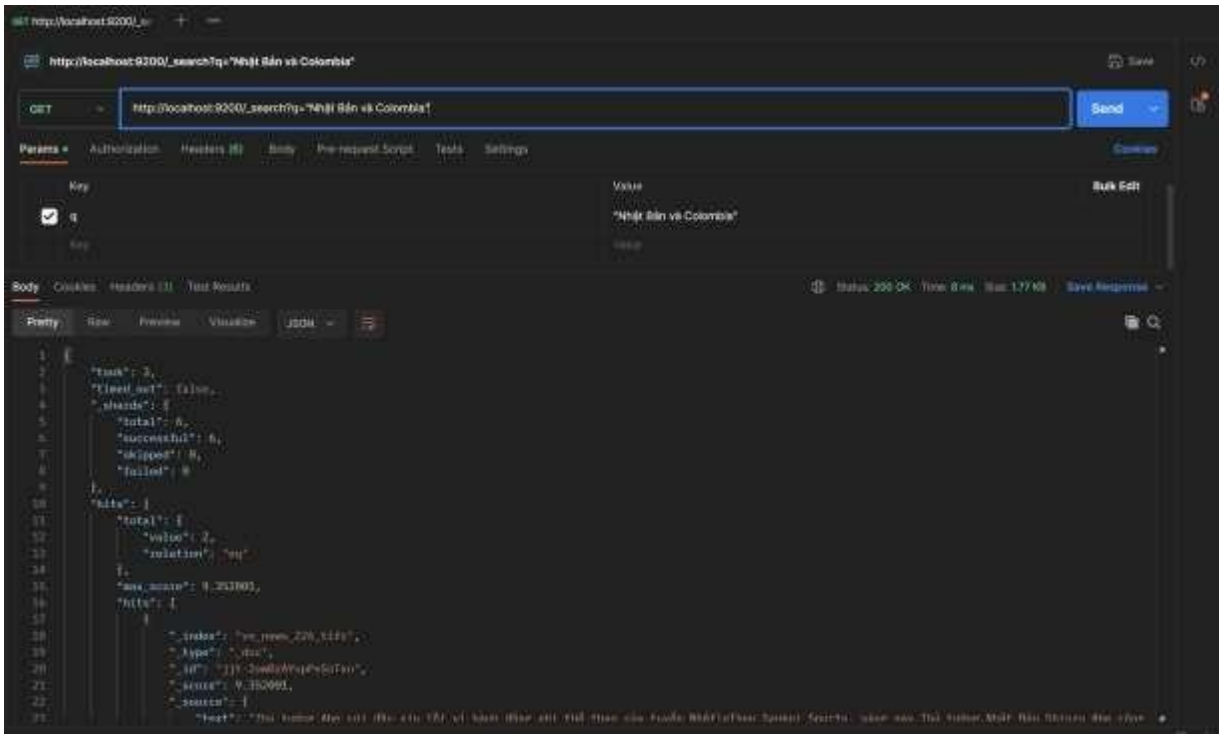




API đọc dữ liệu – Tìm kiếm dữ liệu, document Elasticsearch

Ví dụ tìm kiến tất cả các document có chứ từ " Nhật Bản và Colombia"

```
GET /_search?q="Nhật Bản và Colombia"
```


http://localhost:9200/_search?q="Nhật Bản và Colombia"
GET
http://localhost:9200/_search?q="Nhật Bản và Colombia"
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Key
Value
Bulk Edit
q
"Nhật Bản và Colombia"
Body
Cookies
Headers (11)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"took": 3,
"timed_out": false,
"_shards": {
"total": 6,
"successful": 6,
"skipped": 0,
"failed": 0
},
"hits": {
"total": {
"value": 2,
"relation": "eq"
},
"max_score": 9.253903,
"hits": [

TÀI LIỆU THAM KHẢO

Tiếng Việt

Tiếng Anh