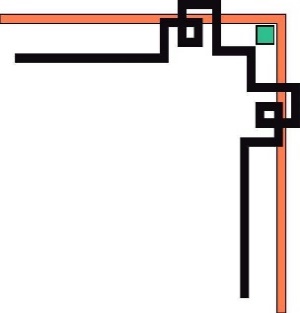
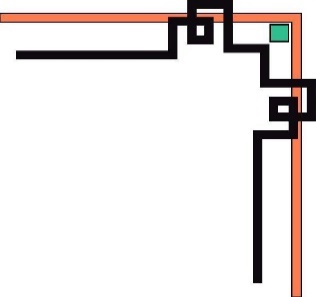
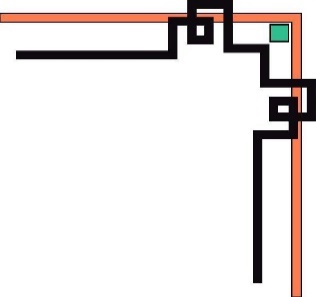
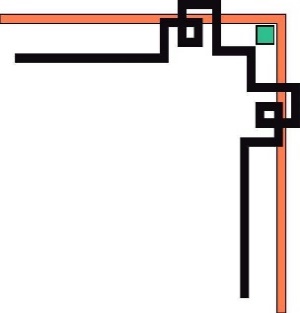
**ĐẠI HỌC HUẾ**



**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

🙠🙟🕮🙝🙢

**Icon

Description automatically generated**

**BÁO CÁO**

**ĐỒ ÁN**

**Học kỳ I, năm học 2023 – 2024**

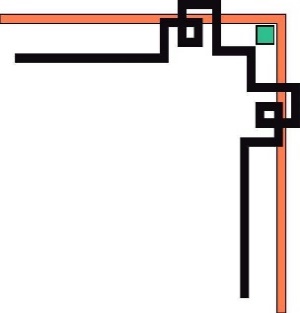
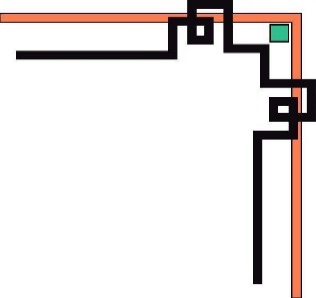
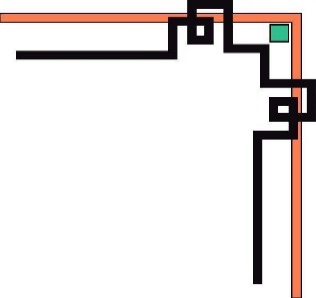
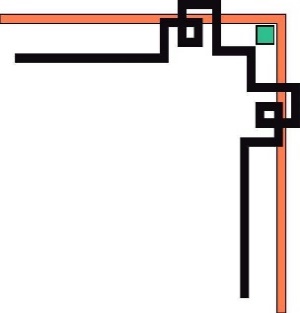
**HỌC PHẦN: KHAI PHÁ DỮ LIỆU LỚN**

**Đề tài: Xây dựng hệ thống gợi ý sản phẩm cho sàn thương mại điện tử Ebay**

|  |
| --- |
| **Số phách**  *(Do hội đồng chấm thi ghi)* |

**Thừa Thiên Huế, ngày….tháng….năm 2024**

**ĐẠI HỌC HUẾ**



**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

🙠🙟🕮🙝🙢

**Icon

Description automatically generated**

**(Bìa phụ 2)**

**BÁO CÁO**

**ĐỒ ÁN**

**Học kỳ I, năm học 2023 – 2024**

**HỌC PHẦN: KHAI PHÁ DỮ LIỆU LỚN**

**Đề tài: Xây dựng hệ thống gợi ý sản phẩm cho sàn thương mại điện tử Ebay**

**Giảng viên hướng dẫn: Hồ Quốc Dũng**

**Lớp: Khoa học dữ liệu & Trí tuệ nhân tạo – Khóa 1**

**Sinh viên thực hiện: Phan Thế Minh Châu**

*(ký và ghi rõ họ tên)*

|  |
| --- |
| **Số phách**  *(Do hội đồng chấm thi ghi)* |

**Thừa Thiên Huế, ngày….tháng….năm 2024**

**ĐẠI HỌC HUẾ**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

**PHIẾU ĐÁNH GIÁ ĐỒ ÁN/TIỂU LUẬN/BÀI TẬP LỚN**

**Học kỳ I, năm học 2023 - 2024**

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1** | **Cán bộ chấm thi 2** |
| **Nhận xét:**  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  **Điểm đánh giá của CBCT1:**  Bằng số: .........................................  Bằng chữ: ....................................... | **Nhận xét:**  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  ..............................................................  **Điểm đánh giá của CBCT2:**  Bằng số: .........................................  Bằng chữ: ....................................... |

Điểm kết luận:

Bằng số:

Bằng chữ:

*Thừa Thiên Huế, ngày…tháng…năm 2024*

**Cán bộ chấm thi 1 Cán bộ chấm thi 2**

*(Ký và ghi rõ họ và tên) (Ký và ghi rõ họ và tên)*

MỤC LỤC

[DANH MỤC HÌNH VẼ I](#_Toc156343453)

[CHƯƠNG I: PHÁT BIỂU BÀI TOÁN 2](#_Toc156343454)

[1. Giới thiệu 2](#_Toc156343455)

[1.1 Ngữ cảnh và lý do nghiên cứu 2](#_Toc156343456)

[2. Mục tiêu nghiên cứu 2](#_Toc156343457)

[3. Các lợi ích của đề tài 2](#_Toc156343458)

[4. Phương pháp nghiên cứu 3](#_Toc156343459)

[5. Kết quả nghiên cứu dự kiến 3](#_Toc156343460)

[CHƯƠNG II: CÁC CÔNG NGHỆ SỬ DỤNG 4](#_Toc156343461)

[1. Ngôn ngữ lập trình Python 4](#_Toc156343462)

[2. Beautifullsoup - Crawler 5](#_Toc156343463)

[3. Mongodb/ Mongodb  atlas - Lưu trữ 5](#_Toc156343464)

[4. Flask API 6](#_Toc156343465)

[5. Ngrok 6](#_Toc156343466)

[CHƯƠNG III: THU THẬP DỮ LIỆU 6](#_Toc156343467)

[1. Mô tả dữ liệu 6](#_Toc156343468)

[2. Thu thập dữ liệu 7](#_Toc156343469)

[2.1 Dữ liệu về thông tin sản phẩm 7](#_Toc156343470)

[2.2 Dữ liệu về hình ảnh 9](#_Toc156343471)

[CHƯƠNG IV: ĐỀ XUẤT GIẢI PHÁP 10](#_Toc156343472)

[1. Giới thiệu mô hình BERT 10](#_Toc156343473)

[2. Biểu diễn đầu vào của BERT 11](#_Toc156343474)

[3. Cải thiện BERT 12](#_Toc156343475)

[4. Kiến trúc của BERT 12](#_Toc156343476)

[CHƯƠNG V: TRIỂN KHAI ỨNG DỤNG 15](#_Toc156343477)

[1. Train model 16](#_Toc156343478)

[2. Sử dụng model 17](#_Toc156343479)

[3. Triển khai ứng dụng model lên web 18](#_Toc156343480)

[4. Kết quả 20](#_Toc156343481)

[CHƯƠNG VI. KẾT LUẬN 22](#_Toc156343482)

[TÀI LIỆU THAM KHẢO 23](#_Toc156343483)

# DANH MỤC HÌNH VẼ

[Hình 1. Các brand cần thu thập 7](#_Toc156344533)

[Hình 2. Kết quả thu thập link sản phẩm 8](#_Toc156344534)

[Hình 3. Kết quả thu thập thông tin sản phẩm 9](#_Toc156344535)

[Hình 4. Ảnh thu thập được lưu trên Google Drive 9](#_Toc156344536)

[Hình 5. Kiến Trúc mô hình BERT 10](#_Toc156344537)

[Hình 6. Mô hình đại diện đầu vào của BERT 11](#_Toc156344538)

[Hình 7. Quy trình tổng thể pre-training và fine-tuning của BERT 12](#_Toc156344539)

[Hình 8. Kiến trúc của Transformers 13](#_Toc156344540)

[Hình 9. Kiến trúc của Kiến trúc của một block transformer 13](#_Toc156344541)

[Hình 10. Mô hình kiến trúc Self-Attention 14](#_Toc156344542)

[Hình 11. Mô hình tính một vector Attention 14](#_Toc156344543)

[Hình 12. Sơ đồ tổng thể về ứng dụng 15](#_Toc156344544)

[Hình 13. Các hình ảnh được lưu trên Google Drive 17](#_Toc156344545)

[Hình 14. Giao diện hệ thống gợi ý sản phẩm 20](#_Toc156344546)

[Hình 15. Kết quả trả về từ web 21](#_Toc156344547)

# CHƯƠNG I: PHÁT BIỂU BÀI TOÁN

## Giới thiệu

### 1.1 Ngữ cảnh và lý do nghiên cứu

Trong những năm gần đây, sự phát triển của thương mại điện tử (E-Commerce) đã đem lại nhiều lợi ích to lớn cho nền kinh tế toàn cầu. Thông qua thương mại điện tử, nhiều loại hình kinh doanh mới được hình thành, trong đó có mua bán hàng qua mạng. Với hình thức mới này, người tiêu dùng có thể tiếp cận với hàng hóa một cách dễ. dàng và nhanh chóng hơn rất nhiều so với phương thức mua bán truyền thống.

Hiện nay, các hệ thống bán hàng trực tuyến đã tạo nhiều điều kiện thuận lợi để người mua có thể tiếp cận nhiều mặt hàng cùng lúc. Tuy nhiên, việc trình bày và trang trí quá nhiều các mặt hàng trên trang web đã gây ra không ít khó khăn cho người mua. Họ khó có thể chọn ra cho mình một sản phẩm ưng ý nhất.

Để khách hàng có thể đến và mua được một sản phẩm ưng ý thì một lời tư vấn, một sự trợ giúp là rất quan trọng. Trong phương thức bán hàng truyền thống những lời tư vấn như thế từ một người bán hàng sẽ tạo ra một lợi thế rất lớn cho cửa hàng. Do đó, để phương thức bán hàng qua mạng thực sự phát triển thì bên cạnh các lợi thế vốn có của mình việc có thêm một “người trợ giúp” là hết sức cần thiết.

Một hệ thống gợi ý tốt có thể đóng vai trò như một người trung gian hỗ trợ khách hàng đưa ra các quyết định mua hàng đúng đắn. Bằng cách xác định mục đích và nhu cầu của khách hàng, hệ thống có thể đưa ra một tập các gợi ý giúp cho người mua dễ dàng chọn lựa sản phẩm yêu thích hơn. Qua đó hiệu suất của việc mua bán hàng trực tuyến được tăng cao một cách đáng kể.

## 2. Mục tiêu nghiên cứu

Mục tiêu nghiên cứu của đề tài là xây dựng hệ thống gợi ý sản phẩm cho sàn thương mại điện tử dự trên độ tương đồng của chi tiết sản phẩm. Hệ thống sẽ sử dụng các thuộc tính của sản phẩm, chẳng hạn như tên sản phẩm, mô tả sản phẩm, thông số kỹ thuật,,... để tính toán độ tương đồng giữa các sản phẩm. Từ đó, hệ thống sẽ gợi ý cho người dùng những sản phẩm có độ tương đồng cao với mô tả của người dùng đưa ra.

## 3. Các lợi ích của đề tài

Việc xây dựng hệ thống gợi ý sản phẩm dựa trên độ tương đồng của chi tiết sản phẩm mang lại một số lợi ích cho sàn thương mại điện tử, bao gồm:

* Tăng trải nghiệm người dùng: Hệ thống gợi ý sản phẩm giúp người dùng dễ dàng tìm thấy những sản phẩm mà họ quan tâm hoặc có thể thích. Điều này giúp nâng cao trải nghiệm mua sắm của người dùng trên sàn thương mại điện tử.
* Tăng doanh số bán hàng: Hệ thống gợi ý sản phẩm giúp sàn thương mại điện tử tiếp cận được nhiều khách hàng tiềm năng hơn. Khi người dùng được gợi ý những sản phẩm phù hợp với nhu cầu của họ, họ sẽ có nhiều khả năng mua hàng hơn.
* Giảm chi phí marketing: Hệ thống gợi ý sản phẩm giúp sàn thương mại điện tử giảm chi phí marketing. Khi sàn thương mại điện tử có thể tiếp cận được nhiều khách hàng tiềm năng hơn, họ sẽ cần ít nỗ lực marketing hơn để bán được hàng.

## 4. Phương pháp nghiên cứu

Nghiên cứu lý thuyết: Nghiên cứu các tài liệu, bài báo khoa học liên quan đến hệ thống gợi ý sản phẩm.

Nghiên cứu thực nghiệm: Thu thập dữ liệu về sản phẩm từ sàn thương mại điện tử và sử dụng các phương pháp thống kê để tính toán độ tương đồng giữa các sản phẩm.

Phát triển hệ thống gợi ý sản phẩm: Xây dựng hệ thống gợi ý sản phẩm dựa trên các kết quả nghiên cứu lý thuyết và thực nghiệm.

## 5. Kết quả nghiên cứu dự kiến

Kết quả nghiên cứu dự kiến của đề tài bao gồm:

* Mô hình tính toán độ tương đồng giữa các sản phẩm: Mô hình sẽ sử dụng các thuộc tính của sản phẩm để tính toán độ tương đồng giữa các sản phẩm.
* Hệ thống gợi ý sản phẩm: Hệ thống sẽ sử dụng mô hình tính toán độ tương đồng để gợi ý cho người dùng những sản phẩm có độ tương đồng cao với các sản phẩm mà họ đã quan tâm hoặc mua trong quá khứ.

Ứng dụng của kết quả nghiên cứu:

Kết quả nghiên cứu của đề tài có thể được áp dụng cho các sàn thương mại điện tử để xây dựng hệ thống gợi ý sản phẩm. Hệ thống gợi ý sản phẩm sẽ giúp nâng cao trải nghiệm người dùng, tăng doanh số bán hàng và giảm chi phí marketing cho các sàn thương mại điện tử.

# CHƯƠNG II: CÁC CÔNG NGHỆ SỬ DỤNG

## 1. Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình bậc cao, mã nguồn mở và đa nền tảng. Python được Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ.

Đặc điểm của Python:

* Dễ học: Python có cú pháp đơn giản, dễ hiểu và gần gũi với ngôn ngữ tự nhiên. Điều này giúp người mới bắt đầu dễ dàng học lập trình Python.
* Dễ sử dụng: Python có thư viện và công cụ phong phú, giúp người lập trình dễ dàng thực hiện các tác vụ phức tạp.
* Tốc độ nhanh: Python có tốc độ thực thi nhanh, tương đương với các ngôn ngữ lập trình bậc thấp.
* Linh hoạt: Python có thể được sử dụng cho nhiều mục đích khác nhau, bao gồm phát triển web, ứng dụng di động, khoa học dữ liệu, máy học,...

Ứng dụng của Python

Python được ứng dụng trong nhiều lĩnh vực khác nhau, bao gồm:

* Phát triển web: Python là một ngôn ngữ lập trình phổ biến để phát triển web. Các framework web phổ biến như Django, Flask và Pyramid đều được viết bằng Python.
* Ứng dụng di động: Python cũng được sử dụng để phát triển ứng dụng di động. Các framework ứng dụng di động phổ biến như Kivy và PyQt5 đều được viết bằng Python.
* Khoa học dữ liệu: Python là một ngôn ngữ lập trình được sử dụng rộng rãi trong khoa học dữ liệu. Các thư viện khoa học dữ liệu phổ biến như NumPy, SciPy và Pandas đều được viết bằng Python.
* Máy học: Python cũng là một ngôn ngữ lập trình được sử dụng phổ biến trong máy học. Các thư viện máy học phổ biến như TensorFlow, PyTorch và Scikit-learn đều được viết bằng Python.

Các lợi ích của Python, Python mang lại một số lợi ích cho người dùng, bao gồm:

* Tăng năng suất: Python giúp người lập trình tăng năng suất bằng cách cung cấp cú pháp đơn giản, dễ hiểu và thư viện phong phú.
* Giảm chi phí: Python là một ngôn ngữ mã nguồn mở, giúp người dùng tiết kiệm chi phí mua bản quyền phần mềm.
* Tăng khả năng mở rộng: Python là một ngôn ngữ linh hoạt, có thể được sử dụng cho nhiều mục đích khác nhau. Điều này giúp người dùng tăng khả năng mở rộng hệ thống của họ.

## 2. Beautifullsoup - Crawler

Beautifulsoup là một thư viện Python được sử dụng để phân tích cú pháp HTML và XML, giúp cho việc trích xuất thông tin từ các trang web trở nên dễ dàng hơn.

Thư viện này cung cấp các công cụ để điều hướng, tìm kiếm và trích xuất dữ liệu từ cấu trúc HTML. Nó có thể giúp bạn tìm kiếm các thẻ HTML, tìm kiếm các thuộc tính của các thẻ đó, truy cập nội dung bên trong các thẻ và thậm chí có thể giúp bạn tìm kiếm các đoạn văn bản cụ thể trong tài liệu.

Sử dụng Beautifulsoup, bạn có thể trích xuất dữ liệu từ các trang web và lưu trữ nó trong các định dạng khác nhau như CSV, Excel, JSON hoặc SQL database để phân tích dữ liệu hoặc xây dựng các ứng dụng khác.

Điều đặc biệt về Beautifulsoup là nó rất linh hoạt và có thể sử dụng để phân tích cú pháp của hầu hết các ngôn ngữ đánh dấu, không chỉ riêng HTML hoặc XML.

## 3. Mongodb/ Mongodb  atlas - Lưu trữ

MongoDB lần đầu ra đời bởi MongoDB Inc., tại thời điểm đó là thế hệ 10, vào tháng Mười năm 2007, nó là một phần của sản phẩm PaaS (Platform as a Service) tương tự như Windows Azure và Google App Engine. Sau đó nó đã được chuyển thành nguồn mở từ năm 2009.

MongoDB đã trở thành một trong những NoSQL database nổi trội nhất bấy giờ, được dùng làm backend cho rất nhiều website như eBay, SourceForge và The New York Times.

MongoDB Atlas là một dịch vụ cloud database của MongoDB. Nó có giao diện người dùng được đơn giản hóa để định cấu hình và quản lý cơ sở dữ liệu cũng như nhiều tính năng khác như phân đoạn, phân cụm, sao chép,… Ta hoàn toàn có thể tùy chọn lưu trữ trên AWS, GCP hoặc Azure.

Với phiên bản Atlas miễn phí, ta sẽ được sử dụng những tính năng sau đây:

* Dung lượng lưu trữ 512 MB
* RAM dùng chung
* Bộ bản sao có sẵn cao, mã hóa đầu cuối, bản vá tự động, API REST
* Ngoài ra, khi nâng cấp lên phiên bản trả phí cao cấp hơn thì các tính năng này cũng được bổ sung như:
* 10 GB dung lượng lưu trữ trở lên
* RAM chuyên dụng
* Công cụ tối ưu hóa hiệu suất
* Sao lưu & phục hồi tại thời điểm
* Các tính năng bảo mật dành cho doanh nghiệp bao gồm quản lý khóa mã hóa và kiểm tra cơ sở dữ liệu chi tiết

## 4. Flask API

Flask là một Framework web của Python nhưng là một Micro-framework bởi nó chỉ có phần khung cơ bản, các phần còn lại cho phép người dùng tùy chọn mà đắp vào, không yêu cầu các công cụ hoặc thư viện cụ thể. Được phát triển bởi Armin Ronacher, người dẫn đầu Pocco – một nhóm những người đam mê Python quốc tế. Flask phát triển dựa trên bộ công cụ Werkzeug WSGI, Jinja2 và cả hai đều là các dự án của Pocco.

Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.

Là một Micro-framework nên Flask rất gọn nhẹ, có tính độc lập và bảo mật cao do ít phụ thuộc vào các thư viện bên ngoài.

## 5. Ngrok

Ngrok là công cụ tạo đường hầm (tunnel) giữa localhost của bạn và internet. Giúp các thiết bị có kết nối internet có thể truy cập được localhost của bạn thông qua custom domain của ngrok Ví dụ: ngrokdoamain => localhost:8080

Lợi ích của Ngrok:

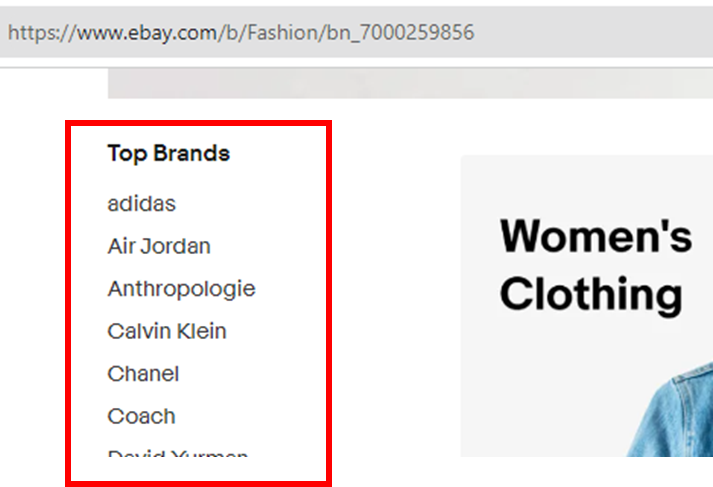
* Giúp bạn chạy demo dự án cho khách hàng xem từ chính máy của bạn mà không cần deploy lên server
* Bạn có thêm cách test responsive trên smart phone một cách dễ dàng thông qua URL mà ngrok cung cấp
* Hỗ trợ http, https, tcp
* Hỗ trợ IP whitelist

# CHƯƠNG III: THU THẬP DỮ LIỆU

## 1. Mô tả dữ liệu

Bộ dữ liệu bao gồm 67477 sản phẩm được crawl trên ebay bao gồm các hình ảnh và thông tin về sản phẩm (Id, brand, Link, Condition, Item\_specifics, Located , Name, Payment, Quantity, sold, category)

Các sản phẩm thuộc 26 thương hiệu trong Top Brands.



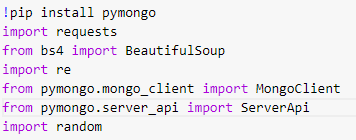
Hình 1. Các brand cần thu thập

## Thu thập dữ liệu

### Dữ liệu về thông tin sản phẩm

- Nơi lưu trữ: MongoDB Atlas

- Các thư viện sử dụng để crawl sản phẩm:



Đầu tiên lấy danh sách các link sản phẩm bằng cách duyệt qua từng trang của từng brand

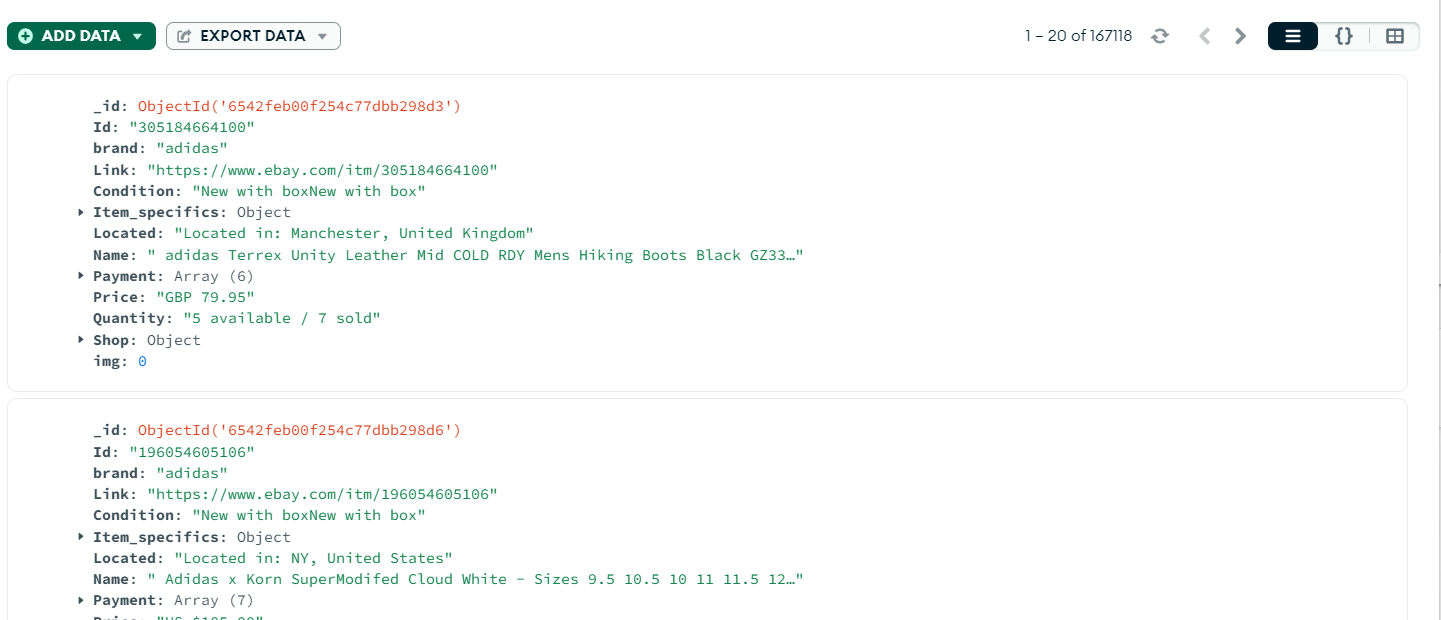




Hình 2. Kết quả thu thập link sản phẩm

Sau khi có danh sách các link sản phẩm ta bắt đầu crawl thông tin từng sản phẩm

Chúng ta duyệt các element để lấy được các link, từ các link chúng ta sẽ lấy được thông tin của trong từng sản phẩm cụ thể. Thực hiện Bước 1: Gửi http request lên trang để lấy về trang dạng document Bước 2: Từ trang đã lấy được về, trích xuất ra những link có được để vào detail của thông tin cần lấy Bước 3: Lấy data dựa trên DOM vừa tìm được.

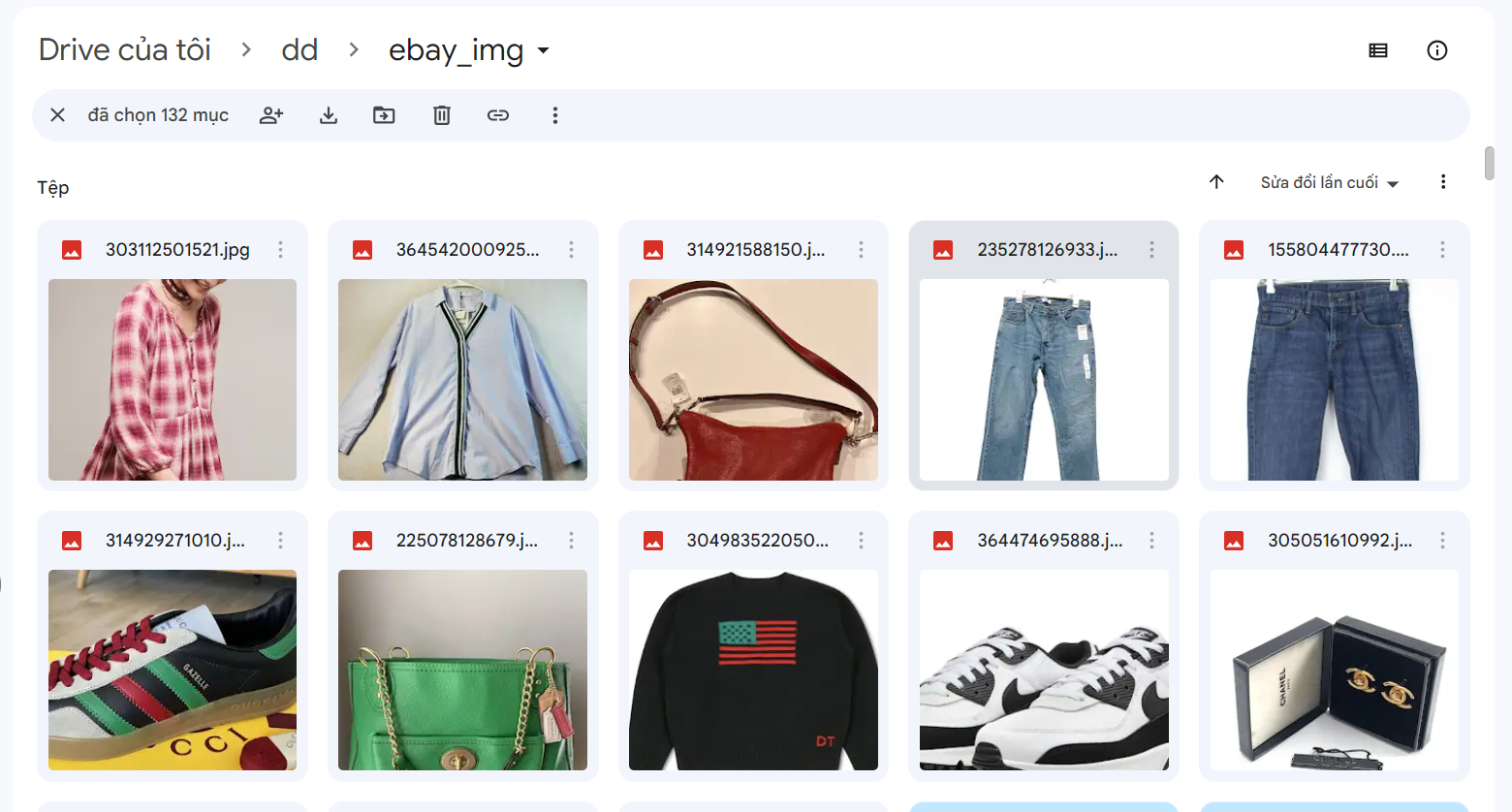


Hình 3. Kết quả thu thập thông tin sản phẩm

### Dữ liệu về hình ảnh

- Nơi lưu trữ: Google Drive

- Tương tự như thu thập thông tin sản phẩm, ta lần lượt duyệt qua các sản phẩm đồng thời lấy các DOM url hình ảnh sản phẩm sau đó lưu lại hình ảnh từ url bằng hàm urlretrieve của thư viện urllib.



Hình 4. Ảnh thu thập được lưu trên Google Drive

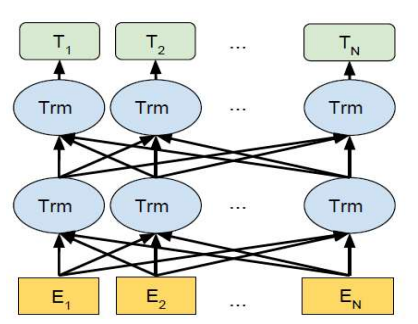
# CHƯƠNG IV: ĐỀ XUẤT GIẢI PHÁP

## Giới thiệu mô hình BERT

BERT (Bidirectional Encoder Representations from Transformers) (tạm dịch: Mô hình mã hóa hai chiều dữ liệu từ các khối Transformer), là một phương pháp kỹ thuật được xây dựng dựa trên mô hình mạng mô phỏng theo hệ thống nơ-ron thần kinh của con người (neural network) dùng để đào tạo trước (pre-train) quá trình xử lý ngôn ngữ tự nhiên. Nói một cách đơn giản, thì nó có thể được sử dụng để giúp Google phân biệt rõ hơn ngữ cảnh của các từ xuất hiện trong truy vấn tìm kiếm.

Điểm đột phá của BERT nằm ở khả năng huấn luyện các mô hình ngôn ngữ dựa trên toàn bộ tổ hợp các từ trong một câu hoặc truy vấn (huấn luyện hai chiều), thay vì cách thức huấn luyện truyền thống dựa trên thứ tự xuất hiện của các từ (từ trái qua phải hoặc kết hợp giữa trái qua phải và phải qua trái). BERT cho phép mô hình ngôn ngữ học về ngữ cảnh của từ vựng dựa trên các từ xung quanh nó, thay vì chỉ dựa vào từ ngữ đứng trước hoặc ngay sau nó.

Kiến trúc mô hình BERT là một bộ mã hóa Transformer hai chiều (bidirectional Transformer encoder). Việc sử dụng Transformer không có gì đáng ngạc nhiên vì đây là một xu hướng gần đây do tính hiệu quả và hiệu suất vượt trội của huấn luyện Transformers trong việc phát hiện các phụ thuộc với khoảng cách xa (long-distance dependencies) so với kiến trúc Recurrent neural network. Trong khi đó, bộ mã hóa hai chiều (bidirectional encoder) là một tính năng nổi bật giúp phân biệt BERT với OpenAI GPT (sử dụng từ trái sang phải Transformer) và kết hợp giữa huấn luyện từ trái sang phải và một mạng riêng rẽ từ phải sang trái LSTM. Kiến trúc của mô hình Bert được mô tả theo sơ đồ Hình 2.3.



Hình 5. Kiến Trúc mô hình BERT

Sử dụng bộ mã hóa Transformer đã được huấn luyện, BERT có thể biểu diễn bất kỳ token nào dựa trên ngữ cảnh hai chiều của nó. Trong quá trình học có giám sát trên các tác vụ, BERT tương tự như GPT ở hai khía cạnh.

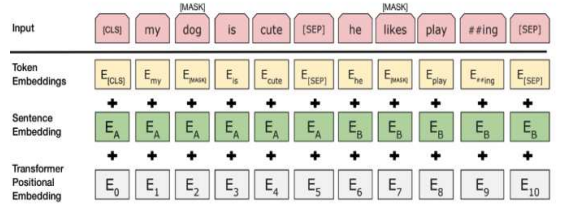
* Đầu tiên, các biểu diễn BERT sẽ được truyền vào một tầng đầu ra được bổ sung, với những thay đổi tối thiểu tới kiến trúc mô hình tùy thuộc vào bản chất của tác vụ, chẳng hạn như dự đoán cho mỗi token hay dự đoán cho toàn bộ chuỗi.
* Thứ hai, tất cả các tham số của bộ mã hóa Transformer đã được huấn luyện đều được tinh chỉnh, trong khi tầng đầu ra bổ sung sẽ được huấn luyện từ đầu.

## Biểu diễn đầu vào của BERT

Đầu vào có thể biểu diễn của một câu văn bản đơn hoặc một cặp câu văn bản. (Ví dụ: [Câu hỏi, câu trả lời]) được đặt thành một chuỗi tạo bởi các từ.

Chuỗi đầu vào BERT biểu diễn một cách tường minh cả văn bản đơn và cặp văn bản. Với văn bản đơn, chuỗi đầu vào BERT là sự ghép nối của token phân loại đặc biệt “”, token của chuỗi văn bản, và token phân tách đặc biệt “”. Với cặp văn bản, chuỗi đầu vào BERT là sự ghép nối của “”, token của chuỗi văn bản đầu, “”, token của chuỗi văn bản thứ hai, và “”. Ta sẽ phân biệt nhất quán thuật ngữ “chuỗi đầu vào BERT” với các kiểu “chuỗi” khác. Chẳng hạn, một chuỗi đầu vào BERT có thể bao gồm cả một chuỗi văn bản hoặc hai chuỗi văn bản.

Khi có một chuỗi đầu vào cụ thể được xây dựng bằng cách tính tổng các token đó với vector phân đoạn và vị trí tương ứng của các từ trong chuỗi. Cho dễ hình dung, biểu diễn đầu vào được trực quan hóa trong hình dưới đây:



Hình 6. Mô hình đại diện đầu vào của BERT

Token đầu tiên cho mỗi chuỗi được mặc định là một token đặt biệt có giá trị là [CLS]. Đầu ra của Transformer tương ứng với token này sẽ là được sử dụng để đại diện cho cả câu trong các nhiệm vụ phân loại nhãn chương trình. Nếu không trong các nhiệm vụ phân loại vector này được bỏ qua.

Trong trường hợp các cặp câu được gộp lại với nhau thành một chuỗi duy nhất chúng ta phân biệt các câu theo hai cách. Đầu tiên, chúng ta tách chúng bởi một token đặt biệt [SEP]. Thứ hai, chúng ta thêm một segment embedding cho câu A và một segment embedding khác cho câu B.

Khi chỉ có một câu đơn duy nhất segement embedding chỉ có cho câu A.

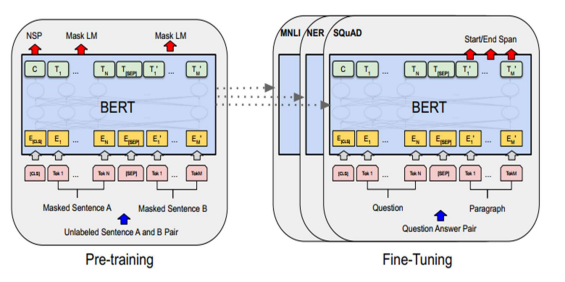
Kiến trúc hai chiều của BERT là bộ mã hóa Transformer. Thông thường trong bộ mã hóa Transformer, các embedding vị trí được cộng vào mỗi vị trí của chuỗi đầu vào BERT. Tuy nhiên, khác với bộ mã hóa Transformer nguyên bản, BERT sử dụng các embedding vị trí có thể học được cho thấy các embedding của chuỗi đầu vào BERT là tổng các embedding của token, embedding đoạn và embedding vị trí.

## Cải thiện BERT

BERT được training bằng cách sử dụng “Mask Language Model”. Ý tưởng của phương pháp là khi ta đưa một nội dung vào mô hình thì 15% token của một nội dung sẽ được thay thế bằng token <mask> việc của mô hình là sẽ dự đoán từ được <mask>.

VD: Sau đó đám cháy đã lan nhanh sang các khu vực khác.

Sau đó <mask> cháy đã lan nhanh sang các khu vực khác.



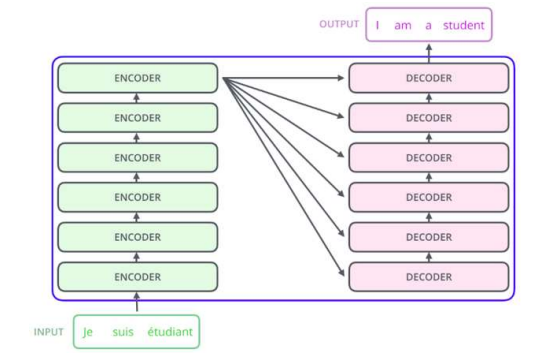
Hình 7. Quy trình tổng thể pre-training và fine-tuning của BERT

Một kiến trúc được sử dụng cho cả pretrain-model và fine-tuning model. Chúng ta sử dụng cùng một tham số huấn luyện trước để khởi tạo mô hình cho các tác vụ downstream khác nhau. Trong suốt quá trình điều chỉnh thì toàn bộ các tham số của các lớp học chuyển giao sẽ được tinh chỉnh. [CLS] là một ký tự (token) đặc biệt được thêm vào trước mọi ví dụ đầu vào và [SEP] là một ký tự phân tách đặc biệt (ví dụ: phân tách các câu hỏi / câu trả lời).

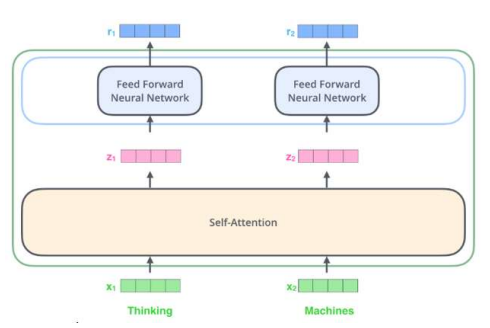
## Kiến trúc của BERT

Transformers là mô hình sử dụng phương thức attention. Sự ra đời của transformer đã giải quyết được vấn để của các mô hình như LSTM, BiLSTM là mất thông tin, thời gian training dữ liệu lâu. Phương thức attention hiểu đơn giản các từ quan trọng trong một câu sẽ có ảnh hưởng lớn hơn đến ý nghĩa của câu.

Transformers gồm 2 phần encoder và decoder: bao gồm nhiều block xếp chồng lên nhau. Với BERT chỉ sử dụng Encoder.

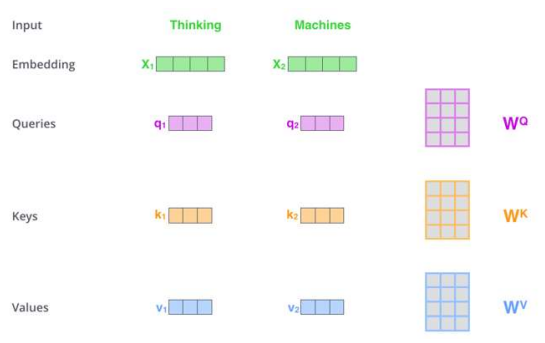


Hình 8. Kiến trúc của Transformers



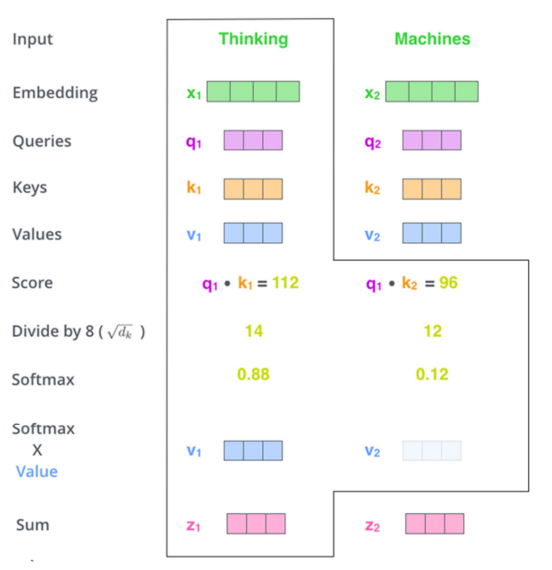
Hình 9. Kiến trúc của Kiến trúc của một block transformer

Các từ khi đưa vào sẽ được embedding về dạng vector (ở ví dụ trên ta có vector có kích thước là 1x64) được thu gọn lại để dễ biểu diễn. Tiếp theo sẽ đến bước tính toán.



Hình 10. Mô hình kiến trúc Self-Attention

Ta có 3 ma trận tham số Wq, Wk, Wv đại diện cho Querry, Key, Value sẽ được khởi tạo có kích thước số hàng bằng với chiều của vector, số cột tùy thuộc vào cách ta chọn. Ta lần lượt lấy các vector của từ nhân với các ma trận này ta sẽ thu được các vector tương ứng như trên Hình 2.9 (Tích ma trận)



Hình 11. Mô hình tính một vector Attention

Tiếp đó ta sẽ lần lượt lấy vector q nhân tích vô hướng với các vector k trong câu ta sẽ thu được trọng số. Các trọng số được chia cho căn bậc 2 chiều của vector (ở đây vector có chiều dài là 64) sau đó đưa qua hàm softmax. Làm lần lượt tương tự với các từ còn lại trong câu ta sẽ thu được các vector z còn lại. Đó là kiến trúc của 1 block encoder transformer.

Như vậy việc xếp chồng nhiều block thì output của 1 block sẽ là input của block khác. (Vector z sẽ có cùng kích thước với vector đầu vào). Mô hình BERT sẽ có 12 block xếp chồng lên nhau, hay còn gọi là 12 lớp. Với BERT large ta sẽ có 24 block.

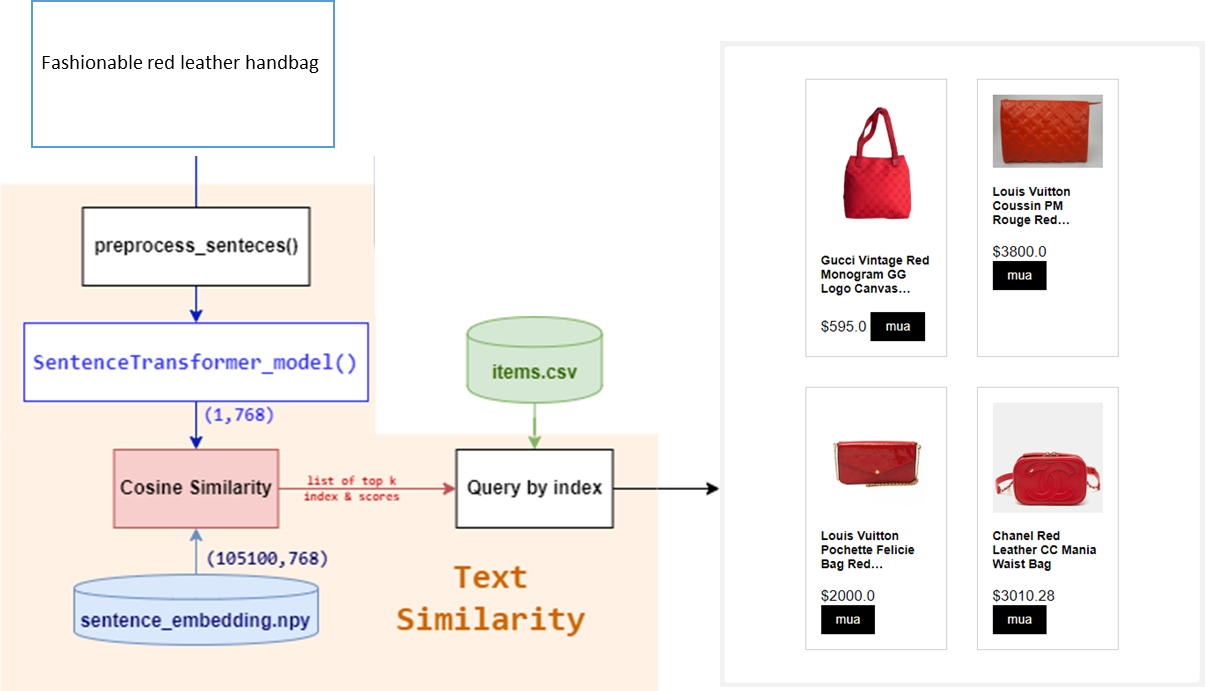
# CHƯƠNG V: TRIỂN KHAI ỨNG DỤNG

Sơ đồ tổng quát về mô hình gợi ý sản phẩm eBay sử dụng thuật toán BERT bao gồm các thành phần chính sau:

Dữ liệu đầu vào: Dữ liệu đầu vào của mô hình bao gồm mô tả về sản phẩm do người dùng nhập vào.

Mô hình BERT: Mô hình BERT được sử dụng để tính độ tương đồng giữa các sản phẩm dựa trên các thông tin đầu vào. BERT là một mô hình học máy lớn được đào tạo trên một tập dữ liệu khổng lồ gồm văn bản và mã.

Kết quả đầu ra: Kết quả đầu ra của mô hình là danh sách các sản phẩm tương tự với sản phẩm mà người dùng đang quan tâm. Danh sách này được sắp xếp theo độ tương đồng giảm dần.



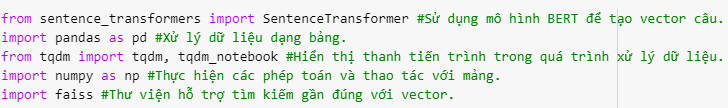
Hình 12. Sơ đồ tổng thể về ứng dụng

## 1. Train model

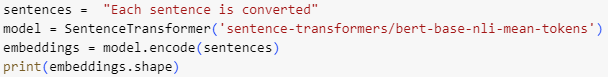
Cài đặt thư viện sentence-transformers



Import các thư viện cần thiết

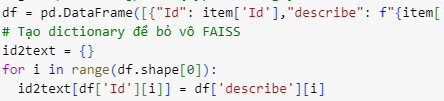


Tải xuống và sử dụng mô hình BERT: Tạo một đối tượng SentenceTransformer với tên model 'sentence-transformers/bert-base-nli-mean-tokens' để sử dụng mô hình BERT đã được huấn luyện sẵn.

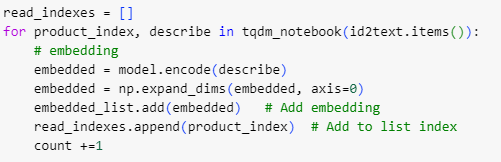


Lọc và chuẩn bị dữ liệu:

* Tạo DataFrame df chứa các thông tin sản phẩm cần thiết (Id, mô tả sản phẩm gộp từ Name và it\_spec).
* Tạo dictionary id2text để ánh xạ giữa Id sản phẩm và mô tả sản phẩm.

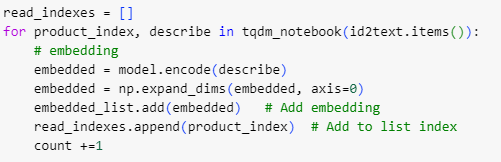


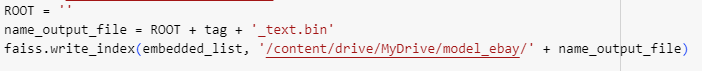
Tạo embedding cho các mô tả sản phẩm: Sử dụng hàm model.encode để tạo vector embedding 768 chiều cho từng mô tả sản phẩm trong dictionary id2text.



Tạo và lưu trữ index FAISS:

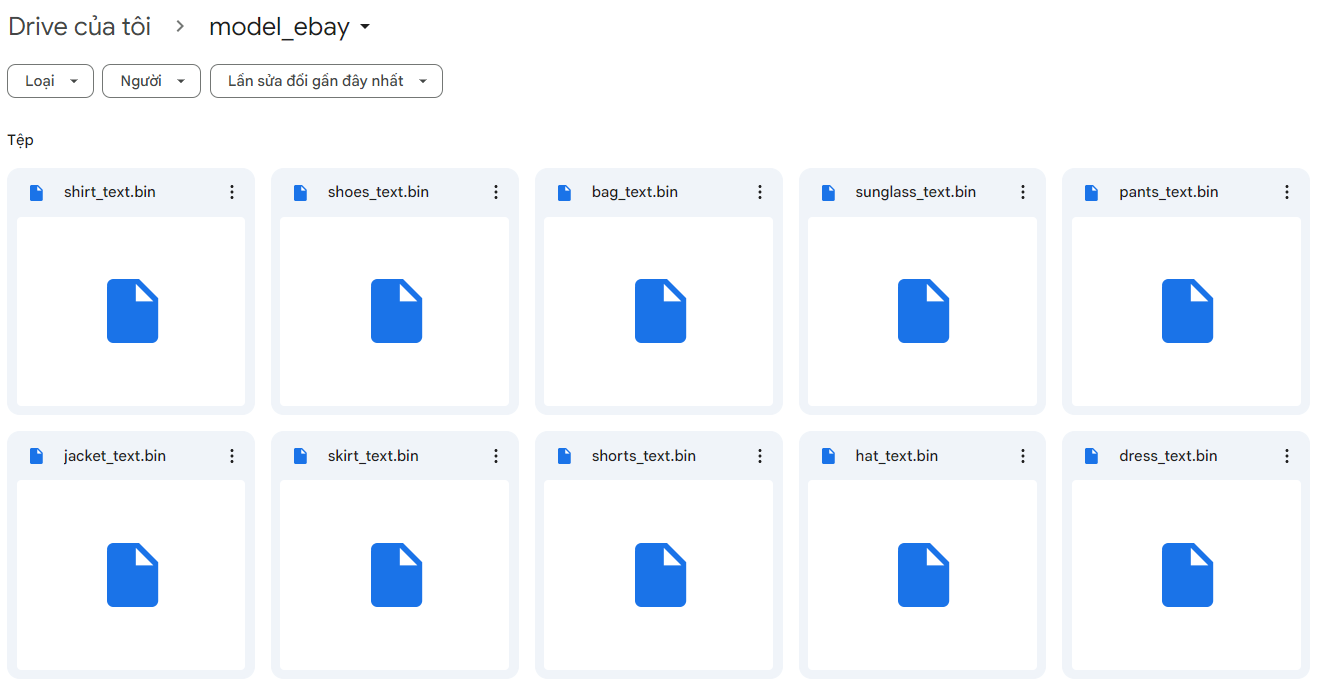
* Tạo đối tượng faiss.IndexFlatL2 với số chiều 768 để lưu trữ các vector embedding.
* Duyệt qua các vector embedding và thêm vào index FAISS.
* Lưu trữ index FAISS vào file .bin để sử dụng cho việc tìm kiếm và gợi ý sản phẩm sau này.





## 2. Sử dụng model

Sau quá trình train, tất các các model danh mục được lưu trong google Drive với tên file <danh mục>\_text.bin



Hình 13. Các hình ảnh được lưu trên Google Drive

Đặt tag:

tag = 'hat' chọn tag "hat" để tìm kiếm sản phẩm liên quan đến mũ.

Tạo embedding cho truy vấn tìm kiếm:

* new\_embed = 'red' đặt truy vấn tìm kiếm là "red" để tìm kiếm sản phẩm mũ màu đỏ.
* Sử dụng model BERT để tạo embedding cho truy vấn: new\_embed = model.encode(new\_embed)
* Mở rộng thêm một chiều cho embedding để phù hợp với FAISS: new\_embed = np.expand\_dims(new\_embed, axis=0)

Đọc index FAISS:

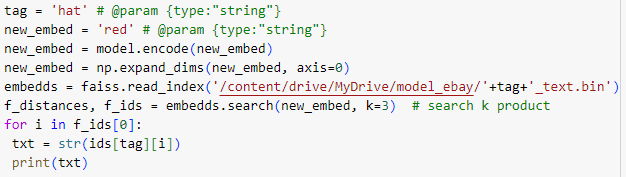
* embedds = faiss.read\_index('/content/drive/MyDrive/model\_ebay/'+tag+'\_text.bin') đọc index FAISS đã lưu từ trước cho tag "hat".

Tìm kiếm sản phẩm tương đồng:

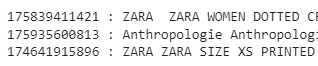
* f\_distances, f\_ids = embedds.search(new\_embed, k=3) tìm kiếm 3 sản phẩm có embedding gần nhất với embedding của truy vấn "red" trong index FAISS.

In thông tin sản phẩm tìm được:

* for i in f\_ids[0]:: Duyệt qua các ID sản phẩm được tìm thấy.
* txt = str(ids[tag][i]): Lấy thông tin mô tả sản phẩm từ dictionary ids (được tạo trong quá trình train model).
* print(txt): In mô tả sản phẩm ra màn hình.



Kết quả: Code sẽ in ra mô tả của 3 sản phẩm tag X có liên quan đến new\_embed nhất trong bộ dữ liệu được sử dụng để train model.



## 3. Triển khai ứng dụng model lên web

Cuối cùng, sử dụng thư viện Flask để hỗ trợ người dùng tương tác với mô hình thông qua nền tảng web.

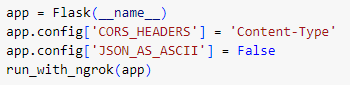
Khởi tạo ứng dụng Flask:

app = Flask(\_\_name\_\_): Tạo một đối tượng Flask mới để khởi động ứng dụng web.

app.config['CORS\_HEADERS'] = 'Content-Type': Cho phép chia sẻ tài nguyên từ nhiều nguồn gốc khác nhau (CORS).

app.config['JSON\_AS\_ASCII'] = False: Đảm bảo mã hóa JSON chính xác, hỗ trợ các ký tự không phải ASCII.

run\_with\_ngrok(app): Sử dụng ngrok để tạo một URL công khai tạm thời, giúp ứng dụng có thể truy cập được từ internet (thường dùng trong quá trình thử nghiệm).

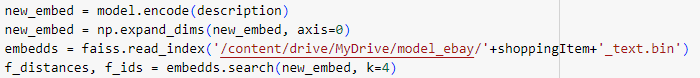


Định nghĩa các route:

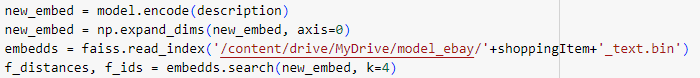
a. route "/" (trang chủ):

* @app.route("/"): Xác định tuyến đường gốc ("/") của ứng dụng.
* def home():: Định nghĩa hàm home() sẽ được gọi khi truy cập đường dẫn "/".
* try...except: Xử lý các lỗi tiềm ẩn.
* description = request.args.get("description"): Lấy mô tả sản phẩm từ tham số truy vấn "description".
* shoppingItem = request.args.get("shoppingItem"): Lấy danh mục sản phẩm từ tham số truy vấn "shoppingItem".

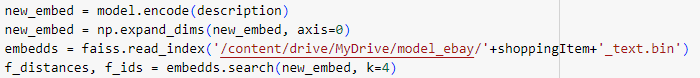
Sử dụng mô hình BERT để tạo vector biểu diễn cho mô tả sản phẩm:



Tải chỉ mục FAISS tương ứng với danh mục sản phẩm:



Tìm kiếm các sản phẩm tương tự trong chỉ mục FAISS:



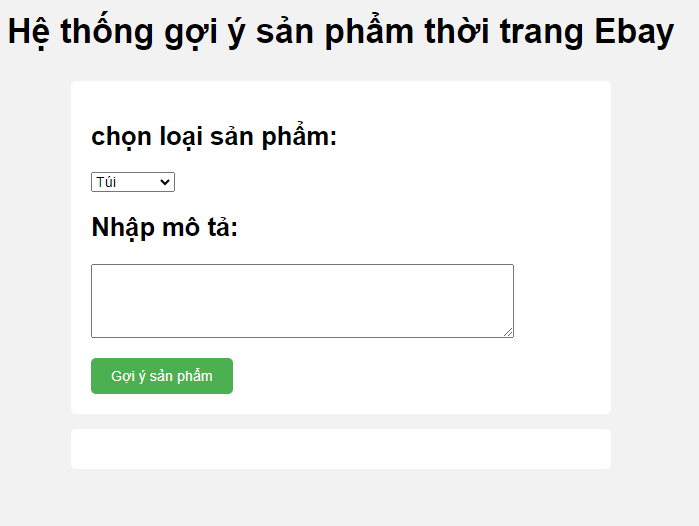
Tạo đoạn mã HTML để hiển thị các sản phẩm tương tự:

* Duyệt qua các ID sản phẩm tìm được (f\_ids)
* Lấy thông tin chi tiết của sản phẩm từ cơ sở dữ liệu ebaydb\_items
* Tạo các thẻ HTML hiển thị hình ảnh, tên, giá và nút "mua" cho mỗi sản phẩm

Trả về HTML đã tạo

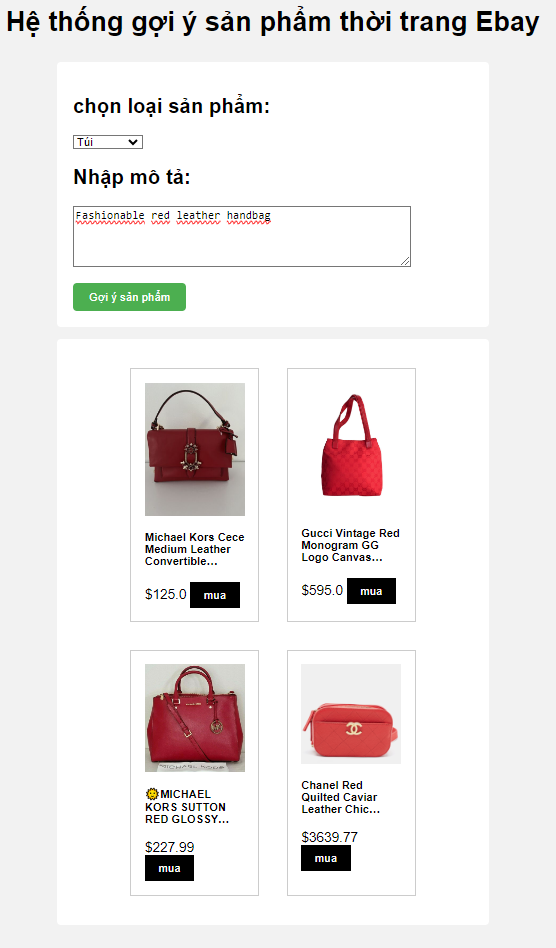
## 4. Kết quả

Kết quả trả về của trang web là một đoạn mã HTML chứa from để người dùng nhập mô tả tìm kiếm và lựa chọn loại sản phẩm cần gợi ý.



Hình 14. Giao diện hệ thống gợi ý sản phẩm

Khi ngưởi dùng nhập câu mô tả (VD: “Fashionable red leather handbag”) và chọn “Túi” thì web xuất ra kết quả:



Hình 15. Kết quả trả về từ web

Dựa trên hình ảnh có thể thấy hệ thống đưa ra các sản phẩm gợi ý khá tương đồng so với câu mô tả mà người dùng nhập vào.

# CHƯƠNG VI. KẾT LUẬN

Qua quá trình nghiên cứu và thực nghiệm, đề tài đã xây dựng thành công mô hình gợi ý sản phẩm eBay thông qua tính độ tương đồng, sử dụng thuật toán BERT. Mô hình sử dụng thuật toán BERT để tính độ tương đồng giữa các sản phẩm dựa trên các thông tin như tên sản phẩm, danh mục sản phẩm, các tính năng sản phẩm và câu mô tả sản phẩm do người dùng đưa ra.

Kết quả thực nghiệm cho thấy, mô hình đưa ra được sản phẩm có độ tương đồng với mô tả người dùng khá cao.

Việc phát triển mô hình gợi ý sản phẩm eBay sẽ giúp người dùng dễ dàng tìm thấy các sản phẩm phù hợp với nhu cầu của mình, từ đó tăng cường trải nghiệm mua sắm trên eBay.

Hướng phát triển, để nâng cao hiệu quả của mô hình, có thể thực hiện các hướng phát triển sau:

* Sử dụng thêm các thông tin khác để tính độ tương đồng giữa các sản phẩm, chẳng hạn như giá cả, hình ảnh, đánh giá sản phẩm, v.v.
* Sử dụng các thuật toán phân tích ngôn ngữ tự nhiên để hiểu rõ hơn về câu mô tả sản phẩm, từ đó có thể tính độ tương đồng chính xác hơn.
* Áp dụng các kỹ thuật học máy nâng cao để nâng cao độ chính xác của mô hình: Các kỹ thuật học máy nâng cao có thể giúp mô hình học hỏi hiệu quả hơn và nâng cao độ chính xác của dự đoán. Một số kỹ thuật học máy nâng cao có thể được sử dụng cho mục đích này bao gồm:

\* Học máy tăng cường: Học máy tăng cường là một kỹ thuật học máy trong đó mô hình học hỏi từ phản hồi của môi trường. Kỹ thuật này có thể được sử dụng để cải thiện hiệu quả của mô hình trong các tác vụ phức tạp.

\* Học máy truyền thông: Học máy truyền thông là một kỹ thuật học máy trong đó mô hình học hỏi từ dữ liệu từ nhiều nguồn khác nhau. Kỹ thuật này có thể được sử dụng để cải thiện hiệu quả của mô hình trong các tác vụ có dữ liệu lớn.

Với những hướng phát triển này, mô hình gợi ý sản phẩm eBay có thể trở nên hiệu quả hơn, giúp người dùng dễ dàng tìm thấy các sản phẩm phù hợp với nhu cầu của mình.

# TÀI LIỆU THAM KHẢO

[1] Bài 36 - BERT model – phamdinhkhanh https://phamdinhkhanh.github.io/2020/05/23/BERTModel.html

[2] Xingjie Feng and Yunze Zeng, Neural Collaborative Embedding from Reviews for Recommendation, IEEE, Vol. 20, 2017, pp. 1-9.

[3] ChanwooJeong, Sion Jang, Hyuna Shin, Eunjeong Park and Sungchul Choi, A Context-Aware Citation Recommendation Model with BERT and Graph Convolutional Networks, Test Engineering and Management, 2019.

[4] PavlosMitsoulis-Ntompos, MeisamHejazinia, Serena Zhang and Travis Brady, A Simple Deep Personalized Recommendation System, 2019, pp. 22-26.

[5] Zeynep Batmaz, Ali Yurekli, Alper Bilge and CihanKaleli, A review on deep learning for recommender systems: challenges and remedies, Artificial Intelligence Review, 2018.

[6] Dawen Liang, Minshu Zhan, and Daniel P. W. Ellis, Content-aware collaborative music recommendation using pre-trained neural networks. Semantic Scholar.

[7] Michael Tsang, Dehua Cheng, Hanpeng Liu, Xue Feng, Eric Zhou and Yan Liu, Feature interaction interpretability: A case for explaining ad-recommendation systems via neural interaction detection, ICLR, 2020, pp. 1-19.

[8] Michael Fleischman and Eduard Hovy, Recommendations Without User Preferences: A Natural Language Processing Approach, Proceedings of the 8th International Conference on Intelligent User Interfaces, Miami, Florida, USA, 2003.