|  |  |
| --- | --- |
|  |  |
| TRẦN XUÂN VŨ | BỘ CÔNG THƯƠNG  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI** |
| **--------------------------------------------------------------** |
| **A yellow sign with red and blue symbols  AI-generated content may be incorrect.** |
|  |
| **ĐỒ ÁN TỐT NGHIỆP**  NGÀNH CÔNG NGHỆ THÔNG TIN  **ĐỀ TÀI**  **XÂY DỰNG HỆ THỐNG PHÂN LOẠI TỔNG HỢP**  **ĐÁNH GIÁ CỦA KHÁCH HÀNG ĐA KÊNH, ĐA NỀN TẢNG**  **SỬ DỤNG MÔ HÌNH AI PHOBERT** |
| CÔNG NGHỆ THÔNG TIN | |  |  | | --- | --- | | **CBHD** | **: ThS. Nguyễn Đức Lưu** | | **Sinh viên** | **: Trần Xuân Vũ** | | **Mã sinh viên** | **: 2021601795** | | **Lớp** | **: 2021DHCNTT02** | | **Khóa** | **: K16** | |
| 2025 |  |
|  |
| Hà Nội – Năm 2025 |
|  |
|  |

# LỜI CẢM ƠN

Đối với sinh viên, được làm đồ án tốt nghiệp là một điều vinh dự và đáng tự hào. Đây không chỉ là cơ hội để sinh viên ứng dụng kiến thức đã tích lũy mà còn là thước đo đánh giá năng lực trước khi bước vào môi trường làm việc chuyên nghiệp. Bản thân em trong quá trình làm đồ án luôn trong tâm thế mong muốn thể hiện hết những gì mình có thể làm, những gì mình đã tích lũy, học tập từ thầy cô, bạn bè, đồng nghiệp. Qua đây, em cũng xin gửi lời cảm ơn chân thành tới ***Đại Học Công Nghiệp Hà Nội*** và các thầy cô trong ***khoa Công nghệ thông tin*** đã tạo điều kiện, môi trường học tập giúp em được tiếp cận với những kiến thức nền tảng, xu hướng xã hội là hành trang vững trãi giúp em tự tin trên con đường sự nghiệp. Đồng thời em xin gửi lời cảm ơn tới thầy ***ThS. Nguyễn Đức Lưu***, trong quá trình thực hiện đồ án tốt nghiệp, em đã nhận được sự quan tâm, giúp đỡ, hướng dẫn tận tình của thầy giúp em hoàn thành tốt khóa luận tốt nghiệp này. Những gì em nhận được không chỉ dừng lại ở kiến thức lý thuyết mà nhiều hơn thế đó là những lời khuyên, chia sẻ thực tế từ thầy. Chính nhờ những lời động viên, khích lệ của thầy mà em có cơ hội khám phá và phát huy khả năng của bản thân.

Một lần nữa, em xin chân thành cảm ơn!

Sinh viên thực hiện

***Vũ***

***Trần Xuân Vũ***

# MỤC LỤC

# bảng danh mục viết tắt

|  |  |  |
| --- | --- | --- |
| **Từ viết tắt** | **Nguyên nghĩa** | **Nghĩa tiếng Việt** |
| JS | JavaScript | Ngôn ngữ JavaScript |
| CSDL | Cơ sở dữ liệu | Cơ sở dữ liệu |
| JWT | Json Web Token |  |
| FE | Front End |  |
| BE | Back End |  |

# DANH MỤC HÌNH ẢNH

# MỞ ĐẦU

## Lý do chọn đề tài

Trong bối cảnh thương mại điện tử phát triển mạnh mẽ tại Việt Nam và trên toàn cầu, việc phân tích đánh giá của khách hàng trên các nền tảng thương mại điện tử đã trở thành một yếu tố cốt lõi giúp người bán hàng tối ưu hóa chiến lược kinh doanh. Các phương pháp đánh giá truyền thống dựa trên số sao tổng quan vẫn còn tồn tại nhiều hạn chế trong việc cung cấp thông tin chuyên sâu. Điểm số tổng thể không thể phản ánh chi tiết các khía cạnh mà khách hàng quan tâm như chất lượng, giá cả, đóng gói hay giao hàng. Một sản phẩm đạt 4 sao có thể nhận phản hồi tích cực về giá nhưng lại tiêu cực về vận chuyển. Bên cạnh đó, với số lượng bình luận khổng lồ trên các nền tảng khiến người bán hàng gặp khó khăn trong việc tổng hợp và khai thác các góc nhìn toàn diện.

Việc xây dựng một hệ thống tự động phân loại đánh giá theo cảm xúc (tích cực, tiêu cực, trung tính) và theo chủ đề (chất lượng, giá cả, giao hàng, …) mang lại những lợi ích thiết thực và có ý nghĩa quan trọng. Hệ thống giúp người bán thấu hiểu sâu sắc hơn về khách hàng thông qua việc xác định xu hướng tập trung vào các khía cạnh cụ thể của sản phẩm, từ đó điều chỉnh chiến lược Marketing hoặc cải tiến dịch vụ một cách hiệu quả. Việc ứng dụng mô hình PhoBERT – một mô hình tối ưu hóa cho tiếng Việt – nhằm xử lý ngữ nghĩa phức tạp và các từ ngữ địa phương trong bình luận, mang lại độ chính xác vượt trội so với các mô hình truyền thống như SVM hay LSTM.

Với những giá trị thực tiễn trên, đề tài “***Xây dựng hệ thống phân loại tổng hợp đánh giá của khách hàng đa kênh, đa nền tảng sử dụng mô hình AI phoBert***” không chỉ góp phần giải quyết các thách thức trong quản lý phản hồi khách hàng mà nâng cao khả năng ứng dụng AI trong việc tối ưu hóa trải nghiệm người dùng trong thời đại số hóa.

## Mục đích đề tài

Xây dựng một hệ thống tự động hóa việc phân tích và phân loại đánh giá của khách hàng trên các nền tảng thương mại điện tử, nhằm hỗ trợ người bán thấu hiểu sâu sắc hơn về phản hồi của khách hàng và đưa ra các chiến lược cải thiện sản phẩm, dịch vụ. Đề tài hướng tới việc cung cấp một giải pháp công nghệ giúp giải quyết vấn đề phân mảnh dữ liệu đa kênh, đồng thời đóng góp vào lĩnh vực ứng dụng trí tuệ nhân tạo trong xử lý ngôn ngữ tự nhiên tiếng Việt, tạo ra giá trị thực tiễn cho người kinh doanh trực tuyến.

## [Mục tiêu đề tài](#_44bvf6o)

* Ứng dụng mô hình PhoBERT để huấn luyện và phân loại bình luận theo cảm xúc và chủ đề, đảm bảo độ chính xác chấp nhận được.
* Xây dựng Website tổng hợp, thống kê, báo cáo đánh giá của khách hàng trên các nền tảng thương mại điện tử và mạng xã hội.
* Đảm bảo an toàn dữ liệu, thông tin người dùng và toàn bộ hệ thống.

## Bố cục đề tài

Nội dung chính đề tài gồm 4 chương

**Chương 1.** Tổng Quan Về Đề Tài

Trong chương này sẽ giới thiệu về bối cảnh, những khó khăn và hiện trạng và giải quyết vấn đề.

**Chương 2.** Cơ Sở Lý Thuyết và Công Nghệ.

Giới thiệu các lý thuyết công nghệ được sử dụng trong đề tài, bao gồm các cơ sở về AI và Website.

**Chương 3.** Thiết Kế và Triển Khai Hệ Thống.

Chương này sẽ giới thiệu về quy trình triển khai “**fine-tuning**” model PhoBert, xây dựng Backend phía Website và tích hợp chúng.

**Chương 4.** Kết Quả và Đánh Giá.

Chương này tập trung vào việc đánh giá độ chính xác của model AI, đánh giá hiệu năng của hệ thống.

## Phương pháp thực hiện

*Phương pháp nghiên cứu tài liệu*: Tham khảo, nghiên cứu thêm các tài liệu, bài báo và các công bố học thuật liên quan.

*Phương pháp thu thập dữ liệu*:  Xây dựng bộ dữ liệu thực tế phục vụ cho việc huấn luyện và đánh giá mô hình AI.

*Phương pháp thực nghiệm:* Huấn luyện và đánh giá mô hình phân loại dựa trên PhoBERT.

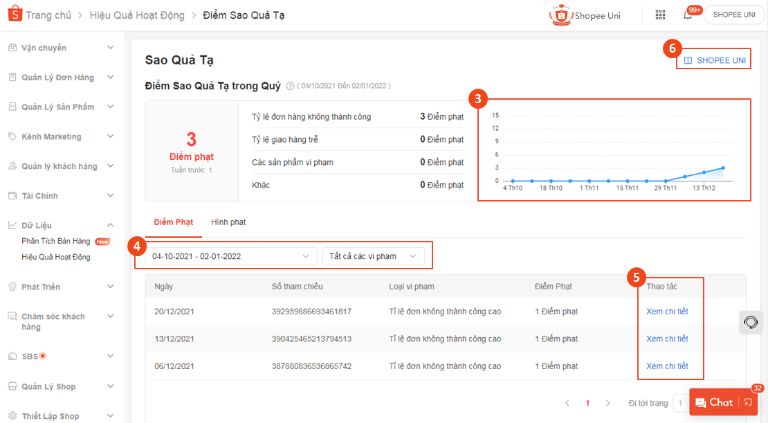
# Chương 1: Tổng quan đề tài

## Khảo sát và thực trạng hiện nay.

Hiện nay, các nền tảng thương mại điện tử phổ biến như Tiki, Shopee, Lazada chủ yếu sử dụng hệ thống đánh giá đơn giản bằng sao mà không đi sâu vào phân tích các khía cạnh cụ thể của sản phẩm như giá cả, chất lượng, đóng gói hay dịch vụ giao hàng. Hạn chế của hệ thống chỉ dựa vào đánh giá sao gây ra:

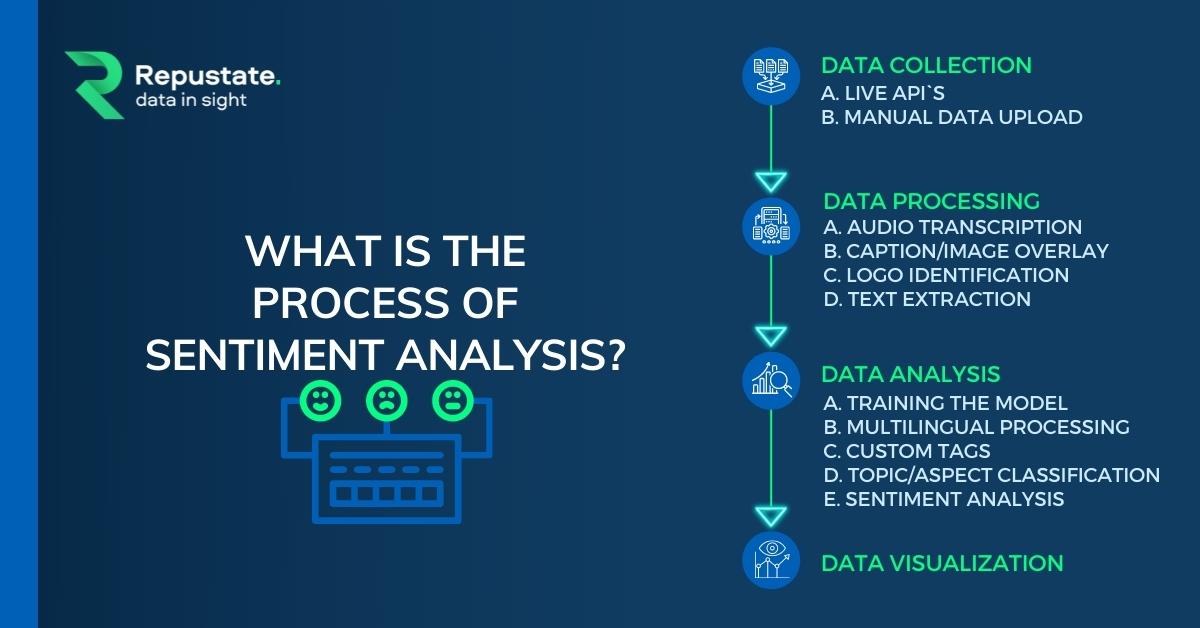
* Không phản ánh được các khía cạnh cụ thể mà khách hàng hài lòng hoặc không hài lòng.
* Không phản ánh được các khía cạnh cụ thể mà khách hàng hài lòng hoặc không hài lòng
* Thiếu khả năng phát hiện và phân loại tự động các loại phản hồi khác nhau.

Trên Shopee, người bán còn phải đối mặt với hệ thống "Sao Quả Tạ" - một hình thức tính điểm phạt dựa trên các yếu tố như tỷ lệ giao hàng thành công, tỷ lệ giao hàng đúng hẹn, hay vấn đề sản phẩm vi phạm. Tuy nhiên, hệ thống này chủ yếu đánh giá hiệu suất bán hàng mà không phân tích nội dung bình luận của khách hàng.

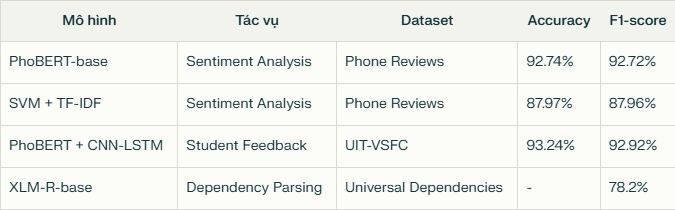


Các công cụ phân tích bình luận chuyên nghiệp trên thị thường cũng tập trung vào phân tích cảm xúc khách hàng như: Revuze, Qualtrics, Medallia, MonkeyLearn, …

Nổi bật có công cụ Repustate hỗ trợ phân loại chủ đề và phân loại cảm xúc trong chủ đề đó. Repustate có giải pháp NLP chuyên biệt cho tiếng Việt nhưng chỉ đưa ra các khẳng định chung chung về độ chính xác cao và khả năng tinh chỉnh, chứ không có báo cáo nghiên cứu hay benchmark cụ thể cho tiếng Việt.



* 1. **Xác định giải pháp cho vấn đề.**
* Sử dụng model AI phoBert - mô hình ngôn ngữ tiền huấn luyện đầu tiên được phát triển đặc biệt cho tiếng Việt, có nhiều nghiên cứu cho thấy hiệu suất của PhoBert tốt hơn so với các model truyền thống khác. Các nghiên cứu đều chỉ ra PhoBERT vượt trội nhờ:
  + Tối ưu hóa cho tiếng Việt: Tokenization theo subword, xử lý từ ghép và dấu thanh.
  + Học ngữ cảnh sâu: Kiến trúc transformer với 12/16 lớp encoder.
  + Tiền huấn luyện quy mô lớn: 20GB dữ liệu đa dạng (báo chí, mạng xã hội, văn bản hành chính).



* Dataset cho lĩnh vực thương mại điện tử
* Tích hợp nền tảng Website để có cái nhìn trực quan và toàn diện.

# Chương 2: Cơ sở lý thuyết và công nghệ

## 2.1. Bài toán phân loại cảm xúc và chủ đề trong văn bản.

### 2.1.1. Ngôn ngữ tự nhiên

Trong ngôn ngữ học, ngôn ngữ tự nhiên là ngôn ngữ nào phát sinh, không suy

nghĩ trước trong não bộ của con người. Một số ngôn ngữ điển hình mà con người được sử dụng để giao tiếp với nhau, có thể ngôn ngữ âm thanh, ngôn ngữ ký hiệu, các ký hiệu xúc giác hay chữ viết. Hiểu một cách đơn giản, ngôn ngữ tự nhiên (Natural Language) là ngôn ngữ mà con người dùng để giao tiếp với nhau như tiếng Việt, tiếng Anh, … và khác với ngôn ngữ nhân tạo như ngôn ngữ máy tính (Pascal, C, Python, …) hay mã Morse, Braille, ….

Theo thống kê, trên thế giới có khoảng 5600 ngôn ngữ, được phân bố rất không

đồng đều và chỉ có một số ít các ngôn ngữ là có chữ viết.

* Một số đặc điểm của ngôn ngữ tự nhiên:
* Ngôn ngữ tự nhiên là một hiện tượng xã hội đặc biệt.
* Ngôn ngữ tự nhiên là một trong những phương tiện giao tiếp quan trọng nhất
* của con người, các phương tiện khác cũng được diễn giải qua ngôn ngữ tự
* nhiên.
* Ngôn ngữ tự nhiên là một hệ thống các tín hiệu đặc biệt.
* Phân loại:
* Phân loại ngôn ngữ theo nguồn gốc lịch sử
* Phân loại ngôn ngữ theo trật tự từ
* Phân loại ngôn ngữ theo loại hình: được nhiều người sử dụng nhất.

Phân loại các ngôn ngữ tự nhiên theo loại hình là cách phân loại ngôn ngữ tự

nhiên theo cấu trúc và chức năng của ngôn ngữ tự nhiên. Từ việc phân loại người ta thu được các loại hình ngôn ngữ. Loại hình ngôn ngữ tự nhiên là một tập hợp các ngôn ngữ tự nhiên. Trong mỗi ngôn ngữ thì có ba nhóm thuộc tính: thuộc tính phổ quát (thuộc tính chung), thuộc tính riêng biệt, thuộc tính loại hình. Trong đó thuộc tính loại hình được dùng làm tiêu chuẩn khi phân loại ngôn ngữ.

### 2.1.2. Ngôn ngữ tiếng Việt

Tiếng Việt là ngôn ngữ đơn lập, nghĩa là trong mỗi âm tiết đều được phát âm tách rời nhau và được biểu diễn bằng một chữ viết cụ thể. Đặc điểm này được thể hiện ở tất cả các mặt như về ngữ âm, từ vựng, ngữ pháp.

**Đặc điểm ngữ âm**: Trong ngôn ngữ tiếng Việt thì ‘tiếng’ là một loại đơn vị đặc biêt. Về mặt ngữ âm, mỗi tiếng của tiếng Việt là một âm tiết. Hệ thống âm vị trong ngôn ngữ tiếng Việt thì rất phong phú và có tính cân đối. Trong ngôn ngữ tiếng Việt có rất nhiều từ được dùng để gợi hình, tượng thanh có giá trị gợi tả đặc sắc. Khi chúng ta viết câu, viết lời trong tiếng Việt thì phải chú ý đến sự hài hoà về ngữ âm, đến ngữ điệu của câu văn.

**Đặc điểm từ vựng**: Trong tiếng Việt. mỗi tiếng đều là một yếu tố có nghĩa. Tiếng là một đơn vị cơ sở trong hệ thống các đơn vị có nghĩa của ngôn ngữ tiếng Việt. Từ tiếng, người ta có thể tạo ra rất nhiều đơn vị từ vựng khác nhau để định danh cho sự vật, hiện tượng,… và chủ yếu được tạo ra bằng các phương thức ghép và phương thức láy. Việc tạo ra các đơn vị trong ngôn ngữ tiếng Việt ở phương thức ghép chịu sự chi phối của quy luật kết hợp về ngữ nghĩa, ví dụ: đất nước. xe lửa, nhà lầu xe hơi, dậu đổ bìm leo,…. Theo phương thức này, tiếng Việt sử dụng các yếu tố cấu tạo từ thuần Việt hay được vay mượn từ các ngôn ngữ khác nhau để tạo ra các từ ngữ mới, ví dụ: nhân viên, karaoke, thư điện tử (e-mail), hộp thư thoại (voice mail), phiên bản (version), xa lộ thông tin, văn bản siêu liên kết, truy cập ngẫu nhiên, .... Việc tạo ra các đơn vị từ vựng bằng phương thức láy thì quy luật phối hợp ngữ âm chi phối chủ yếu việc tạo ra các đơn vị từ vựng, chẳng hạn: đom đóm, bơ vơ, long lanh, ầm ầm, lấm tấm, …

**Đặc điểm ngữ pháp:** Từ của tiếng Việt đặc trưng là không biến đổi hình thái. Khi kết hợp các từ thành các kết cấu như ngữ, câu, phương thức trật tự từ và hư từ rất quan trọng. Việc sắp xếp các từ trong tiếng Việt theo một trật tự nhất định sẽ mạng ý nghĩa khác nhau qua đó biểu thị các quan hệ cú pháp. Trong tiếng Việt khi nói “Mùa xuân lại đến” là khác với “Lại đến mùa xuân“.Nhờ kết hợp trật tự của từ mà ngữ nghĩa của chúng cũng khác nhau. Trong tiếng Việt thì trật tự kết cấu câu chủ ngữ đứng trước, vị ngữ đúng sau là trật tự phổ biến nhất.

Phương thức hư từ cũng là một trong những phương thức ngữ pháp chủ yếu được sử dụng trong ngôn ngữ tiếng Việt. Nhờ hư từ mà tổ hợp các từ khác nhau có nghĩa khác nhau. Hư từ kết hợp với trật tự từ cho phép tiêng Việt tạo ra các câu về hình thức và nội dung cơ bản giống nhau nhưng khác nhau hoàn toàn về sắc thái biểu cảm. Ví dụ, so sánh các câu sau đây:

* Bạn ấy không uống nước ngọt.
* Nước ngọt, bạn ấy không uống.
* Nước ngọt, bạn ấy cũng không uống.

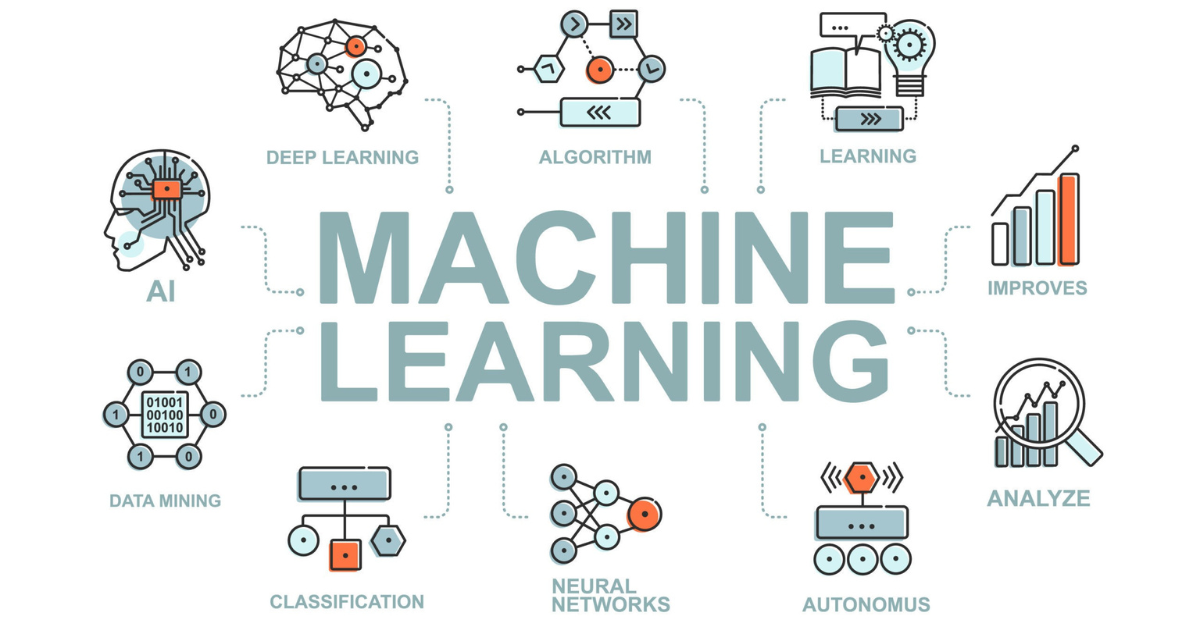
### 2.1.3. Các kiến thức học máy cơ bản

* + - 1. **Tổng quan về học máy**

**A diagram of machine learning

AI-generated content may be incorrect.**

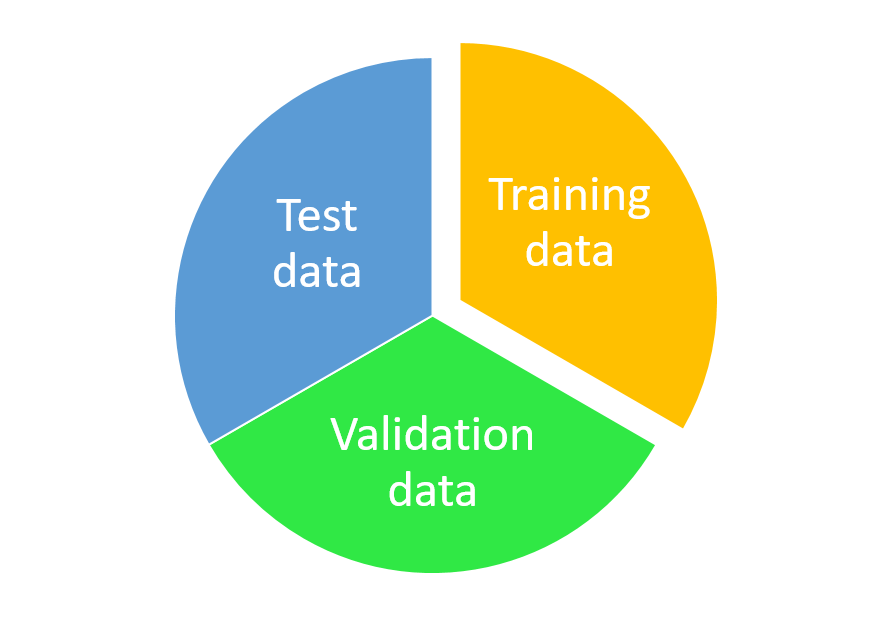
**Học máy (Machine Learning)** lĩnh vực nghiên cứu các thuật toán cho phép máy tính học từ dữ liệu mà không cần lập trình thủ công từng quy tắc. Mục tiêu chính là để máy tính “học” được mối quan hệ ẩn giữa dữ liệu đầu vào và kết quả đầu ra. Quá trình này bắt đầu bằng việc lựa chọn một mô hình, tức là một công thức hoặc cấu trúc toán học có tham số (parameters) mà ta sẽ điều chỉnh sao cho kết quả dự đoán càng gần với giá trị thực tế càng tốt. Ví dụ đơn giản nhất là mô hình hồi quy tuyến tính, với công thức y = w×x+b, trong đó w và b chính là các trọng số (weights) cần được “học” trong quá trình huấn luyện.

****

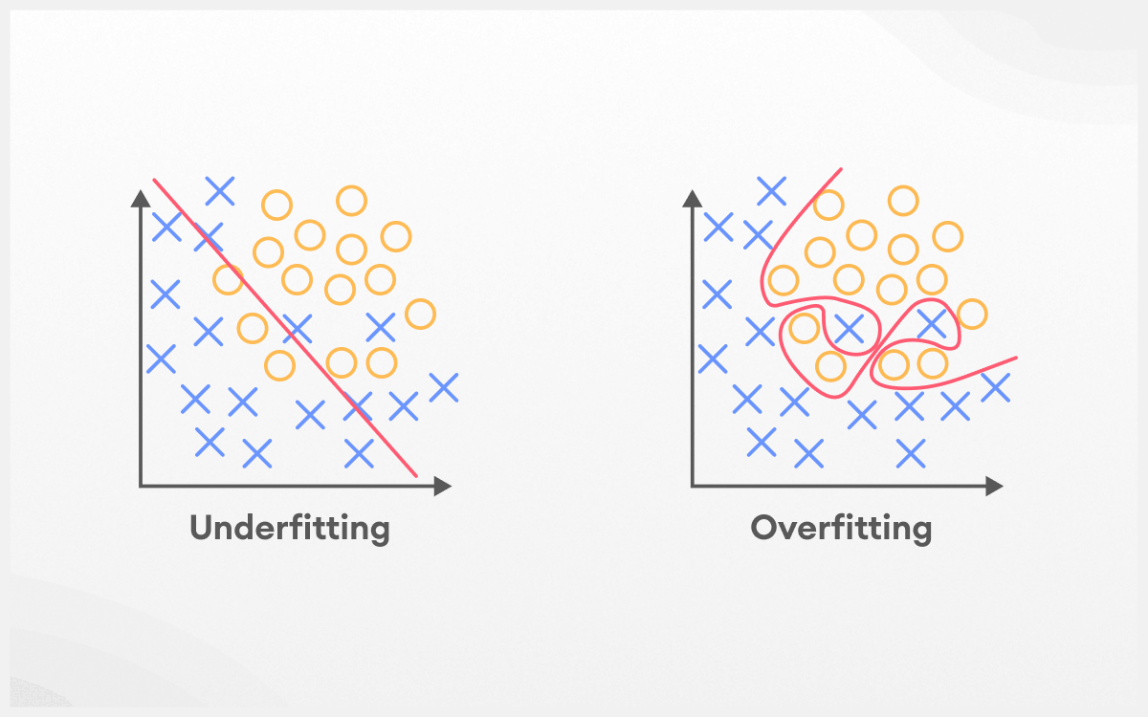
**Quá trình huấn luyện (Training)** diễn ra khi ta đưa tập dữ liệu đã gắn nhãn (gồm các cặp input-output) vào mô hình. Mỗi lần mô hình dự đoán (forward pass), ta sẽ so sánh kết quả dự đoán với nhãn thật và tính toán độ sai hay “hàm mất mát” (loss function). Tùy từng loại bài toán, ta chọn hàm mất mát phù hợp - ví dụ mean squared error cho bài toán hồi quy, hoặc cross-entropy cho bài toán phân loại. Mục tiêu của huấn luyện là điều chỉnh dần các trọng số sao cho giá trị hàm mất mát giảm xuống thấp nhất. Để điều chỉnh các trọng số này, hầu hết các mô hình hiện đại đều sử dụng thuật toán gradient descent. Về cơ bản, ta tính đạo hàm (gradient) của hàm mất mát theo từng trọng số, từ đó biết được hướng nào làm giảm sai số nhanh nhất và thực hiện một bước “lùi” nhỏ về hướng đó. Quá trình này lặp lại nhiều lần qua từng epoch - mỗi epoch là một lượt duyệt qua toàn bộ tập huấn luyện - cho đến khi hội tụ hoặc đạt được hiệu suất thoả mãn.

Machine Learning bao gồm học có giám sát (supervised learning), học không giám sát (unsupervised learning), Học tăng cường (Reinforcement Learning). Trong phạm vi đề tài này ta sẽ đi sâu vào học có giám sát. *Học có giám sát* là phương pháp học máy trong đó mô hình được huấn luyện trên tập dữ liệu đã có nhãn. Mỗi mẫu dữ liệu có một đầu vào (input) và một đầu ra (label). Mục tiêu là học được *hàm ánh xạ* từ đầu vào đến đầu ra, để có thể dự đoán cho dữ liệu mới chưa từng thấy như các bài toán phân loại (Classification), hồi quy (Regression),…

#### Phân chia tập dữ liệu.

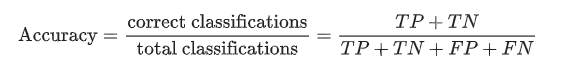


Trong thực tế, chia tập dữ liệu thành 3 tập nhỏ hơn bao gồm*: training set, validation set, test set* để đảm bảo mô hình không chỉ “nhớ” dữ liệu huấn luyện mà còn có khả năng áp dụng tốt trên dữ liệu mới, ta chia dữ liệu thành ba phần: tập huấn luyện (training set) dùng để học trọng số, tập kiểm định (validation set) dùng để đánh giá tạm thời và điều chỉnh siêu tham số (hyperparameters) như tốc độ học (learning rate), kiến trúc mạng, độ “rơi rụng” (dropout)…, và cuối cùng là tập kiểm tra (test set) dùng để đánh giá khách quan khả năng tổng quát hóa của mô hình sau khi mọi điều chỉnh đã hoàn tất. Nếu hiệu suất trên tập huấn luyện tốt nhưng trên tập validation kém hơn nhiều, đó là dấu hiệu của overfitting - mô hình học quá khớp vào dữ liệu cũ mà không hiểu được quy luật chung.



***Underfitting và Overfitting*** thể hiện việc mô hình chưa đạt được khả năng học và tổng quát hóa tốt, và thường biểu hiện rõ ràng qua độ sai số (loss) hoặc độ chính xác (accuracy):

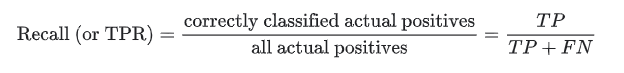
* *Chưa khớp (Underfitting)* xảy ra khi mô hình quá đơn giản so với tính phức tạp của dữ liệu. Nó không đủ khả năng học được các quy luật cơ bản trong tập huấn luyện, dẫn đến việc mô hình dự đoán sai ngay cả trên dữ liệu đã thấy. Khi Underfitting xảy ra, mô hình cần được tăng độ phức tạp, ví dụ như thêm tham số, sử dụng mô hình phi tuyến, hoặc huấn luyện lâu hơn.
* Quá khớp (overfitting) xảy ra khi mô hình quá phức tạp hoặc được huấn luyện quá lâu đến mức nó “học thuộc lòng” dữ liệu huấn luyện. Mô hình lúc này ghi nhớ từng điểm dữ liệu một cách chi tiết, bao gồm cả nhiễu (noise), thay vì học ra quy luật tổng quát. Biểu hiện thường thấy là mô hình có độ chính xác rất cao trên tập huấn luyện nhưng lại giảm mạnh khi áp dụng lên tập kiểm định hoặc kiểm tra.
  + - 1. **Đánh giá mô hình (Evaluation Metrics)**
* **Độ chính xác (Accuracy)**: Tỷ lệ phần trăm dự đoán đúng trên tổng số dự đoán.



* **Độ chính xác dương (Precision)**: Tỷ lệ phần trăm các dự đoán dương tính của mô hình mà thực sự đúng.



* **Độ thu hồi (Recall, hay Độ nhạy)**: Tỷ lệ phần trăm các trường hợp dương tính thực tế mà mô hình dự đoán đúng.



* **Điểm F1 (F1-score)**: Trung bình hài hòa của Precision và Recall

A math equation with black text

AI-generated content may be incorrect.

### 2.1.4. Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiêu (Natural Language Processing - NLP) là một công nghệ máy học, cung cấp cho máy tính khả năng diễn giải, tương tác và hiểu được ngôn ngữ của con người. Các tổ chức ngày nay có khối lượng lớn dữ liệu thoại và văn bản từ nhiều kênh liên lạc khác nhau như email, tin nhắn văn bản, bảng tin trên mạng xã hội, tệp video, tệp âm thanh và nhiều hơn nữa. Họ sử dụng phần mềm NLP để tự động sử lý dữ liệu này, phân tích ý định hoặc cảm xúc trong tin nhắn và phản hồi bằng người thật theo thời gian thực.

Trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), các mô hình học máy và học sâu thường gồm hai thành phần chính: bộ phân loại (classifier) và thành phần trích xuất đặc trưng (feature extraction). Sự kết hợp giữa hai thành phần này đóng vai trò cốt lõi trong việc biến đổi dữ liệu văn bản thô thành các dự đoán có ý nghĩa, chẳng hạn như phân loại cảm xúc hoặc nhận diện ý định.

**Biểu diễn ngôn ngữ (Language Representation):** là cách biến đổi văn bản thành dạng số giúp cho mô hình có thể xử lý được. Các bước như stemming, lemmatization (rút gọn từ), loại bỏ dấu câu cũng có thể áp dụng trước khi biểu diễn văn bản.

* **One-hot encoding**: Cách đơn giản nhất: gán mỗi từ trong từ vựng một vectơ có chiều dài bằng kích thước từ vựng, giá trị chỉ có 0 và 1 (1 tại vị trí tương ứng với từ đó). Nhược điểm là vectơ rất thưa và không phản ánh được quan hệ giữa các từ.
* **Bag-of-Words (Túi từ)**: Mô tả văn bản bằng ma trận/tập hợp tần số xuất hiện của mỗi từ (hoặc n-gram) trong văn bản, bỏ qua thứ tự. Kết hợp với TF-IDF để giảm trọng số của từ thông dụng. Phương pháp đơn giản, nhưng không nắm bắt được thứ tự hay ngữ nghĩa sâu.
* **Word Embeddings (Embedding từ)**: Kỹ thuật ánh xạ từ ngữ thành các vectơ số liên tục. Ví dụ nổi tiếng là Word2Vec (bao gồm mô hình skip-gram và CBOW) và GloVe. Các phương pháp này học được vectơ cố định đại diện mỗi từ sao cho các từ có ý nghĩa gần nhau sẽ có vectơ gần nhau trong không gian vectơ. Chẳng hạn, Word2Vec “học” biểu diễn cho mỗi từ cố định và giúp phản ánh tương tự về mặt ngữ nghĩa giữa các từ. Word2Vec biểu diễn mỗi từ bằng một vector có độ dài cố định và sử dụng những vector này để biểu thị tốt hơn độ tương tự và các quan hệ loại suy giữa các từ. Trước kia, embedding từ trở thành kiến thức cơ bản trong NLP.
* **Embedding theo ngữ cảnh (Contextual Embeddings)**: Khác với Word2Vec/GloVe tạo vectơ cố định cho một từ bất kể ngữ cảnh, mô hình như BERT sinh ra vectơ phụ thuộc ngữ cảnh của từ. Ví dụ, từ “running” trong câu “He is running a company” và trong “He is running a marathon” sẽ có vectơ giống nhau nếu dùng Word2Vec, nhưng BERT sẽ cho hai vectơ khác nhau tùy ngữ cảnh. Các mô hình không có ngữ cảnh như Word2Vec hay GloVe tạo ra biểu diễn nhúng cố định trong khi BERT sẽ cung cấp một nhúng theo ngữ cảnh mà sẽ khác nhau tùy thuộc câu. Điều này giúp mô hình hiểu nghĩa từ trong từng ngữ cảnh cụ thể.
* **Subword và Byte-Pair Encoding (BPE)**: Để xử lý từ mới hoặc ngôn ngữ có nhiều từ ghép, ta thường chia nhỏ từ thành subword (kí tự phụ) bằng phương pháp như BPE hoặc WordPiece. Ví dụ, từ “hoàn-thành” có thể được cắt thành các phần con. PhoBERT sử dụng BPE để tạo từ vựng ~64K subword. Phương pháp này giúp giảm số từ lạ (OOV) và ghi nhận cấu trúc của từ.
* **Mô hình ngôn ngữ tiền huấn luyện (Pre-trained Language Models)**: Hiện nay thường sử dụng các mô hình như BERT, GPT, ELMo… được huấn luyện trước trên kho văn bản lớn (ví dụ Wikipedia, sách, báo). Những mô hình này học được biểu diễn ngữ nghĩa sâu và có thể *transfer* (chuyển giao) sang nhiều tác vụ downstream khác. PhoBERT là một mô hình tiền huấn luyện như vậy dành cho tiếng Việt.

### 2.1.5. Các kiến thức về PhoBERT

PhoBERT là mô hình ngôn ngữ tiền huấn luyện đầu tiên được phát triển đặc biệt cho tiếng Việt, dựa trên nền tảng kiến trúc RoBERTa (phiên bản cải tiến của BERT). Phát triển bởi đội ngũ nghiên cứu VinAI Research, PhoBERT đã đạt được hiệu suất vượt trội so với các mô hình đa ngôn ngữ trước đó trên nhiều tác vụ xử lý ngôn ngữ tự nhiên tiếng Việt.

#### 2.1.5.1.  Kiến trúc Transformer và BERT

##### a. Tổng quan về Transformer

A diagram of a multi-headed attention

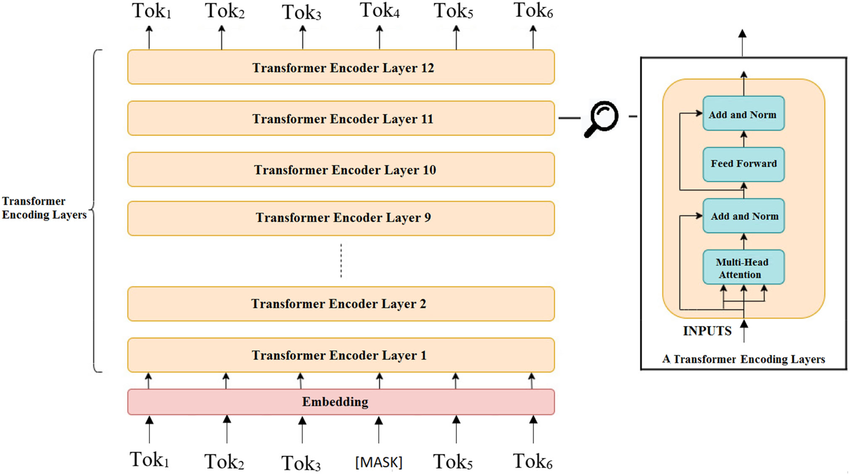
AI-generated content may be incorrect.

**Transformer** là một kiến trúc mô hình học sâu dựa trên cơ chế self-attention, cho phép mô hình này hiểu được mối quan hệ giữa các từ trong một câu mà không cần đến kiến trúc tuần tự truyền thống như RNN (Recurrent Neural Networks) hay LSTM (Long Short-Term Memory). Transformer có khả năng xử lý toàn bộ câu cùng một lúc, điều này giúp tăng tốc độ huấn luyện và cải thiện hiệu quả xử lý.

Cấu trúc encoder-decoder: Transformer gồm hai phần chính encoder xử lý đầu vào, decoder tạo đầu ra. Mỗi phần bao gồm nhiều lớp lặp lại. Ở bài toán phân loại và BERT, ta chỉ dùng phần encoder. Mỗi lớp encoder gồm: Multi-Head Self-Attention – chia thông tin qua nhiều “đầu” attention, cho phép mô hình học các mặt khác nhau của mối quan hệ từ khóa; Feed-forward Neural Network (FNN) - lớp thần kinh tuyến tính xử lý kết quả. Kết quả của hai khối này được cộng rồi chuẩn hóa. Việc có nhiều đầu attention cho phép xử lý thông tin đồng thời ở nhiều khía cạnh khác nhau.

Positional Encoding (Mã hóa vị trí): Vì Transformer xử lý cả câu không theo thứ tự, cần thêm thông tin về vị trí từ vào vectơ đầu vào. Thông thường dùng ma trận encode hàm sin/cos để mã hóa thứ tự từ. Ví dụ, từ ở vị trí 5 sẽ có một vectơ đặc trưng, khác với từ ở vị trí 3

##### b. Tổng quan về BERT



BERT (Bidirectional Encoder Representations from Transformers) là mô hình tiền huấn luyện dựa trên kiến trúc Transformer, được Google giới thiệu năm 2018. Khác với các mô hình trước đó chỉ xử lý thông tin theo một chiều (từ trái sang phải hoặc ngược lại), BERT có khả năng học biểu diễn ngữ nghĩa hai chiều, nhờ cơ chế Self-Attention trong Transformer. Điều này cho phép BERT hiểu mối quan hệ giữa các từ trong toàn bộ ngữ cảnh câu, kể cả khi chúng cách xa nhau. BERT được huấn luyện trước trên hai nhiệm vụ chính: Masked Language Modeling (MLM) (dự đoán từ bị che trong câu) và Next Sentence Prediction (NSP) (xác định hai câu có liên tiếp nhau không). Kiến trúc của BERT gồm nhiều lớp Encoder của Transformer, với hai phiên bản phổ biến là BERT-Base (12 lớp, 110 triệu tham số) và BERT-Large (24 lớp, 340 triệu tham số). Sau khi tiền huấn luyện, BERT có thể được tinh chỉnh (fine-tuning) cho nhiều tác vụ như phân loại văn bản, hỏi đáp, hoặc nhận dạng thực thể, mang lại hiệu suất vượt trội so với các mô hình truyền thống.

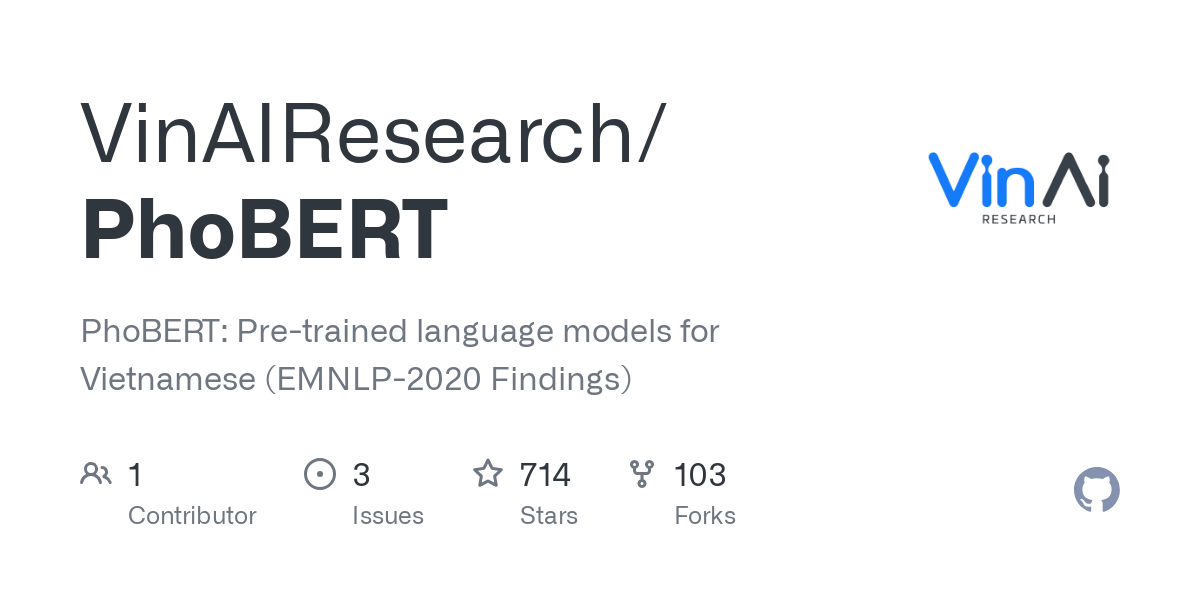
A diagram of a diagram

AI-generated content may be incorrect.

RoBERTa (Robustly Optimized BERT Approach), được Facebook AI nghiên cứu năm 2019, là phiên bản cải tiến của BERT, tập trung vào tối ưu hóa quá trình tiền huấn luyện. RoBERTa loại bỏ nhiệm vụ NSP, thay vào đó tập trung mở rộng quy mô dữ liệu (160GB so với 16GB của BERT) và áp dụng Dynamic Masking (tạo mặt nạ khác nhau cho mỗi epoch) để tránh overfitting. Bên cạnh đó, RoBERTa sử dụng batch size lớn hơn (8K samples/batch) và huấn luyện lâu hơn, giúp mô hình học sâu hơn vào cấu trúc ngôn ngữ. Những thay đổi này giúp RoBERTa đạt kết quả vượt trội trên các bộ benchmark như GLUE (điểm số tăng từ 80.5 lên 88.5) hay SQuAD.

Sự khác biệt chính giữa BERT và RoBERTa nằm ở chiến lược huấn luyện. Trong khi BERT dựa trên cả MLM và NSP, RoBERTa chỉ sử dụng MLM và tối ưu hóa việc học thông qua quy mô dữ liệu và kỹ thuật masking linh hoạt. Về ứng dụng, BERT được triển khai rộng rãi trong các hệ thống như Google Search để hiểu truy vấn người dùng, trong khi RoBERTa thường được dùng cho các tác vụ yêu cầu độ chính xác cao như dịch máy hoặc phân tích cảm xúc. Cả hai mô hình đều minh chứng cho sức mạnh của việc học biểu diễn ngôn ngữ dựa trên Transformer, mở đường cho các mô hình đa ngôn ngữ như PhoBERT trong tiếng Việt.

#### 2.1.5.2. Tổng quan về PhoBert - Vietnamese Pre-trained Language Model



PhoBERT (Vietnamese Pre-trained Bidirectional Encoder Representations from Transformers) là mô hình ngôn ngữ đơn ngữ đầu tiên quy mô lớn được tiền huấn luyện dành riêng cho tiếng Việt. Được phát triển bởi VinAI Research vào năm 2020, PhoBERT đã tạo nên bước đột phá trong lĩnh vực xử lý ngôn ngữ tự nhiên tiếng Việt. PhoBERT được xây dựng dựa trên kiến trúc RoBERTa - một phiên bản tối ưu hóa của BERT (Bidirectional Encoder Representations from Transformers), nhưng được điều chỉnh đặc biệt để xử lý các đặc thù của ngôn ngữ tiếng Việt.

PhoBERT hiện có hai phiên bản chính: PhoBERT-base với 135 triệu tham số và PhoBERT-large với 370 triệu tham số. Cả hai phiên bản đều được huấn luyện trên bộ dữ liệu lớn gồm 20GB văn bản tiếng Việt, bao gồm 1GB từ Wikipedia tiếng Việt và 19GB dữ liệu tin tức đã được lọc kỹ để loại bỏ nội dung trùng lặp. Điểm đặc biệt của PhoBERT là sử dụng phương pháp tokenization BPE (Byte-Pair Encoding) với từ vựng 64K, nhưng yêu cầu dữ liệu đầu vào phải được tách từ trước - một đặc điểm quan trọng khi xử lý tiếng Việt.

Kết quả thực nghiệm đã chứng minh PhoBERT vượt trội hơn các mô hình đa ngôn ngữ trước đó như mBERT và XLM-R, đồng thời thiết lập các tiêu chuẩn mới trong nhiều tác vụ NLP tiếng Việt như gán thẻ từ loại (POS tagging), phân tích cú pháp phụ thuộc (Dependency parsing), nhận dạng thực thể có tên (NER) và suy luận ngôn ngữ tự nhiên (NLI).

**Đặc điểm kỹ thuật của PhoBERT**:PhoBERT được xây dựng trên nền tảng kiến trúc RoBERTa - một phiên bản cải tiến của BERT nhằm tối ưu hóa hiệu suất. Về mặt kỹ thuật, PhoBERT có các đặc điểm quan trọng sau:

**Kiến trúc và tham số mô hình**: PhoBERT-base có kiến trúc gồm 12 lớp transformer, 12 attention heads, và vector nhúng kích thước 768 chiều, tổng cộng 135 triệu tham số có thể học. PhoBERT-large lớn hơn đáng kể với 24 lớp transformer, 16 attention heads, vector nhúng 1024 chiều, và tổng cộng 370 triệu tham số. Độ dài tối đa đầu vào cho PhoBERT là 256 token, ít hơn so với BERT thông thường (512 token) nhằm tối ưu hóa tốc độ huấn luyện và tiết kiệm tài nguyên.

**Môi trường và framework**: PhoBERT được huấn luyện với framework PyTorch thông qua thư viện fairseq, sử dụng 4 GPU NVIDIA V100 mạnh mẽ. Thời gian huấn luyện khá dài: 3 tuần cho PhoBERT-base và 5 tuần cho PhoBERT-large. Điều này phản ánh quy mô và độ phức tạp của quá trình huấn luyện, đồng thời cho thấy mức đầu tư đáng kể vào tài nguyên tính toán để tạo ra mô hình chất lượng cao.

**Đặc điểm xử lý ngôn ngữ**: Một điểm đặc biệt của PhoBERT là yêu cầu dữ liệu đầu vào phải được tách từ trước khi tokenization, khác với nhiều mô hình BERT đa ngôn ngữ khác. Điều này phù hợp với đặc thù của tiếng Việt, nơi ranh giới từ không được xác định bằng khoảng trắng như trong tiếng Anh. PhoBERT sử dụng phương pháp tokenization BPE với kích thước từ vựng 64K token con từ để xử lý hiệu quả các từ tiếng Việt sau khi đã được tách.

**Quá trình huấn luyện PhoBERT**: Quá trình huấn luyện PhoBERT là một công đoạn kỹ lưỡng và đòi hỏi nhiều tài nguyên. Mô hình được huấn luyện trên bộ dữ liệu lớn và đa dạng, với quy trình được thiết kế riêng cho đặc thù của tiếng Việt.

**Dữ liệu huấn luyện**: PhoBERT sử dụng 20GB dữ liệu văn bản tiếng Việt cho quá trình huấn luyện, bao gồm 1GB từ Wikipedia tiếng Việt và 19GB dữ liệu từ một kho dữ liệu tin tức 50GB đã được lọc để loại bỏ các bài viết tương tự và trùng lặp. Việc sử dụng dữ liệu đa dạng và chất lượng cao là yếu tố quan trọng giúp PhoBERT có khả năng hiểu và xử lý ngôn ngữ tiếng Việt trong nhiều ngữ cảnh khác nhau.

**Tiền xử lý dữ liệu**: Trước khi huấn luyện, dữ liệu được tiền xử lý kỹ lưỡng. Các thẻ HTML bị loại bỏ, nội dung trùng lặp được lọc ra. Đặc biệt, toàn bộ dữ liệu văn bản được tách từ sử dụng công cụ RDRSegmenter, tạo ra khoảng 145 triệu câu đã được phân đoạn từ. Đây là bước quan trọng với tiếng Việt, giúp mô hình nắm bắt được cấu trúc từ ngữ chính xác trước khi áp dụng tokenization.

**Phương pháp huấn luyện**: PhoBERT được huấn luyện theo phương pháp của RoBERTa, cải tiến từ BERT. Mô hình sử dụng Masked Language Modeling (MLM) làm tác vụ huấn luyện chính, trong đó một phần từ ngữ bị che đi và mô hình phải dự đoán các từ này. Khác với BERT gốc, PhoBERT không sử dụng tác vụ Next Sentence Prediction (NSP), theo xu hướng của RoBERTa nhằm tập trung vào khả năng nắm bắt ngữ nghĩa tốt hơn.

**Cấu hình huấn luyện**: PhoBERT được huấn luyện trong 40 epochs với batch size động tối đa 8192 tokens. Thuật toán tối ưu hóa Adam được sử dụng với learning rate 1e-4 và warm-up 10K bước. Đối với PhoBERT-base, quá trình huấn luyện trải qua khoảng 540K training steps, trong khi PhoBERT-large là khoảng 1.08M steps, trên 4 GPU NVIDIA V100 (16GB mỗi GPU).

Tổng thời gian huấn luyện lên đến 3 tuần cho PhoBERT-base và 5 tuần cho PhoBERT-large, phản ánh khối lượng tính toán khổng lồ cần thiết để tạo ra mô hình ngôn ngữ tiền huấn luyện chất lượng cao

**Kiến trúc mô hình đa nhiệm vụ**: Một chiến lược hiệu quả là xây dựng mô hình đa nhiệm vụ (multi-task) dựa trên PhoBERT, có thể đồng thời xử lý cả nhận diện khía cạnh và phân loại cảm xúc. Kiến trúc phổ biến bao gồm việc sử dụng PhoBERT làm nền tảng, sau đó thêm các lớp đầu ra chuyên biệt cho từng khía cạnh và cảm xúc tương ứng. Mô hình end-to-end ABSA dựa trên PhoBERT cho phép tích hợp thông tin giữa các nhiệm vụ liên quan, giúp cải thiện hiệu suất chung. Kiến trúc này đặc biệt hữu ích trong môi trường thương mại điện tử, nơi việc hiểu chính xác từng khía cạnh và cảm xúc tương ứng mang lại giá trị kinh doanh lớn.

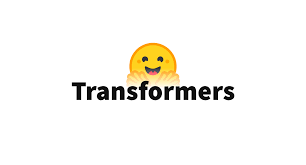
#### 2.1.5.3. Các thư viện và Framework sử dụng

* **PyTorch**



PyTorch là một thư viện học máy mã nguồn mở được phát triển bởi Facebook (Meta) AI Research. Nó cung cấp một nền tảng linh hoạt và hiệu quả để xây dựng các mô hình học sâu. Khác với các framework tĩnh truyền thống, PyTorch sử dụng phương pháp "define-by-run" cho phép thay đổi mạng neural động trong quá trình chạy, giúp quá trình gỡ lỗi và thử nghiệm trở nên dễ dàng hơn. Trong mã nguồn được cung cấp, PyTorch được sử dụng làm nền tảng chính để xây dựng, huấn luyện và đánh giá mô hình đa nhiệm vụ, với các module như torch.nn để định nghĩa cấu trúc mô hình, torch.optim để tối ưu hóa quá trình học, và torch.utils.data để quản lý dữ liệu.

* **Transformers**



Thư viện Transformers của Hugging Face được thiết kế để làm việc với các mô hình ngôn ngữ tiên tiến dựa trên kiến trúc Transformer. Ứng dụng sử dụng các class AutoModel và AutoTokenizer từ thư viện này để tải mô hình PhoBERT (một mô hình ngôn ngữ tiền huấn luyện cho tiếng Việt) và tokenizer tương ứng. Kiến trúc Transformer đã cách mạng hóa lĩnh vực xử lý ngôn ngữ tự nhiên với cơ chế self-attention, cho phép mô hình nắm bắt mối quan hệ giữa các từ trong câu bất kể khoảng cách giữa chúng. Transformers cung cấp nhiều mô hình tiền huấn luyện và API đơn giản để tinh chỉnh các mô hình này cho các nhiệm vụ cụ thể.

* **Scikit-Learn**



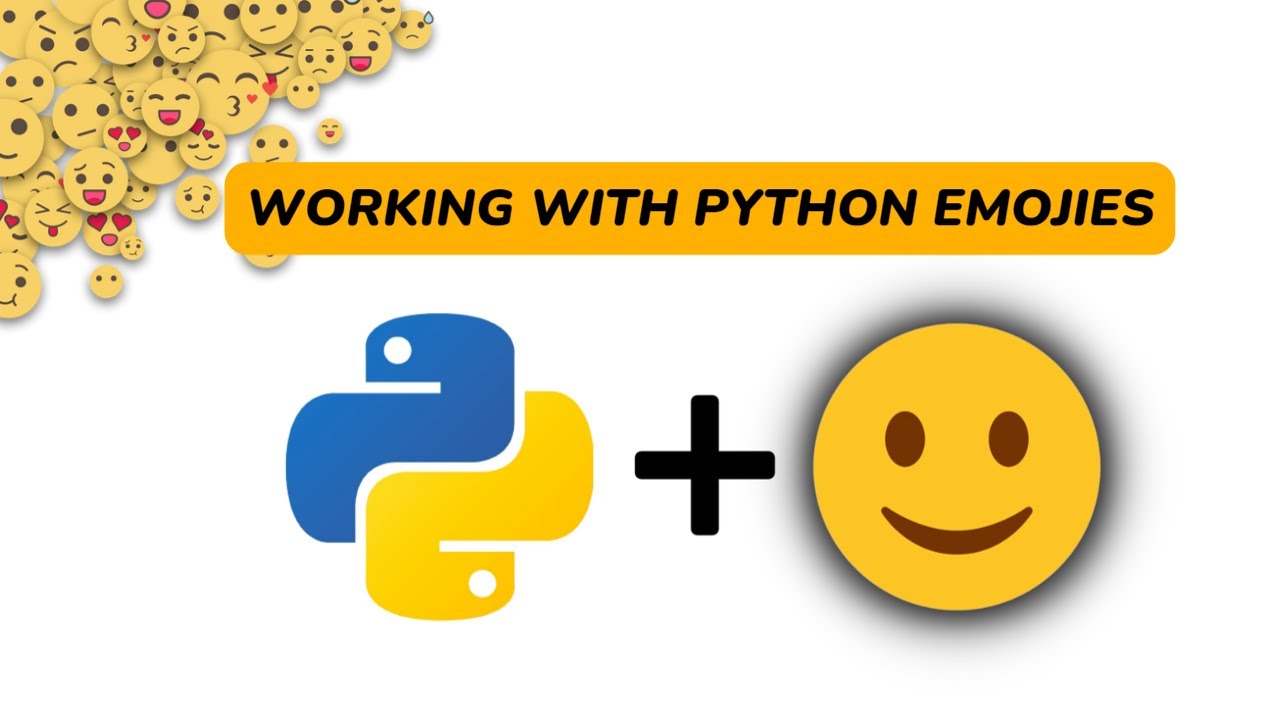
Scikit-learn là một thư viện học máy Python mã nguồn mở cung cấp nhiều thuật toán, công cụ đánh giá và tiện ích cho các tác vụ học có giám sát và không giám sát. Scikit-learn được sử dụng để phân chia dữ liệu (qua hàm train\_test\_split), tính toán các chỉ số đánh giá như F1-score và accuracy (qua module metrics), và tạo ma trận nhầm lẫn (confusion matrix) để đánh giá hiệu suất mô hình. Scikit-learn đặc biệt hữu ích nhờ giao diện nhất quán, tài liệu toàn diện và tính ổn định cao trong hệ sinh thái Python cho học máy.

* **Pandas**



Pandas là một thư viện phân tích và thao tác dữ liệu mạnh mẽ cung cấp cấu trúc dữ liệu như DataFrame và Series, giúp làm việc với dữ liệu dạng bảng một cách trực quan và hiệu quả. Pandas được sử dụng để đọc và xử lý dữ liệu từ file CSV, lọc và chuyển đổi dữ liệu, và tổ chức dữ liệu trước khi chuyển sang huấn luyện. DataFrame của Pandas giúp dễ dàng thực hiện các thao tác như lựa chọn cột, áp dụng hàm cho các giá trị, và thao tác với dữ liệu có nhãn.

* **Emoji**



Thư viện emoji được sử dụng để xử lý các biểu tượng cảm xúc (emoji) trong văn bản tiếng Việt. Phương thức emoji.demojize() được sử dụng để chuyển đổi emoji thành biểu diễn dạng văn bản, giúp chuẩn hóa dữ liệu đầu vào và tránh mất thông tin cảm xúc có thể được biểu thị qua emoji. Việc xử lý emoji là một khía cạnh quan trọng trong phân tích cảm xúc từ nội dung mạng xã hội và bình luận trực tuyến.

* **AdamW và Linear Schedule with Warmup**

AdamW là một biến thể của thuật toán tối ưu hóa Adam, thêm vào cải tiến về weight decay (giảm trọng số) để cải thiện khả năng tổng quát hóa của mô hình. Trong khi đó, Linear Schedule with Warmup là một lịch trình học (learning rate schedule) điều chỉnh tốc độ học theo thời gian: bắt đầu từ giá trị nhỏ, tăng dần trong giai đoạn "warmup", sau đó giảm tuyến tính. Cả hai kỹ thuật này đều là thực hành tốt trong huấn luyện mô hình Transformer, được triển khai trong mã nguồn thông qua AdamW và get\_linear\_schedule\_with\_warmup từ thư viện Transformers.

## 2.2. Kiến thức cơ sở về phát triển hệ thống Website.

### 2.2.1. JSON Web Token

#### 2.2.1.1. Khái niệm Json Web Token

JSON Web Mã (JWT) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA.

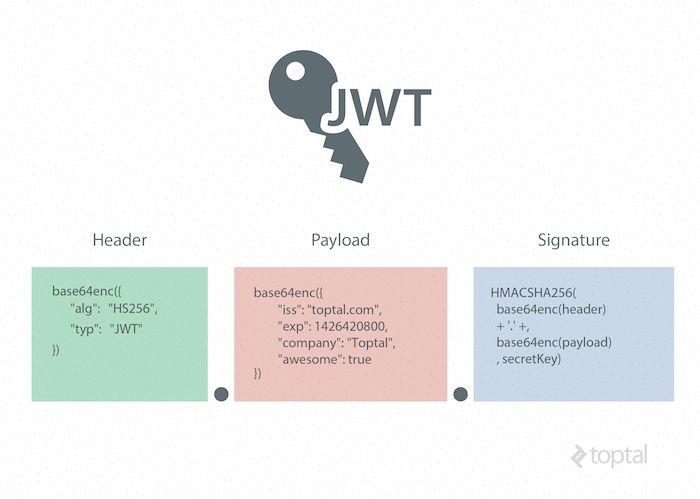
#### 2.2.1.2. Cấu trúc của một JWT

Dưới đây là 1 JSON Web Token: A group of letters and numbers

Description automatically generated

*Hình 2.1. Hình ảnh minh họa ví dụ JWT*

**JWT** trên bao gồm 3 phần: Header, Payload, Signature.



*Hình 2.2. Hình ảnh minh họa cấu trúc JWT*

* Header bao gồm hai phần chính:

typ – Loại token (mặc định là JWT – cho biết đây là một Token JWT)

alg – Thuật toán đã dùng để mã hóa (HMAC SHA256 – HS256 hoặc RSA).

* Payload: Là nơi chứa các nội dung của thông tin (claim). Thông tin truyền đi có thể là mô tả của 1 thực thể (ví dụ như người dùng) hoặc cũng có thể là các thông tin bổ sung thêm cho phần Header. Chúng được chia làm 3 loại: reserved, public và private.
* Signature: Phần chữ ký được tạo bằng cách kết hợp 2 phần Header + Payload, rồi mã hóa nó lại bằng 1 giải thuật encode bất kỳ ví dụ như HMAC SHA-256.

#### 2.2.1.3. Hệ thống sử dụng JWT

A screenshot of a computer

Description automatically generated

*Hình 2.3. Hình ảnh minh họa cách thức hoạt động của JWT*

Nhìn vào hình ta có thể thấy flow đi như sau:

1. User thực hiện login bằng cách gửi id/password hay sử dụng các tài khoản mạng xã hội lên phía Authentication Server (Server xác thực)
2. Authentication Server tiếp nhận các dữ liệu mà User gửi lên để phục vụ cho việc xác thực người dùng. Trong trường hợp thành công, Authentication Server sẽ tạo một JWT và trả về cho người dùng thông qua response.
3. Người dùng nhận được JWT do Authentication Server vừa mới trả về làm “chìa khóa” để thực hiện các “lệnh” tiếp theo đối với Application Server.
4. Application Server trước khi thực hiện yêu cầu được gọi từ phía User, sẽ verify JWT gửi lên. Nếu OK, tiếp tục thực hiện yêu cầu được gọi.

### 2.2.2. Restful API

#### 2.2.2.1. Restful API là gì?

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

Diễn giải các thành phần:

* REST(REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.
* API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

#### 2.2.2.2. Cách thức hoạt động

A diagram of a api

Description automatically generated

*Hình 2.7. Hình ảnh minh họa cách thức hoạt động của RESTful API*

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

* GET (SELECT): Trả về một Resource hoặc một danh sách Resource.
* POST (CREATE): Tạo mới một Resource.
* PUT (UPDATE): Cập nhật thông tin cho Resource.
* DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

#### 2.2.2.3. Ưu điểm của RESTful API

RESTful API mang đến rất nhiều lợi ích và những hiệu quả nhất định cho các lập trình viên. Dưới đây là một vài ưu điểm nổi bật của RESTful API:

* RESTful API giúp cho ứng dụng rõ ràng và dễ nhìn hơn.
* RESTful API giúp cho các dữ liệu được trả về dưới nhiều định dạng khác nhau như XML, HTML, JSON…
* RESTful API cho phép sử dụng các lệnh call thủ tục HTTP tiêu chuẩn để truy xuất dữ liệu và request.
* Code của REST API đơn giản và ngắn gọn.
* RESTful API dựa trên code và có thể dùng nó để đòng bộ hóa dữ liệu với website.
* REST chú trọng vào nhiều tài nguyên của hệ thống.

Rất nhiều trang web hiện nay đang sử dụng REST API để cho phép các ứng dụng bên ngoài kết nối dữ liệu với họ dễ dàng hơn.

### 2.2.3. ReactJs (Frontend)

Để xây dựng Website cũng như các ứng dụng nói chung cần có 2 thành phần chính là Front-end và Back-end, trong đó Front-end là việc xây dựng các hiển thị giao diện, tương tác phía người dùng. Trong đó không thể không kể đến các các ngôn ngữ chính soạn thảo siêu văn bản là HTML, CSS và ngôn ngữ lập trình JavaScript. Với sự phát triển tiến bộ của ngành công nghệ thông tin, các thư viện hỗ trợ, framework ra đời giúp cho quá trình xây dựng website, các ứng dụng trở nên dễ dàng hơn.

#### 2.2.3.1. Javascript

**JavaScript** là một **ngôn ngữ lập trình thông dịch** với khả năng hướng đến đối tượng. Là một trong 3 ngôn ngữ chính trong lập trình web và có mối liên hệ lẫn nhau để xây dựng một website sống động, chuyên nghiệp:

* **HTML**: Hỗ trợ trong việc xây dựng layout, thêm nội dung dễ dàng trên website.
* **CSS**: Hỗ trợ việc định dạng thiết kế, bố cục, style, màu sắc,…
* **JavaScript**: Tạo nên những nội dung “động” trên website. Cùng tìm hiểu rõ hơn ở phần dưới đây.

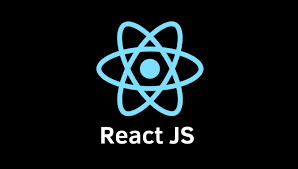
**JS** là viết tắt của **JavaScript**, khi có JS bạn sẽ hiểu đó đang nói đến JavaScript.

Nhiệm vụ của Javascript là xử lý những đối tượng HTML trên trình duyệt. Nó có thể can thiệp với các hành động như thêm / xóa / sửa các thuộc tính CSS và các thẻ HTML một cách dễ dàng. Hay nói cách khác, Javascript là một ngôn ngữ lập trình trên trình duyệt ở phía client. Tuy nhiên, hiện nay với sự xuất hiện của NodeJS đã giúp cho Javascript có thể làm việc ở backend.

#### 2.2.3.2. JavaScript Framework là gì?

**JavaScript Framework** là thư viện được xây dựng dựa vào ngôn ngữ lập trình JavaScript. Từ đó, mỗi framework được tạo ra để phục cho từng lĩnh vực khác nhau. Bạn có thể tìm hiểu kỹ hơn về framework là gì, sẽ giúp bạn có thêm nhiều thông tin rõ ràng hơn. Hiện nay, có rất nhiều JavaScript Framework thông dụng như: **ReactJs, NodeJs, Angular, …**

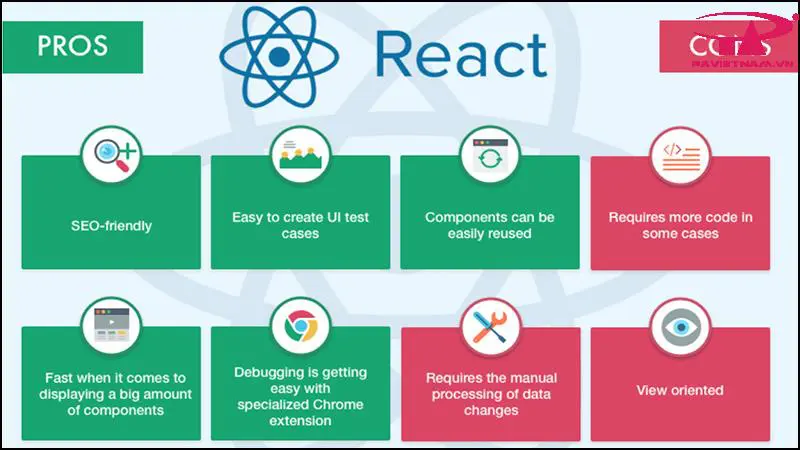
Đề tài này em đã sử dụng **ReactJs** để xây dựng chương trình, **ReactJs** là một framework mạnh mẽ được phát triển bởi Facebook, ra mắt vào năm 2013. **React** (Reactjs hay React.js) là một thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng hoặc các thành phần UI. Được tạo ra bởi sự cộng tác giữa Facebook và Instagram. Nó được duy trì bởi Facebook và một cộng đồng các nhà phát triển và công ty cá nhân. React có thể được sử dụng như một cơ sở để phát triển các ứng dụng trang đơn hoặc di động. Một trong những đặc trưng duy nhất của React là việc render dữ liệu không những có thể thực hiện ở tầng Server mà còn ở tầng Client.

**

*Hình 2.10. Hình ảnh minh họa logo Framework ReactJS*

Sơ lược qua các ưu điểm của framework này, ta có các ưu điểm đáng chú ý như sau:

* **Dễ dàng sử dụng, tạo được các component nhẹ**: React Component dễ viết hơn vì nó sử dụng JSX, mở rộng cú pháp tùy chọn cho JavaScript cho phép bạn kết hợp HTML với JavaScript. React cung cấp việc tạo component nhẹ, các thành phần phi trạng thái rất dễ dàng.
* **API thanh lịch**: Khuyến khích bạn nắm lấy bố cục bằng các component.
* **Hỗ trợ cộng đồng lớn**:React rất phổ biến, minh chứng rõ ràng bởi cộng đồng hỗ trợ lớn của nó.
* **Phổ biến trong giới StartUp**: Mức độ phổ biến của React đã giúp thúc đẩy sự phát triển của các công ty khởi nghiệp.
* **Nhiều tiện ích nguồn mở**: Tính khả dụng của một loạt các tiện ích mở rộng do cộng đồng nguồn mở phát triển cho React cung cấp cho bạn rất nhiều tùy chọn để xây dựng các giải pháp hoàn chỉnh.
* **Thân thiện với SEO**: Hầu như các JS Frameworks không thân thiện với các tìm kiếm mặc dù đã được cải thiện nhiều nhưng dưới sự hỗ trợ của các render dữ liệu trả về dưới dạng web page giúp cho SEO chuẩn hơn.

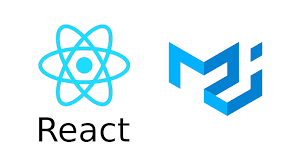
**

*Hình 2.11. Hình ảnh minh họa các ưu điểm của ReactJS*

Tuy nhiên, ReactJs cũng có những nhược điểm nhất định:

* **Điều chỉnh cho JSX**: Các Component không phải HTML nguyên bản mà được viết bằng JSX. Vậy nên phải mất thời gian để chờ đợi đội ngũ phát triển React điều chỉnh.
* **Các giải pháp hoàn chỉnh yêu cầu thư viện của bên thứ ba**: Do tập trung vào việc xây dựng giao diện người dùng, bạn có thể cần phải mở rộng React với các thư viện của bên thứ ba.
* **Ví dụ**: Nếu bạn cần hỗ trợ định tuyến phía máy khách trong ứng dụng của mình, bạn có thể sử dụng thư viện React Router của bên thứ ba.
* **Reactjs chỉ phục vụ cho tầng View**: React chỉ là View Library, không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax.
* **Tính khả dụng của các tùy chọn có thể gây nhầm lẫn**: Có nhiều sự lựa chọn có thể gây nhầm lẫn – đặc biệt đối với người mới bắt đầu.

#### 2.2.3.3. Material UI



Material UI là một thư viện giao diện người dùng (UI) mã nguồn mở, được sử dụng phổ biến trong các dự án phát triển ứng dụng web với React. Thư viện này cung cấp các thành phần giao diện (components) được thiết kế theo chuẩn Material Design của Google, giúp xây dựng các giao diện đẹp, hiện đại, và thân thiện với người dùng.

Đặc điểm nổi bật của Material UI:

* **Tính mô-đun**: Material UI cung cấp các thành phần giao diện độc lập như Button, TextField, Dialog, Table, v.v., cho phép nhà phát triển dễ dàng tích hợp và tùy chỉnh theo nhu cầu.
* **Tùy chỉnh dễ dàng**: Thư viện hỗ trợ hệ thống theme linh hoạt, giúp thay đổi màu sắc, kiểu chữ, và các thuộc tính giao diện thông qua các đối tượng cấu hình hoặc CSS-in-JS.
* **Hỗ trợ responsive**: Các thành phần của Material UI được tối ưu hóa cho các thiết bị khác nhau, đảm bảo giao diện hoạt động tốt trên cả máy tính và thiết bị di động.
* **Cộng đồng mạnh mẽ**: Là một dự án mã nguồn mở, Material UI có cộng đồng đông đảo, tài liệu phong phú, và được cập nhật thường xuyên.

### 2.2.4. Java Spring Boot (Backend)

#### 2.2.4.1. Giới thiệu về Back-end

Backend là những chức năng hỗ trợ hoạt động của một trang web hoặc ứng dụng mà người dùng không nhìn thấy được (hoặc có thể ví như đây là phần chìm của tảng băng). Nó có cơ chế hoạt động gần giống như bộ não của con người, xử lý các yêu cầu, lệnh và chọn thông tin thích hợp để hiển thị trên màn hình. BackEnd của một trang web bao gồm ba thành phần: máy chủ, ứng dụng và cơ sở dữ liệu. Điều này cho phép trang web hoạt động hiệu quả và cung cấp cho người dùng thông tin chính xác nhanh chóng.

Một trang web sẽ chứa một hoặc nhiều tập lệnh được chạy trên máy chủ mỗi khi truy cập vào website. Mọi hoạt động hiển thị trên trình duyệt web có sự đóng góp một phần Backend. Quy trình của Backend bao gồm:

* Xử lý các yêu cầu của web đến.
* Chạy tập lệnh như (JSP,ASP, PHP,…) để tạo ra HTML.
* Truy cập vào dữ liệu từ cơ sở dữ liệu bằng sử dụng truy vấn SQL.
* Lưu trữ và cập nhật hồ sơ có trong cơ sở dữ liệu.
* Giải mã và mã hóa dữ liệu.
* Xử lý các dữ liệu tệp tải lên và tải xuống.
* Xử lý người dùng bằng JavaScript.

Trong phạm vi đề tài, nhóm chúng em sử dụng Java Spring Boot để xây dựng backend cho chương trình.

#### 2.2.4.2. Java Spring Boot

Spring Boot là một trong số các module của Spring framework chuyên cung cấp các tính năng RAD (Rapid Application Development) cho phép tạo ra và phát triển các ứng dụng độc lập dựa trên Spring một cách nhanh chóng.

Spring Boot ra đời với mục đích loại bỏ những cấu hình phức tạp của Spring, nó không yêu cầu cấu hình XML và nâng cao năng suất cho các nhà phát triển. Với sự góp mặt của Spring Boot, hệ sinh thái Spring đã trở nên mạnh mẽ, phổ biến và hiệu quả hơn bao giờ hết.



*Hình 2.14. Hình ảnh minh họa java spring boot*

**Ưu điểm của Spring Boot:**

Với mục đích ra đời rất rõ ràng của mình, Spring Boot đã khắc phục được những hạn chế về cấu hình của Spring. Dưới đây, Stringee sẽ giới thiệu thêm đến bạn một số lợi ích của Spring Boot.

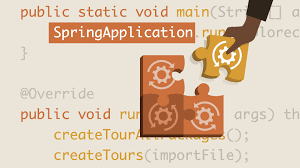
Những ưu điểm đó bao gồm:

* Hội tụ đầy đủ các tính năng của Spring framework.
* Đơn giản hóa cấu hình và xây dựng được các ứng dụng độc lập có khả năng chạy bằng “java -jar” nhờ các dependency starter.
* Dễ dàng deploy vì các ứng dụng server được nhúng trực tiếp vào ứng dụng để tránh những khó khăn khi triển khai lên môi trường production mà không cần thiết phải tải file WAR.
* Cấu hình ít, tự động hỗ trợ bất cứ lúc nào cho chức năng giống với Spring như tăng năng suất, giảm thời gian viết code và không yêu cầu XML config.
* Cung cấp nhiều plugin, số liệu, cấu hình ứng dụng từ bên ngoài.

**Các đặc tính cơ bản của Spring Boot:**

Vì là một framework RAD nên Spring Boot mang trong nó nhiều đặc tính nổi trội phục vụ cho việc phát triển và cài đặt nhanh một ứng dụng chạy trên Java.

Đặc tính đầu tiên có thể kể đến đó là Spring Boot cung cấp sẵn cho chúng ta một lớp có c và được hoạch định làm điểm mở đầu cho toàn bộ chương trình. Lớp này được đặt tên là Spring Application, nó giúp khởi chạy các ứng dụng từ hàm main, khi chạy nó chúng ta chỉ cần gọi phương thức run.

**

*Hình 2.15. Hình ảnh minh họa đặc tính cơ bản của spring boot*

Spring Boot giúp giảm bớt độ phức tạp trong việc cấu hình ứng dụng trong trường hợp chúng ta sử dụng nhiều môi trường. Với Profiles, Spring Boot cung cấp cho người sử dụng một cách phân chia cấu hình cho từng môi trường. Các bên thực hiện việc cấu hình ứng dụng hoàn toàn có thể được đánh dấu profiles để giới hạn thời điểm hay môi trường mà nó sẽ được tải các cấu hình lên ứng dụng.

Externalized Configurations: Externalized Configuration cho phép bạn có khả năng cấu hình được từ bên ngoài. Vì vậy, một ứng dụng được xây dựng có thể được vận hành và hoạt động trên nhiều môi trường khác nhau. Để thực hiện Externalized Configuration bạn có thể sử dụng các file properties, YAML, các tham số command line hay các biến môi trường.

Đặc tính cuối cùng mà bạn nên biết đó là tất cả các tính năng log nội bộ của Spring Boot đều sử dụng common logging. Chúng được quản lý một cách mặc định, vì vậy bạn không nên sửa các dependency logging nếu không được yêu cầu.

# CHƯƠNG 3: THỰC NGHIỆM ĐỀ TÀI

## 3.1. Giới thiệu ứng dụng

Ứng dụng “***SentiScope***” giúp người bán hàng có góc nhìn sâu sắc hơn về các bình luận đánh giá của khách hàng về sản phẩm của họ. Không chỉ dừng lại ở việc nhận diện cảm xúc mà còn phân loại các đánh giá vào các chủ đề phổ biến về sản phẩm như chất lượng, giá cả, đóng gói, giao hàng, … góp phần giúp người bán hàng có những quyết định kinh doanh, cải thiện chất lượng sản phẩm, dịch vụ để tăng doanh số.

**Tính Năng Chính của Ứng Dụng:** Ứng dụng mang đến những tính năng chính vô cùng hữu ích cho những người bán hàng trên các nền tảng mạng xã hội và các sàn thương mại điện tử. Các chức năng chính bao gồm:

* **Thu thập các bình luận đánh giá:**  trên nhiều nền tảng được hỗ trợ, tổng hợp giúp người bán hàng cho cái nhìn tổng quát về các sản phẩm của mình trên các nền tảng và từng nền tảng.
* **Phân loại cảm xúc và chủ đề**: Các bình luận sẽ được gán nhãn cảm xúc (tiêu cực, trung tính, tích cực) và chủ đề (chất lượng, giá cả, đóng gói, giao hàng, …).
* **Thống kê và trực quan hóa**: Thống kê tỉ lệ % các cảm xúc của sản phẩm theo thời gian, theo tổng số các bình luận. Thống kê tỉ lệ % các chủ đề mà khách hàng quan tâm về sản phẩm. Thống kê tỉ lệ % các nhãn cảm xúc của từng chủ đề. Trực quan hóa các dữ liệu thống kê trên các biểu đồ giúp người bán hàng dễ hình dung và có cái nhìn bao quát.

## 3.2. Fine-tuning mô hình PhoBERT

### 3.2.1. Xây dựng bộ dữ liệu

Với nội dung đã tìm hiểu về chủ đề phản hồi, đánh giá của khách hàng, bộ

dữ liệu của em được thu thập từ các trang bán hàng trực tuyến. Ở đây em sử dụng kết hợp 2 tập dữ liệu:

* 1. **Vietnamese Sentiment Analysis (30635 bản ghi) (label: POS, NEG, NEU).**

A screenshot of a computer

AI-generated content may be incorrect.

* 1. **AIVIVN-2019(12870 bản ghi) (label 1-0)**

A screenshot of a computer

AI-generated content may be incorrect.

* 1. **Customize Dataset**
* Chuẩn hóa các nhãn cảm xúc bao gồm 3 nhãn (tiêu cực, trung tính và tích cực).
* Thêm cột nhãn chỉ các khía cạnh quan tâm của lĩnh cực thương mại điện tử bao gồm chất lượng, đóng gói, giao hàng, giá thành, Hỗ trợ khách hàng, Đặt hàng, Trả hàng, Khuyến mãi, Khác.
* Sử dụng công nghệ Selenium và Model AI của Copilot Microsoft kết hợp kỹ thuật viết Promp để cho ra Output như mong đợi.

A screenshot of a computer

AI-generated content may be incorrect.

Sử dụng Model AI và kỹ thuật viết Promp nên đầu ra sẽ có một số bất thường. Trong những trường hợp này mặc định trả về label 1 và task 9 nên chúng tôi lọc những bản ghi có lable 1 và task 9 sau đó tiếp tục thực hiện kỹ thuật trên để lấy các nhãn chuẩn hơn. Sau 3 lần chúng tôi có 1 tập dữ liệu tương đối:

Bộ dữ liệu được xây dựng thành 1 file dataset\_sentiment\_analysis.csv. Mỗi dòng dữ liệu có 4 cột chính:

* id: Đánh dấu thứ tự Comment trong tập Dataset
* comment: là các phản hồi, đánh giá sản phẩm của khách hàng.
* lable: là nhãn cảm xúc (0. Tiêu cực, 1. Trung tính, 2. Tích cực)
* task: là nhãn chủ đề phân loại trong lĩnh vực thương mại điện tử (1. chất lượng, 2. đóng gói, 3. giao hàng, 4. giá thành, 5. Hỗ trợ khách hàng, 6. Đặt hàng, 7. Trả hàng, 8. Khuyến mãi, 9. Khác)
* Tổng có 40618 bản ghi.
  1. **Làm sạch bộ dữ liệu**

Tiền xử lý dữ liệu là một trong những bước quan trọng nhất khi giải quyết bất kỳ bài toán nào trong lĩnh vực Học máy. Để mô hình có thể đưa ra kết quả có độ chính xác cao thì bộ dữ liệu luôn cần được xử lý, làm sạch và biến đổi trước khi trở thành dữ liệu huấn luyện cho mô hình học máy.

* Xóa bỏ các dòng có khoảng trắng.
* Xóa bỏ các dòng trùng lặp.

### 3.2.2. Fine-tuning mô hình

A multicolored rectangular box with text

AI-generated content may be incorrect.

Quy trình huấn luyện mô hình PhoBERT đa nhiệm vụ được thực hiện qua các bước chặt chẽ như sau. Đầu tiên, dữ liệu thô được tải lên và xử lý tiền xử lý bao gồm chuẩn hóa emoji, loại bỏ ký tự đặc biệt, chuyển về chữ thường và thay thế từ viết tắt dựa trên danh sách cung cấp. Sau khi làm sạch, dữ liệu được phân chia thành ba tập train, validation và test với tỷ lệ lần lượt là 70%, 15% và 15%, đồng thời đảm bảo phân phối đồng đều theo cả hai nhãn cảm xúc và chủ đề (stratify theo label và task). Văn bản sau xử lý được mã hóa bằng tokenizer của PhoBERT và đóng gói thành dataset phục vụ cho DataLoader.

Mô hình sử dụng là MultiTaskModel, kế thừa PhoBERT-base, với hai nhánh phân loại song song cho hai nhiệm vụ: cảm xúc (3 lớp) và chủ đề (9 lớp). Hàm mất mát tổng hợp được xây dựng từ hai thành phần cross-entropy, với trọng số điều chỉnh α=0.45 cho cảm xúc và β=0.55 cho chủ đề. Quá trình huấn luyện diễn ra trong 5 epoch, tối ưu hóa bằng AdamW (learning rate 2e-5, weight decay 0.01) kết hợp scheduler linear warmup (10% tổng số bước). Sau mỗi 100 bước, mô hình được đánh giá trên tập validation, tính toán F1-score cho cả hai nhiệm vụ và lưu lại trạng thái tốt nhất dựa trên trung bình F1. Cơ chế early stopping với patience=3 được áp dụng để tránh overfitting. Cuối cùng, mô hình tốt nhất được nạp lại và đánh giá trên tập test, thu thập các chỉ số accuracy, F1-score, classification report và ma trận nhầm lẫn cho cả hai nhiệm vụ, đồng thời trực quan hóa kết quả để phục vụ phân tích sâu hơn.

#### 3.2.2.1. Tải và làm sạch dữ liệu

* Tải dữ liệu từ tệp CSV với các cột bắt buộc: 'id', 'comment', 'label', 'task'
* Xác thực dữ liệu đầu vào để đảm bảo nhãn cảm xúc là 0, 1, 2 và nhãn chủ đề từ 1 đến 9
* Ánh xạ nhãn vào các tên có ý nghĩa:
  + Sentiment: {0: "Tiêu cực", 1: "Trung tính", 2: "Tích cực"}
  + Task: {1: "Chất lượng", 2: "Đóng gói", 3: "Giao hàng", 4: "Giá thành", 5: "Hỗ trợ khách hàng", 6: "Đặt hàng", 7: "Trả hàng", 8: "Khuyến mãi", 9: "Khác"}

#### 3.2.2.2. Xử lý văn bản

* Triển khai hàm preprocess\_text() để chuẩn hóa văn bản:
* Chuyển emojis thành văn bản sử dụng thư viện emoji
* Loại bỏ ký tự đặc biệt và khoảng trắng dư thừa
* Chuyển đổi sang chữ thường
* Chuẩn hóa từ viết tắt (nếu có danh sách)
* Tạo cột 'processed\_text' chứa văn bản đã qua xử lý

#### 3.2.2.3. Tách dữ liệu thành tập huấn luyện, xác thực và kiểm tra

A screenshot of a computer code

AI-generated content may be incorrect.

* Sử dụng train\_test\_split để chia dữ liệu theo tỷ lệ:
  + 70% huấn luyện (train)
  + 15% xác thực (validation)
  + 15% kiểm tra (test)
* Đảm bảo phân tầng (stratify) dữ liệu dựa trên cả hai nhãn để duy trì phân phối

**3.2.2.4. Tạo lớp MultiTaskDataset cho PyTorch**

**A screen shot of a computer program

AI-generated content may be incorrect.**

* Kế thừa từ lớp Dataset của PyTorch để tạo lớp tùy chỉnh MultiTaskDataset
* Xử lý cả hai loại nhãn (cảm xúc và chủ đề)
* Chuyển đổi nhãn chủ đề từ dải 1-9 sang 0-8 để phù hợp với chỉ số tensor
* Mã hóa văn bản sử dụng tokenizer của PhoBERT với padding và truncation

#### 3.2.2.5. Tạo DataLoader

* Khởi tạo DataLoader cho các tập dữ liệu train, validation và test
* Đặt batch\_size và shuffle phù hợp (batch\_size=16, shuffle=True cho tập train)

A screenshot of a computer code

AI-generated content may be incorrect.

#### 3.2.2.6. Thiết kế kiến trúc MultiTaskModel

A computer screen shot of a code

AI-generated content may be incorrect.

Tạo lớp MultiTaskModel kế thừa từ nn.Module của PyTorch

Sử dụng PhoBERT làm mô hình nền:

* Thêm hai đầu phân loại tuyến tính cho hai nhiệm vụ:
  + Đầu phân loại cảm xúc***: self.sentiment\_classifier = nn.Linear(hidden\_size, num\_sentiment\_labels)***
  + Đầu phân loại chủ đề: ***self.task\_classifier = nn.Linear(hidden\_size, num\_task\_labels)***
* Xác định trọng số cho hàm mất mát kết hợp (alpha cho sentiment, beta cho task)

## 3.3. Mô hình hóa dữ liệu

### 3.3.1. Các yêu cầu về dữ liệu

Website cần lưu thông tin về:

* **Thông tin user:** gồm các thông tin cơ bản như username, password, email, avatar. Quyền với ứng dụng và chức năng settings như role, receive\_notification. Thông tin trạng thái tài khoản: create\_at, modify\_at, is\_active.
* **Thông tin danh mục:** gồm các thông tin cơ bản về danh mục như name và trạng thái như create\_at, modify\_at, is\_active. Mỗi user có thể quản lý nhiều danh mục, mỗi danh mục thuộc 1 user.
* **Thông tin về nền tảng**: gồm các thông tin cơ bản về danh mục như name và trạng thái như name, create\_at, modify\_at, is\_active.
* **Thông tin về sản phẩm:** gồm các thông tin cơ bản về danh mục như name, sku, description và trạng thái như name, create\_at, modify\_at, is\_active. Mỗi sản phẩm thuộc về 1 danh mục. Mỗi danh mục có thể có nhiều sản phẩm. Mỗi sản phẩm được bán trên nhiều nền tảng và mỗi nền tảng có nhiều sản phẩm được bán trên đó. Mỗi user bán được nhiều sản phẩm trên nhiều nền tảng.
* **Thông tin về bình luận đánh giá**: bao gồm các thông tin cơ bản như nội dung (content), nhãn (sentiment) và tỉ lệ đúng (sentiment\_confidence) cảm xúc, nhãn (task) và tỉ lệ đúng (task\_confidence) chủ đề. Mỗi bình luận thuộc về 1 sản phẩm được bán trên nền tảng cụ thể.

### 3.3.2. Thiết kế bảng

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

### 3.3.3. Biểu đồ thực thể liên kết mức vật lý

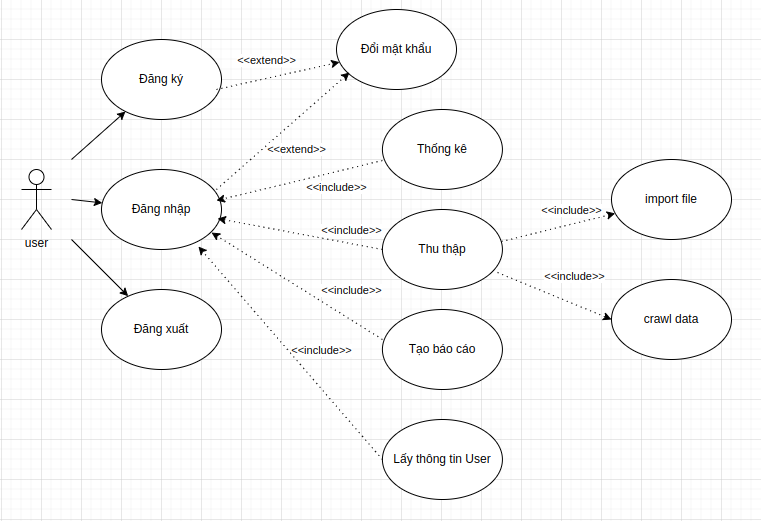
A screenshot of a computer screen

AI-generated content may be incorrect.

*Hình 3.10. Biểu đồ thực thể liên kết mức vật lý*

## 3.4. Mô hình hóa chức năng

### 3.4.1. Biểu đồ UseCase



### 3.4.2. Đặc tả chi tiết các UseCase

#### 3.4.2.1. Đặc tả UseCase Đăng nhập

|  |  |  |
| --- | --- | --- |
| **Tên UseCase** | | **Đăng nhập** |
| Mô tả vắn tắt | | Use case này cho phép người dùng đăng nhập vào hệ thống để truy cập các chức năng và thông tin cá nhân của mình. |
| Luồng sự kiện | Luồng cơ bản | Bước 1: Use case này bắt đầu khi người dùng muốn đăng nhập vào hệ thống.  Bước 2: Người dùng nhập tên đăng nhập và mật khẩu vào các trường tương ứng trên màn hình đăng nhập.  Bước 3: Người dùng kích vào nút "Đăng nhập". Hệ thống kiểm tra thông tin đăng nhập.  Bước 4: Nếu thông tin đăng nhập đúng, hệ thống chuyển người dùng đến màn hình chính của ứng dụng và use case kết thúc. Nếu thông tin không chính xác, hệ thống đưa ra thông báo lỗi và yêu cầu người dùng nhập lại. |
| Các luồng rẽ nhánh | Tại bước 2 ở luồng cơ bản, nếu người dùng kích vào "Quên mật khẩu", hệ thống chuyển người dùng đến màn hình khôi phục mật khẩu và use case kết thúc.  Tại bước 3 ở luồng cơ bản, nếu người dùng kích vào "Đăng ký", hệ thống chuyển người dùng đến màn hình đăng ký mới và use case kết thúc.  Trong tất cả các bước tại luồng cơ bản, nếu hệ thống không thể kết nối được với CSDL, hiển thị ra màn hình thông báo lỗi. Use case kết thúc. |
| Các yêu cầu đặc biệt | | Không có. |
| Tiền điều kiện | | Người dùng cần phải có tài khoản đã được đăng ký trước đó để đăng nhập. |
| Hậu điều kiện | | Nếu use case kết thúc thành công, người dùng sẽ được chuyển đến màn hình chính và thông tin đăng nhập của họ được lưu lại trong phiên làm việc hiện tại. |
| Điểm mở rộng | | Không có |

A diagram of a login

AI-generated content may be incorrect.

#### 3.4.2.2. Đặc tả UseCase Đăng ký

|  |  |  |
| --- | --- | --- |
| **Tên UseCase** | | **Đăng ký** |
| Mô tả vắn tắt | | Use case này cho phép người dùng tạo một tài khoản mới trong hệ thống để có thể sử dụng các chức năng và tính năng của ứng dụng. |
| Luồng sự kiện | Luồng cơ bản | Bước 1: Use case này bắt đầu khi người dùng muốn đăng ký tài khoản mới.  Bước 2: Người dùng nhập thông tin cần thiết như tên, địa chỉ email, mật khẩu, và các thông tin khác liên quan vào các trường trên màn hình đăng ký.  Bước 3: Người dùng kích vào nút "Đăng ký". Hệ thống kiểm tra tính hợp lệ của thông tin và sự sẵn có của địa chỉ email.  Bước 4: Nếu thông tin đăng ký hợp lệ, hệ thống tạo tài khoản mới và chuyển người dùng đến màn hình chính của ứng dụng. Use case kết thúc. Nếu có lỗi hoặc thông tin không hợp lệ, hệ thống đưa ra thông báo lỗi và yêu cầu người dùng nhập lại. |
| Các luồng rẽ nhánh | Tại bước 2 ở luồng cơ bản, nếu người dùng kích vào "Quay lại", hệ thống chuyển người dùng đến màn hình đăng nhập và use case kết thúc.  Tại bước 3 ở luồng cơ bản, nếu người dùng nhập email đã tồn tại, hệ thống đưa ra thông báo lỗi và yêu cầu người dùng nhập lại thông tin.  Trong tất cả các bước tại luồng cơ bản, nếu hệ thống không thể kết nối được với CSDL, hiển thị ra màn hình thông báo "Hệ thống không thể kết nối được với dữ liệu". Use case kết thúc. |
| Các yêu cầu đặc biệt | | Mật khẩu phải đáp ứng các yêu cầu bảo mật cụ thể, và email phải có định dạng hợp lệ. |
| Tiền điều kiện | | Người dùng không được có tài khoản đã đăng ký với địa chỉ email đã nhập trước đó. |
| Hậu điều kiện | | Nếu use case kết thúc thành công, tài khoản mới sẽ được tạo và thông tin đăng ký sẽ được lưu vào CSDL.. |
| Điểm mở rộng | | Không có |

A diagram of a sequence register

AI-generated content may be incorrect.

#### 3.4.2.3. Đặc tả UseCase Đăng xuất

|  |  |  |
| --- | --- | --- |
| **Tên UseCase** | | **Đăng xuất** |
| Mô tả vắn tắt | | Use case này cho phép người dùng đăng xuất khỏi hệ thống để kết thúc phiên làm việc và bảo vệ thông tin cá nhân. |
| Luồng sự kiện | Luồng cơ bản | Bước 1: Use case này bắt đầu khi người dùng muốn đăng xuất khỏi hệ thống.  Bước 2: Người dùng kích vào nút "Đăng xuất" hoặc tương tự trên giao diện người dùng.  Bước 3: Hệ thống xác nhận việc đăng xuất và chuyển người dùng đến màn hình đăng nhập hoặc màn hình chào mừng không đăng nhập. Use case kết thúc. |
| Các luồng rẽ nhánh | Tại bước 2 ở luồng cơ bản, nếu người dùng chọn "Hủy bỏ" hoặc không thực hiện bất kỳ hành động nào, hệ thống giữ người dùng ở màn hình hiện tại và use case kết thúc.  Trong tất cả các bước, nếu hệ thống không thể kết nối được với CSDL, hiển thị ra màn hình thông báo "Hệ thống không thể kết nối được với dữ liệu". Use case kết thúc. |
| Các yêu cầu đặc biệt | | Không có |
| Tiền điều kiện | | Người dùng đã đăng nhập vào hệ thống. |
| Hậu điều kiện | | Nếu use case kết thúc thành công, người dùng sẽ được chuyển đến màn hình đăng nhập hoặc màn hình chào mừng không đăng nhập, và thông tin đăng nhập của họ sẽ không còn hiệu lực trong phiên làm việc hiện tại. |
| Điểm mở rộng | | Không có |

A diagram of a sequence of a log out

AI-generated content may be incorrect.

#### 3.4.2.4. *Đặc tả UseCase đổi mật khẩu*

|  |  |  |
| --- | --- | --- |
| **Tên UseCase** | | **Quên mật khẩu** |
| Mô tả vắn tắt | | Use case này cho phép người dùng đặt lại mật khẩu khi quên. Sau khi điền chính xác thông tin tài khoản, hệ thống xác minh và gửi email để người dùng có thể đặt lại mật khẩu |
| Luồng sự kiện | Luồng cơ bản | Luồng cơ bản:  Bước 1: Use case này bắt đầu khi người dùng quên mật khẩu và chọn chức năng "Quên mật khẩu" trên giao diện đăng nhập.  Bước 2: Hệ thống hiển thị form nhập email đăng ký.  Bước 3: Người dùng nhập email và xác nhận.  Bước 4: Hệ thống kiểm tra email.  Bước 5: Nếu email hợp lệ, hệ thống gửi email chứa link đặt lại mật khẩu.  Bước 6: Người dùng truy cập link trong email, nhập mật khẩu mới và xác nhận.  Bước 7: Hệ thống cập nhật mật khẩu mới, thông báo thành công và chuyển về trang đăng nhập |
| Các luồng rẽ nhánh | - Nếu email không tồn tại hoặc không hợp lệ, hệ thống thông báo lỗi và yêu cầu nhập lại.  - Nếu người dùng không hoàn thành các bước, hệ thống hủy quá trình đặt lại mật khẩu. |
| Các yêu cầu đặc biệt | | Không có |
| Tiền điều kiện | | Người dùng đã có tài khoản và truy cập trang đăng nhập |
| Hậu điều kiện | | Người dùng có thể đăng nhập bằng mật khẩu mới hoặc quá trình đặt lại mật khẩu bị hủy nếu không hoàn thành. |
| Điểm mở rộng | | Không có |

*A screenshot of a computer screen

AI-generated content may be incorrect.*

#### 3.4.2.5*. Đặc tả UseCase Thống kê*

|  |  |  |
| --- | --- | --- |
| **Tên UseCase** | | **Thống kê** |
| Mô tả vắn tắt | | Use case này cho phép người dùng xem các biểu đồ thống kê bao gồm: bảng thống kê chung, biểu đồ tỉ lệ cảm xúc theo thời gian, biểu đồ tỉ lệ các nhãn chủ đề trên tổng số, biểu đồ tỉ lệ các nhãn cảm xúc theo từng chủ đề. |
| Luồng sự kiện | Luồng cơ bản | Bước 1: Người dùng nhấn vào nút đăng nhập và đăng nhập thành công.  Bước 2: Hệ thống sẽ tự động vào trang thống kê, gọi các API từ phía Backend và Frontend hiện thị các loại biểu đồ cho người dùng. |
| Các luồng rẽ nhánh | Tại bước 2 của luồng cơ bản: Nếu gọi các API ở phía Backend xảy ra lỗi thì sẽ hiển thị thông báo cho người dùng. |
| Các yêu cầu đặc biệt | | Không có |
| Tiền điều kiện | | Người dùng đã đăng nhập thành công. |
| Hậu điều kiện | | Không có |
| Điểm mở rộng | | Không có |

***A screenshot of a computer screen

AI-generated content may be incorrect.***

#### 3.4.2.6*. Đặc tả UseCase Import File*

|  |  |  |
| --- | --- | --- |
| **Tên UseCase** | | **Import File** |
| Mô tả vắn tắt | | Use case này cho phép người dùng cho file (csv, xlsx) chứa các thông tin về bình luận đánh giá trên các nền tảng về sản phẩm bao gồm: id, content, created\_at vào hệ thống phân tích và thống kê. |
| Luồng sự kiện | Luồng cơ bản | Bước 1: Người dùng nhấn vào biểu tượng mũi tên ở header của trang Web.  Bước 2: Hệ thống hiển thị thông báo để người dùng chọn loại file.  Bước 3: Người dùng chọn file ở máy local lên hệ thống và nhấn vào nút “Nhập.”  Bước 4. Hệ thống phân tích file và lưu dữ liệu vào hệ thống.  Bước 5: Thông báo thành công cho người dùng. |
| Các luồng rẽ nhánh | Tại bước 4, nếu file gửi không đúng định dạng hỗ trợ hoặc không chứa các cột cần thiết. Hệ thống sẽ hiển thị thông báo lỗi.  Trong quá trình chương trình chạy nếu có lỗi database thì hệ thống sẽ gửi thông báo lỗi cho người dùng. |
| Các yêu cầu đặc biệt | | Không có |
| Tiền điều kiện | | Người dùng đã đăng nhập thành công vào hệ thống. |
| Hậu điều kiện | | Hệ thống sẽ cập nhập dữ liệu mới vào Database từ file dữ liệu của người dùng. |
| Điểm mở rộng | | Không có |

***A screenshot of a computer

AI-generated content may be incorrect.***

#### 3.4.2.7*. Đặc tả UseCase Crawl Data*

|  |  |  |
| --- | --- | --- |
| **Tên UseCase** | | **Quên mật khẩu** |
| Mô tả vắn tắt | | Use case này cho phép người dùng đặt lại mật khẩu khi quên. Sau khi điền chính xác thông tin tài khoản, hệ thống xác minh và gửi email để người dùng có thể đặt lại mật khẩu |
| Luồng sự kiện | Luồng cơ bản | Luồng cơ bản:  Bước 1: Use case này bắt đầu khi người dùng quên mật khẩu và chọn chức năng "Quên mật khẩu" trên giao diện đăng nhập.  Bước 2: Hệ thống hiển thị form nhập email đăng ký.  Bước 3: Người dùng nhập email và xác nhận.  Bước 4: Hệ thống kiểm tra email.  Bước 5: Nếu email hợp lệ, hệ thống gửi email chứa link đặt lại mật khẩu.  Bước 6: Người dùng truy cập link trong email, nhập mật khẩu mới và xác nhận.  Bước 7: Hệ thống cập nhật mật khẩu mới, thông báo thành công và chuyển về trang đăng nhập |
| Các luồng rẽ nhánh | - Nếu email không tồn tại hoặc không hợp lệ, hệ thống thông báo lỗi và yêu cầu nhập lại.  - Nếu người dùng không hoàn thành các bước, hệ thống hủy quá trình đặt lại mật khẩu. |
| Các yêu cầu đặc biệt | | Không có |
| Tiền điều kiện | | Người dùng đã có tài khoản và truy cập trang đăng nhập |
| Hậu điều kiện | | Người dùng có thể đăng nhập bằng mật khẩu mới hoặc quá trình đặt lại mật khẩu bị hủy nếu không hoàn thành. |
| Điểm mở rộng | | Không có |

#### 3.4.2.8*. Đặc tả UseCase tạo báo cáo*

|  |  |  |
| --- | --- | --- |
| **Tên UseCase** | | **Tạo báo cáo** |
| Mô tả vắn tắt | | Use case này cho phép người dùng tạo báo cáo chi tiết cho sản phẩm và tải xuống dạng pdf. |
| Luồng sự kiện | Luồng cơ bản | Bước 1: Người dùng nhấn vào trang báo cáo.  Bước 2: Hệ thống sẽ tạo 1 bản xem trước báo cáo.  Bước 3: Người dùng chọn các loại biểu đồ muốn có trong báo cáo.  Bước 4: Người nhấn vào nút “Xuất PDF”.  Bước 5: Hệ thống sẽ xuất file báo cáo dạng pdf và tự động tải về cho người dùng. |
| Các luồng rẽ nhánh | Trong quá trình tạo File PDF bị lỗi hệ thống sẽ thông báo cho người dùng. |
| Các yêu cầu đặc biệt | | Không có |
| Tiền điều kiện | | Người dùng đã có tài khoản và truy cập trang đăng nhập |
| Hậu điều kiện | | Người dùng có thể đăng nhập bằng mật khẩu mới hoặc quá trình đặt lại mật khẩu bị hủy nếu không hoàn thành. |
| Điểm mở rộng | | Không có |

*A diagram of a export report

AI-generated content may be incorrect.*

## 3.5. Kết quả đạt được

### 3.5.1. Đánh giá mô hình PhoBert trên tập dữ liệu

A screenshot of a computer code

AI-generated content may be incorrect.

Độ chính xác (accuracy) của mô hình đối với nhiệm vụ phân loại cảm xúc đạt 90.4%, với F1-score là 0.898. Đối với nhiệm vụ phân loại chủ đề (task), mô hình đạt accuracy 82.7% và F1-score 0.826. Các chỉ số này cho thấy mô hình đạt hiệu suất cao và ổn định ở cả hai nhiệm vụ, đặc biệt là ở nhiệm vụ phân loại cảm xúc, đồng thời khẳng định tính hiệu quả của kiến trúc đa nhiệm.

A screenshot of a computer screen

AI-generated content may be incorrect.

* Lớp “Tiêu cực” và “Tích cực” đều có F1-score rất cao (0.93 và 0.94), recall lần lượt là 0.96 và 0.95, cho thấy mô hình nhận diện tốt các cảm xúc rõ ràng.
* Lớp “Trung tính” có F1-score thấp hơn (0.57), recall chỉ đạt 0.49, phản ánh việc mô hình gặp khó khăn khi phân biệt các trường hợp trung tính, dễ bị nhầm lẫn sang các lớp cảm xúc mạnh hơn.

Độ chính xác tổng thể đạt 90%, macro F1-score là 0.82, xác nhận mô hình hoạt động tốt nhưng vẫn còn dư địa cải thiện ở lớp trung tính.

A screenshot of a computer

AI-generated content may be incorrect.

* Các lớp phổ biến như “Chất lượng”, “Giao hàng”, “Đóng gói” đạt F1-score từ 0.82 đến 0.89, recall cao, cho thấy mô hình nhận diện tốt các chủ đề có nhiều dữ liệu.
* Các lớp ít xuất hiện như “Đặt hàng”, “Trả hàng”, “Khác” có F1-score dao động từ 0.62 đến 0.70, recall thấp hơn, thể hiện sự khó khăn trong nhận diện các chủ đề hiếm hoặc có nội dung đa dạng.

Độ chính xác tổng thể đạt 83%, macro F1-score là 0.77, phản ánh hiệu suất tốt nhưng vẫn cần cải thiện ở các lớp nhỏ và phức tạp.

A screenshot of a graph

AI-generated content may be incorrect.

* Mô hình dự đoán chính xác phần lớn các mẫu “Tiêu cực” (2,495/2,612) và “Tích cực” (2,688/2,823).
* Lớp “Trung tính” bị nhầm lẫn khá nhiều: 187 mẫu bị gán nhầm thành “Tiêu cực”, 148 mẫu thành “Tích cực”, chỉ có 323/658 mẫu được nhận diện đúng. Điều này lý giải vì sao F1-score của lớp này thấp hơn hẳn hai lớp còn lại.
* Kết quả này cho thấy mô hình có xu hướng dự đoán về hai cảm xúc mạnh, trong khi các trường hợp trung tính dễ bị bỏ sót hoặc nhầm lẫn.

A screenshot of a computer

AI-generated content may be incorrect.

* Các lớp “Chất lượng”, “Giao hàng” và “Đóng gói” có số lượng dự đoán đúng cao, thể hiện qua các giá trị lớn trên đường chéo chính.
* Một số lớp nhỏ như “Đặt hàng”, “Trả hàng”, “Khác” có nhiều mẫu bị nhầm lẫn sang các lớp phổ biến hơn (ví dụ: 23 mẫu “Đặt hàng” bị nhầm thành “Giao hàng”, 235 mẫu “Khác” bị nhầm thành “Chất lượng”).
* Việc nhầm lẫn giữa các lớp có nội dung gần nhau (ví dụ “Đặt hàng” và “Giao hàng”, “Hỗ trợ khách hàng” và “Khác”) cho thấy mô hình gặp khó khăn khi phân biệt các chủ đề có ranh giới không rõ ràng hoặc ít dữ liệu huấn luyện.

### 3.5.2. Tích hợp Website

Trang đăng nhập – đăng ký:

A screenshot of a computer

AI-generated content may be incorrect.

Trang thống kê:

A screenshot of a computer

AI-generated content may be incorrect.

Trang tạo báo cáo:

A screenshot of a computer

AI-generated content may be incorrect.

Trang thu thập dữ liệu:

A screenshot of a computer

AI-generated content may be incorrect.

Trang giới thiệu:

A screenshot of a computer

AI-generated content may be incorrect.

# Chương 4: Kết Luận

## Nội dung đã làm

## Bài học kinh nghiệm

**Tóm lại**

# Tài Liệu Tham Khảo