BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC KINH TẾ TP. HỒ CHÍ MINH

ĐỒ ÁN MÔN HỌC

XỬ LÝ NGÔN NGỮ TỰ NHIÊN

PHÂN TÍCH CẢM XÚC NGƯỜI DÙNG DỰA VÀO ĐÁNH GIÁ ỨNG DỤNG SHOPEE TRÊN GOOGLE PLAY

Họ và tên	MSSV	Lớp
Đinh Trọng Hữu	31211027643	DS001
Nguyễn Thị Phương Thảo	31211027671	DS001
Nguyễn Quốc Việt	31211027687	DS001

GVHD: TS. Đặng Ngọc Hoàng Thành

TP. Hồ Chí Minh, Ngày 21 tháng 12 năm 2023

Mục lục

LỜI MỞ ĐẦU

Trong bối cảnh thương mại điện tử ngày càng phát triển mạnh mẽ, các ứng dụng mua sắm trực tuyến trở nên phổ biến và thu hút hàng triệu người dùng tại Việt Nam cũng như trên thế giới. Shopee hiện đang là một trong những ứng dụng thương mại điện tử hàng đầu Việt Nam với hơn 50 triệu lượt tải về và hàng triệu lượt đánh giá của khách hàng. Điều này cho thấy Shopee thực sự có ảnh hưởng lớn đến thị trường mua sắm online cũng như trải nghiệm của người tiêu dùng. Ngoài ra, với sự thuận tiện và linh hoạt mà Shopee mang lại, nó giúp người dùng dễ dàng lựa chọn và mua sắm với hàng nghìn sản phẩm đa dạng từ các nhà cung cấp khác nhau.

Tuy nhiên, để duy trì sự tăng trưởng và nâng cao vị thế của mình, Shopee cần không ngừng cải thiện trải nghiệm khách hàng thông qua việc lắng nghe và tiếp thu các phản hồi. Do ý kiến phản hồi của người dùng là nguồn thông tin quý báu giúp nhà phát triển nắm được thông tin về chất lượng và trải nghiệm người dùng trên ứng dụng của mình. Nhận thức được tầm quan trọng đó, nhóm thực hiện nghiên cứu đề tài **Phân tích cảm xúc người dùng dựa trên đánh giá ứng dụng Shopee trên nền tảng Google Play**. Thông qua việc phân tích một cách tự động các bình luận của người dùng để xác định cảm xúc là tiêu cực hay tích cực, nghiên cứu này đề xuất mô hình phân loại cảm xúc người dùng giúp hiểu rõ hơn về mong muốn, nhu cầu, và cảm xúc của người dùng đối với ứng dụng Shopee. Qua đó giúp nhà phát triển nhanh chóng cập nhật ứng dụng kịp thời để đáp ứng mong đợi ngày càng cao từ phía người dùng.

Nhóm cũng xin gửi lời cảm ơn đến Thầy Đặng Ngọc Hoàng Thành đã truyền đạt những kiến thức và kinh nghiệm quý báu của mình góp phần làm cho bài đồ án kết thúc môn học của nhóm trở nên hoàn chỉnh. Trong quá trình làm đồ án vẫn còn các hạn chế, sai sót, chưa tối ưu về mặt kiến thức và kỹ năng. Nhóm chúng em mong nhận được sự phản hồi, nhận xét của Thầy để nhóm có thể cải thiện được báo cáo tốt hơn nữa.

1 CHƯƠNG 1: TỔNG QUAN

1.1 Vai trò và ý nghĩa của Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) là một nhánh cực kỳ quan trọng của Trí tuệ nhân tạo (AI), là giao điểm của Ngôn ngữ học, Khoa học Máy tính và AI được hình thành lần đầu tiên vào thập niên 1950s [31]. Ứng dụng của NLP đóng vai trò then chốt trong việc cải thiện khả năng giao tiếp giữa người và máy, sự phát triển nhanh chóng của các thiết bị thông minh, trợ lý ảo, các giải pháp đám mây, cũng như nhu cầu tương tác với máy tính bằng ngôn ngữ tự nhiên ngày càng tăng. NLP làm nhiệm vụ xử lý và phân tích một lượng lớn dữ liệu ngôn ngữ tự nhiên để bắt chước các tương tác giữa con người theo cách giống con người một cách nhanh chóng và chính xác. Một hệ thống NLP tốt có thể hiểu được nội dung của văn bản, bao gồm cả ngữ cảnh và cảm xúc trong văn bản. Nhờ khả năng hiểu được ngữ cảnh và cảm xúc trong văn bản, NLP mở ra nhiều cơ hội phát triển và có ý nghĩa quan trọng trong việc đưa ra các quyết định dựa trên dữ liệu văn bản cho nhiều lĩnh vực như y tế, tài chính, giáo dục.

1.2 Giới thiệu đề tài

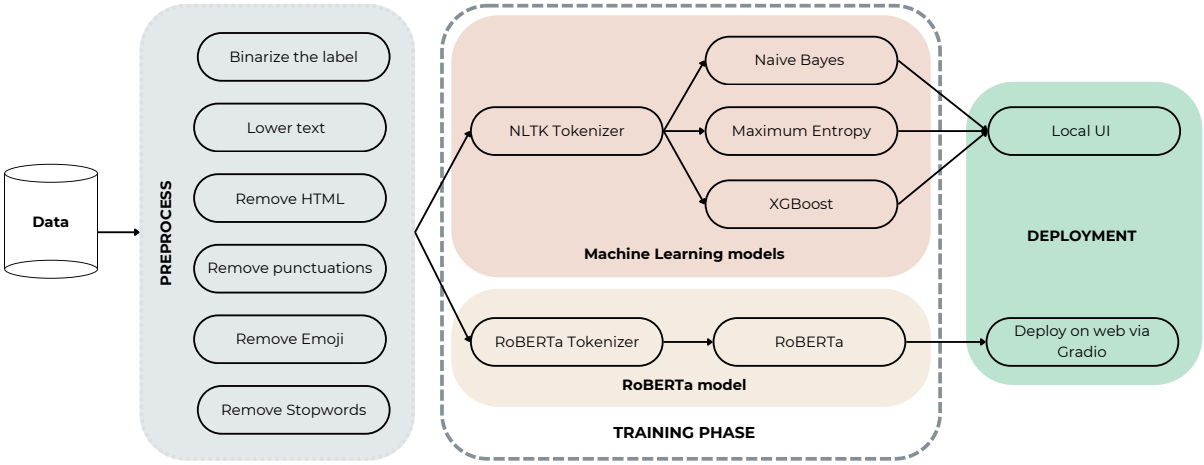
Sự xuất hiện của các trang thương mại điện tử đã đánh dấu một bước tiến mới, mang theo những thay đổi đáng kể trong quá trình giao dịch hàng hóa. Công nghệ đã đóng vai trò quan trọng trong việc làm cho mọi quy trình trở nên thuận tiện và nhanh chóng, chỉ cần vài cú nhấp chuột là giao dịch đã được hoàn tất. Những nền tảng thương mại điện tử nổi bật như Shopee, Lazada, Tiki không chỉ đơn thuần là nơi mua bán, mà còn mở rộng chức năng cho phép người mua đánh giá sản phẩm ngay sau khi nhận hàng.

Điều đặc biệt ở các sàn thương mại này là khả năng thu thập và phản hồi ý kiến từ cộng đồng trực tuyến, giúp doanh nghiệp theo dõi hành vi mua sắm, nhận biết sở thích, và đánh giá sự hài lòng của người dùng về chất lượng sản phẩm và dịch vụ. Và cũng chính các ứng dụng là này là sản phẩm ở trên các nền tảng khác ví dụ như Google Play.

Trong thời đại công nghệ được phát triển hàng đầu, nhu cầu tự động hóa việc thu thập và khai thác ý kiến bình luận trở nên ngày càng quan trọng, đặc biệt đối với những nhà đầu tư và doanh nghiệp quan tâm đến Sentiment Analysis and Classification. Điều này là do khả năng thu thập thông tin đa dạng từ hàng triệu ý kiến trên mạng, giúp hiểu rõ nhu cầu và quan điểm của người tiêu dùng đối với sản phẩm và dịch vụ. Với những thông tin thu thập được, doanh nghiệp có thể hiểu rõ hơn về yêu cầu và mong muốn của thị trường. Thông qua việc nắm bắt ý kiến của người dùng, doanh nghiệp có thể tối ưu hóa sản phẩm và dịch vụ của mình, từ đó đáp ứng tốt hơn nhu cầu của người dùng cũng như cải thiện và khắc phục các nhược điểm còn hiện hữu.

1.3 Mục đích đề tài

Bài báo cáo tập trung vào nghiên cứu mô hình phân tích cảm xúc và xử lý ngôn ngữ tự nhiên, thực nghiệm đề xuất kết quả dự đoán trên tập dữ liệu tiếng Anh là bình luận của người dùng về trên nền tảng Google Play về ứng dụng thương mại điện tử Shopee. Bộ dữ liệu Shopee App Review đầu tiên sẽ được tiền xử lý và trực quan hoá ở Chương 2. Tiếp theo, cơ sở lý thuyết của các mô hình máy học cho bài toán phân loại cảm xúc và cách huấn luyện sử dụng thư viện NLTK [1] sẽ được trình bày tại Chương 3.



Hình 1: Tổng quan quy trình thực hiện

Chương 4 trình bày lý thuyết mô hình RoBERTa và cách huấn luyện sử dụng thư viện PyTorch [2]. Chi tiết về kết quả phân lớp của các mô hình sẽ được trình bày tại Chương 5, các mô hình đã được sẽ được triển khai với giao diện trực quan trên ứng dụng và web tại Chương 6. Cuối cùng, nhóm tiến hành tổng kết các kết quả đạt được cũng như hướng phát triển cho đồ án tại Chương 7. Hình 1 thể hiện tổng quan các bước nhóm đã thực hiện trong phạm vi đồ án.

[1] https://www.nltk.org/

[2] https://pytorch.org/

2 CHƯƠNG 2: TỔNG QUAN VÀ TIỀN XỬ LÝ DỮ LIỆU

2.1 Tổng quan và trực quan hoá dữ liệu Shopee App Review

2.2 Tổng quan bộ dữ liệu

Bộ dữ liệu Shopee App Review thu thập các thông tin đánh giá tích cực và tiêu cực trên Google Play của Shopee Singapore. Bộ dữ liệu bao gồm 12 thuộc tính và 7404 dòng. Các thuộc tính có thể được mô tả như sau:

STT	Tên thuộc tính	Ý nghĩa
1	reviewId	Mã nhận dạng duy nhất của bài đánh giá
2	userName	Tên tài khoản
3	userImage	Liên kết đến ảnh đại diện của người dùng
4	content	Nội dung của bài đánh giá
5	score	Số sao mà người dùng đánh giá (1-5)
6	thumbsUpCount	Số lượng người thích bài đánh giá
7	reviewCreatedVersion	Phiên bản ứng dụng
8	at	Ngày và giờ viết bài đánh giá
9	replyContent	Phản hồi của Shopee cho bài đánh giá
10	repliedAt	Ngày và giờ Shopee trả lời
11	sort_order	Cho biết dữ liệu lấy từ phần “Liên quan nhất” hay “Mới nhất” trên Google Play
12	app_id	URL nơi đánh giá được thu thập

Bảng 1: Mô tả thuộc tính của dữ liệu đánh giá

2.3 Tiền xử lý dữ liệu

2.3.1 Nhị phân hóa điểm số

Đầu tiên nhóm xác định ngưỡng cho 2 mức đánh giá là tiêu cực và tích cực, với các đánh giá có sao thấp hơn 3 sao được xem tiêu cực và gán nhãn là 0, các đánh giá từ 3 sao đến 5 sao được xem là tích cực và gán nhãn là 1.

```
threshold = 3
df['target'] = (df['score'] > threshold).astype(int)
df = df[['content','target']]
df.head()
```

2.3.2 Tiền xử lý văn bản

Nhóm tiến hành tiền xử lý theo các bước như sau:

Đầu tiên sẽ chuyển đổi tất cả các chữ cái trong văn bản thành chữ thường là một phương pháp tiền xử lý dữ liệu phổ biến trong quá trình làm sạch dữ liệu. Bước này giúp đảm bảo sự nhất quán, không phụ thuộc vào việc chữ cái viết hoa hay viết thường, từ đó giảm độ phức tạp của dữ liệu và tối ưu hóa quá trình phân tách từ.

Tiếp theo nhóm tiến hành thay thế các đường link từ web bằng các khoảng trắng vì các đường link này không mang lại nhiều thông tin hữu ích trong việc phân loại, và chúng còn là nguồn gây nhiễu. Loại bỏ các đường link còn giúp giảm kích thước của dữ liệu và làm tăng hiệu suất xử lý.

Nếu văn bản có chứa mã HTML, nhóm sẽ tiến hành loại bỏ thẻ HTML với mong muốn sẽ lấy được nội dung văn bản thuần túy mà không bị ảnh hưởng bởi các thẻ định dạng hay mã nguồn HTML.

Các ký tự đặc biệt và số không mang nhiều thông tin ý nghĩa trong ngữ cảnh dữ liệu này. Nhóm quyết định thay thế chúng bằng các khoảng trắng giúp loại bỏ nhiễu và tăng tính tập trung vào các từ ngữ quan trọng

Nhóm còn tiến hành loại bỏ các khoảng trắng dư thừa giúp đồng nhất hóa không gian giữa các từ. Đồng thời, loại bỏ khoảng trắng dư thừa cũng giúp tăng hiệu suất trong quá trình xử lý văn bản, vì các thao tác tiếp theo như tách từ, loại bỏ stop words, hay đào tạo mô hình máy học sẽ được thực hiện trên dữ liệu đã được chuẩn hóa một cách đồng nhất. Điều này có thể giúp giảm thời gian và tài nguyên tính toán cần thiết cho các công đoạn xử lý và đào tạo mô hình.

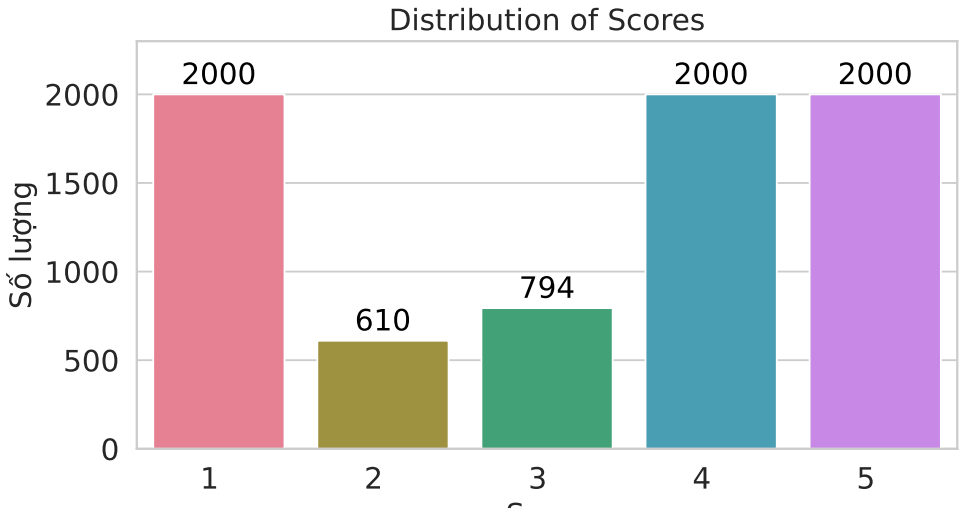
Một bước quan trọng nữa trong việc làm sạch dữ liệu văn bản là loại bỏ emoji. Emoji thường chiếm nhiều byte trong chuỗi ký tự so với các ký tự thông thường. Loại bỏ chúng giúp giảm kích thước của dữ liệu. Ngoài ra nó còn mang ý nghĩa tương đối phức tạo và gây nhiễu cho các mô hình máy học. Bằng cách loại bỏ emoji, dữ liệu trở nên tập trung vào các yếu tố ngôn ngữ chính, giúp mô hình tập trung vào thông điệp ngôn ngữ chính xác hơn.

Tiếp theo, bước *Tách từ* (Tokenize) có vai trò quan trọng trong việc chuẩn hóa đầu vào của văn bản. Điều này được thực hiện thông qua quá trình chia văn bản thành các đơn vị ý nghĩa nhỏ nhất, còn được gọi là *tokens*. Tokenize là bước tiền xử lý quan trọng để chuẩn bị dữ liệu cho các bước tiếp theo như loại bỏ stop words, vector hóa từ, hay huấn luyện các mô hình.

Cuối cùng nhóm thực hiện loại bỏ các stop words. Stop words là nhóm các từ phổ biến như "and", "the", "is",... có xu hướng xuất hiện nhiều trong văn bản mà thường không mang lại nhiều ý nghĩa quan trọng. Trong nhiều trường hợp, chúng là các từ liên quan đến ngữ pháp và không phản ánh thông tin chủ đề chính. Việc làm này được xem là một phần quan trọng trong tiền xử lý văn bản, nhằm giảm nhiễu và tập trung vào các từ quan trọng trong quá trình xử lý và phân loại văn bản.

2.4 Trực quan hoá dữ liệu

Nhóm sẽ trực quan rõ hơn về một thuộc tính quan trọng được nhắc đến ở phần trên chính là các mức điểm đánh giá:



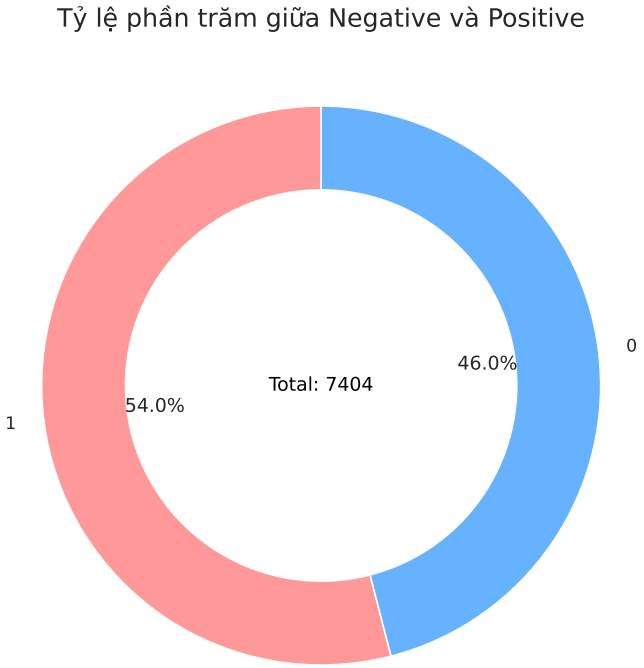
Hình 2: Thống kê số lượng đánh giá theo sao

Thông qua biểu đồ, có thể thấy có 2000 lượt đánh giá ở các mức sao là 1, 4 và 5. Có thể thấy một số lượng lớn người khá hài lòng và có những trải nghiệm tích cực ở ứng dụng. Tuy nhiên, không tránh khỏi việc có một số người dùng đã gặp những trải nghiệm không tốt với ứng dụng ở một số khía cạnh.

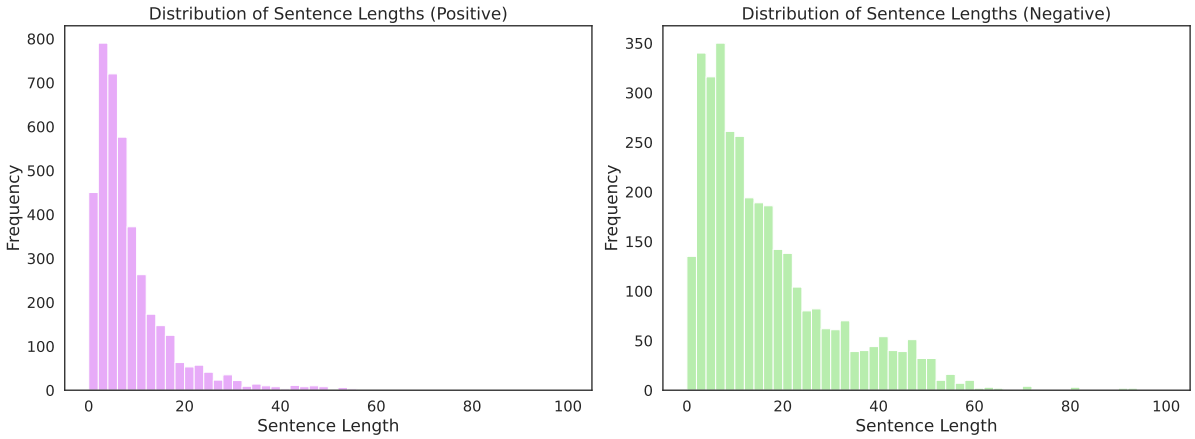
Ngoài ra, với các mức sao là 2 và 3 thì có số lượng người dùng đánh giá thấp với lần lượt là 610 và 794. Điều này có thể cho thấy có một phần đáng kể của người dùng không hoàn toàn hài lòng và có một vài điểm cần ứng dụng phải cải thiện hơn. Những nhận định trên sẽ giúp nhóm có cái nhìn toàn diện hơn về sự đa dạng trong trải nghiệm của người dùng trên ứng dụng.

Tỷ lệ phần trăm giữa Negative và Positive

Thông qua biểu đồ có thể thấy sau khi thực hiện các bước tiền xử lý, nhóm thu được dữ liệu đã được gán nhãn, với dữ liệu là Positive chiếm 54% và dữ liệu Negative chiếm 46%. Sự chênh lệch nhỏ này có thể chấp nhận được trong quá trình xây dựng mô hình.



Tiếp theo, nhóm tiến hành xem xét chi tiết hơn về độ dài của các câu ở cả 2 khía cạnh positive và negative.



Hầu hết các câu trong đánh giá Positive có độ dài tập trung chủ yếu dưới 20 từ. Điều này có thể chỉ ra rằng người dùng thường chia sẻ nhận xét ngắn gọn và tập trung vào các ưu điểm tích cực của ứng dụng. Đối lập với đánh giá Positive, đánh giá Negative có xu hướng có độ dài trải rộng hơn, kéo dài đến khoảng 60 từ. Sự phân phối rộng có thể chỉ ra rằng người dùng khi phê phán hoặc góp ý tiêu cực có thể trình bày chi tiết hơn về vấn đề hoặc trải nghiệm không hài lòng của họ.

WordCloud

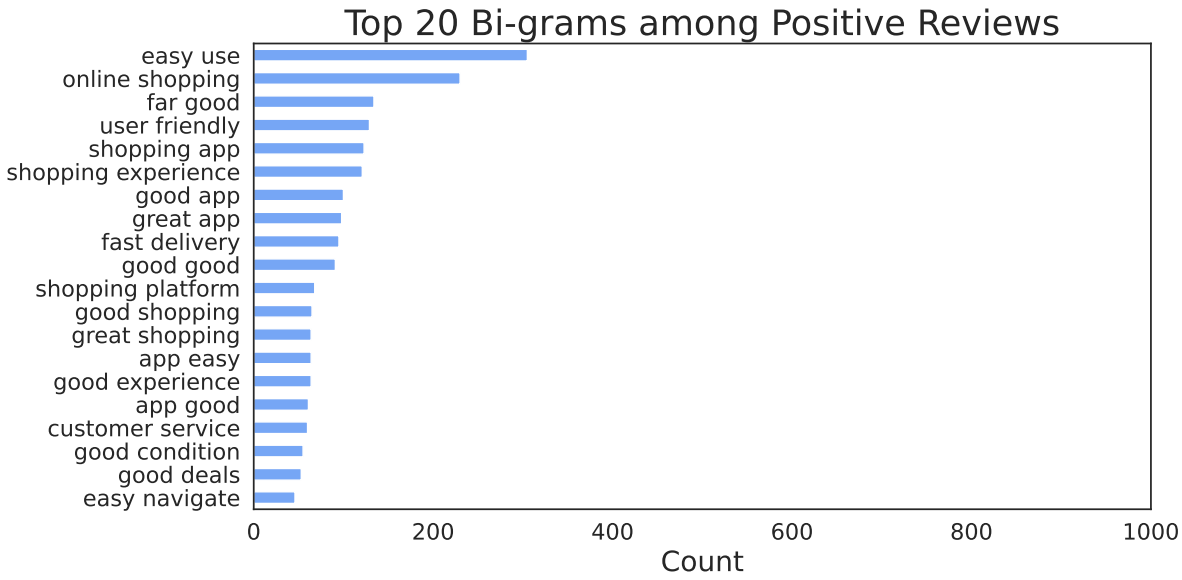
Nhóm tiến hành trực quan hóa bằng WordCloud để xem tần số xuất hiện của các từ trong các đánh giá mang nhãn Positive và Negative. Đầu tiên sẽ là WordCloud với các đánh giá mang nhãn Positive:

Có thể thấy các từ có tần số suất hiện nhiều nhất là good, item, app, easy, great… có thể thấy người tiêu dùng miêu tả và đánh giá tích cực về ứng dụng. Từ “good” xuất hiện nhiều nhất, có thể là dấu hiệu cho sự hài lòng tổng thể. Từ "easy"cũng đáng chú ý, là một từ khóa thường xuyên xuất hiện, thể hiện rằng người dùng đánh giá cao tính đơn giản và dễ sử dụng của sản phẩm. Điều này có thể được coi là một yếu tố quan trọng khi người dùng đánh giá tích cực về trải nghiệm của họ với ứng dụng. Tổng thể, WordCloud của đánh giá Positive cho thấy một bức tranh lạc quan và tích cực, đồng thời mang lại thông tin quan trọng về những đặc điểm mà người dùng đánh giá và miêu tả tích cực trong ứng dụng.

Các từ có tần số xuất hiện nhiều nhất có thể kể đến như là “refund, app, shopee, item, even, problem…”. Các từ khóa như "refund", "app", và "problem"thường liên quan đến chất lượng và hiệu suất của ứng dụng và dịch vụ, cho thấy rằng người dùng có thể gặp phải các vấn đề liên quan đến sự hoàn trả, trải nghiệm ứng dụng, và các vấn đề khác. Sự đa dạng của các từ khóa như "refund", "app", "item", "even", và "problem"chỉ ra rằng có nhiều khía cạnh khác nhau của trải nghiệm người dùng được gán nhãn negative, bao gồm cả các vấn đề về dịch vụ người dùng, sản phẩm, và tính ổn định của ứng dụng.

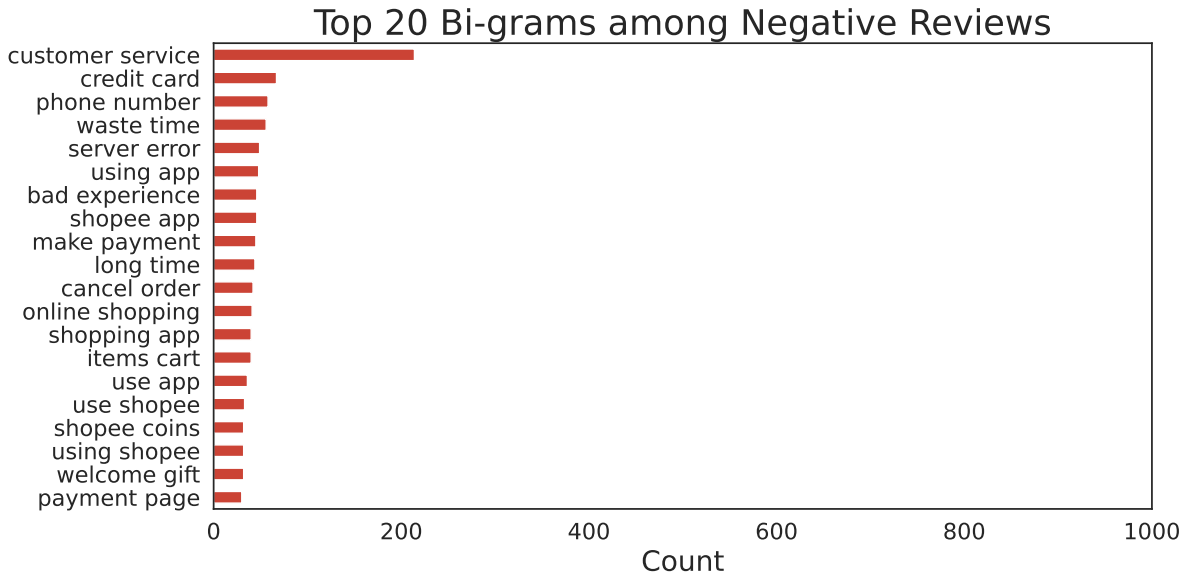
Bigram

Sau khi thực hiện đánh giá sơ bộ về tần suất xuất hiện của từ đơn trong 2 nhãn, nhóm tiếp tục nghiên cứu chi tiết về các cụm từ (cụm 2 từ) phổ biến nhất trong tập dữ liệu. Việc này giúp nhóm hiểu rõ hơn về ngữ cảnh và mối quan hệ giữa các từ, từ đó cung cấp thông tin quan trọng về cách người dùng miêu tả và đánh giá ứng dụng:



Khi chi tiết hóa việc phân tích cụm từ (bi-grams) trong các nhận xét được gán nhãn Positive, nhóm nhận thấy một số cụm từ phổ biến thường xuất hiện, ví dụ như "easy use", "app easy", "easy navigate", cũng như "good app", "good shopping", "good experience".

Từ những cụm từ này, có thể suy đoán rằng người dùng thường đánh giá cao khả năng sử dụng dễ dàng của ứng dụng và trải nghiệm tích cực khi mua sắm qua nó.



Với các nhận xét mang nhãn Negative, việc phân tích cụm từ tiếp tục cho thấy cụm từ "customer service"xuất hiện nhiều nhất. Điều này chỉ ra rằng các vấn đề liên quan đến dịch vụ người dùng đang là một điểm yếu của ứng dụng và cần sự cải thiện. Ngoài ra, cụm từ như "credit card", "make payment", "payment page"chỉ ra rằng có những vấn đề liên quan đến thanh toán, trong khi "server error""using app", "Shopee app"chỉ ra rằng có các vấn đề hệ thống cần được khắc phục.

3 CHƯƠNG 3: PHÂN TÍCH CẢM XÚC VÀ DỰ ĐOÁN BẰNG CÁC PHƯƠNG PHÁP MÁY HỌC

3.1 Cơ sở lý thuyết

3.1.1 Naive Bayes Classifier

Naive Bayes là một thuật toán xác suất cơ bản được áp dụng rộng rãi trong lĩnh vực phân loại văn bản, trong đó bao gồm cả bài toán phân tích cảm xúc. Naive Bayes hoạt động theo nguyên tắc định lý Bayes [2]. Trong đó, định lý Bayes được định nghĩa như sau:

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} \tag{1}$$

Giả sử ta có tập văn bản d và các lớp c (trong bài toán phân tích cảm xúc, $c \in \{0,1\}$ tương ứng với việc văn bản là tích cực hay tiêu cực), mục tiêu của Naive Bayes là tìm lớp c sao cho xác suất $P(c|d)$ là lớn nhất, hay:

$$\hat{c} = \underset{c \in \mathscr{C}}{\text{argmax}} P(c|d) \tag{2}$$

Khi này, bằng định lý Bayes, ta có thể viết lại hàm mục tiêu được biểu diễn ở Phương trình 2 của thuật toán Naive Bayes như sau:

$$\hat{c} = \underset{c \in \mathscr{C}}{\text{argmax}} P(c|d) = \underset{c \in \mathscr{C}}{\text{argmax}} \frac{P(d|c)P(c)}{P(d)} = \underset{c \in \mathscr{C}}{\text{argmax}} P(d|c)P(c) \tag{3}$$

Trong đó, dấu bằng thứ ba xảy ra vì ta có thể xem $P(d)$ như một hằng số và không ảnh hưởng đến việc tìm nghiệm tối ưu.

Đối với thuật toán Naive Bayes, thông tin văn bản được chuyển thành các feature dạng số bằng sử dụng các mô hình như Bag of Words hoặc TF-IDF. Theo đó, từ tập văn bản d ban đầu, thông qua các kĩ thuật như Bag of Words hay TF-IDF, ta sẽ được các vector biểu diễn từng từ $w_1, w_2, \cdots, w_n$ trong tập văn bản d. Khi này, phương trình 3 có thể được viết lại dưới dạng:

$$\hat{c} = \underset{c \in \mathscr{C}}{\text{argmax}} P(d|c)P(c) = \underset{c \in \mathscr{C}}{\text{argmax}} P(w_1, w_2, \cdots, w_n|c)P(c)$$

Một đặc trưng của thuật toán Naive Bayes là giả định rằng một thuộc tính cụ thể trong một lớp là độc lập với các thuộc tính khác. Trong bài toán phân tích cảm xúc, điều này

có nghĩa là đóng góp của từng từ cho lớp của văn bản là độc lập với các từ khác. Nói cách khác, với $w_1, w_2, \cdots w_n$ lần lượt là các vector biểu diễn từ có trong tập văn bản và c là lớp của văn bản đó, với giả định trên, ta có thể viết lại xác suất xuất hiện của các từ $w_1, w_2 \cdots w_n$ biết văn bản thuộc lớp c $P(w_1, w_2 \cdots w_n|c)$ như sau:

$$P(w_1, w_2 \cdots w_n|c) = P(w_1|c)P(w_2|c) \cdots P(w_n|c) = \prod_{i=1}^{n} P(w_i|c)$$

Mặc dù giả định là ngây thơ *(Naive)*, mô hình Naive Bayes Classifier tỏ ra hiệu quả đối với các bộ dữ liệu trên thực tế. Với giả định rằng các từ độc lập với nhau, ta có thể viết lại hàm mục tiêu của mô hình Naive Bayes như sau:

$$\begin{aligned}
\hat{c} &= \underset{c \in \mathrm{C}}{\operatorname{argmax}} P(d|c)P(c) \\
&= \underset{c \in \mathrm{C}}{\operatorname{argmax}} P(w_1, w_2, \cdots w_n|c)P(c) \\
&= \underset{c \in \mathrm{C}}{\operatorname{argmax}} P(w_1|c)P(w_2|c) \cdots P(w_n|c)P(c) \quad \text{(Naive Assumption)} \\
&= \underset{c \in \mathrm{C}}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|c)P(c)
\end{aligned}$$

Để tránh tình trạng *underflow* khi các giá trị xác suất có giá trị nhỏ nhân với nhau dẫn đến việc hàm mục tiêu cần tối ưu tiến về 0, ta có thể thực hiện việc lấy log cho hàm mục tiêu:

$$\begin{aligned}
\hat{c} &= \underset{c \in \mathrm{C}}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i|c)P(c) \\
&= \underset{c \in \mathrm{C}}{\operatorname{argmax}} \log \prod_{i=1}^{n} P(w_i|c)P(c) \\
&= \underset{c \in \mathrm{C}}{\operatorname{argmax}} \log \prod_{i=1}^{n} P(w_i|c) + \log P(c) \\
&= \underset{c \in \mathrm{C}}{\operatorname{argmax}} \sum_{i=1}^{n} \log P(w_i|c) + \log P(c)
\end{aligned}$$

Trong quá trình huấn luyện, mô hình Naive Bayes sẽ thực hiện tính toán xác suất của từng lớp cảm xúc (negative/positive) dựa trên tập dữ liệu huấn luyện bằng cách đếm số lượng từ. Đôi khi, Laplace Smoothing sẽ được áp dụng để tăng tính ổn định cho thuật toán.

Giả định độc lập thường không đúng trong sự phức tạp của ngôn ngữ con người, trong đó phụ thuộc ngữ cảnh và từ đóng một vai trò quan trọng trong việc truyền đạt cảm xúc. Hơn nữa, sự phụ thuộc của mô hình vào tần suất từ có thể dẫn đến việc giải thích sai cảm xúc trong các cụm từ trong đó phủ định hoặc tính từ nhất định là mấu chốt. Tuy nhiên, tốc độ và sự dễ dàng xử lý khối lượng dữ liệu lớn của Naive Bayes là điều làm mô hình này vẫn được sử dụng phổ biến trong các bài toán phân tích cảm xúc.

3.1.2 Maximum Entropy

Đại lượng Entropy [26] của một phân phối xác suất là một độ đo về sự không chắc chắn, hay sự không dự đoán được của một biến ngẫu nhiên liên tục tuân theo một phân phối xác suất cho trước.

Entropy cũng có thể được định nghĩa là khối lượng thông tin chứa trong một bộ dữ liệu cho trước. Ví dụ, giả sử ta có các quan sát $X_n \sim p$ và tập dữ liệu $\mathscr{D} = (X_1, \ldots, X_n)$ được tạo ra từ một phân phối xác suất p, phân phối p sẽ có giá trị Entropy cao nếu các quan sát X_n là khó để dự đoán. Khi này bộ dữ liệu $\mathscr{D}$ sẽ có khối lượng thông tin cao, ngược lại, nếu tất cả các quan sát trong tập dữ liệu $\mathscr{D}$ là giống nhau, khi này tập dữ liệu $\mathscr{D}$ không chứa quá nhiều thông tin.

Một cách toán học, với một biến ngẫu nhiên rời rạc với các giá trị đầu ra $X_1, X_2, \ldots, X_n$ và xác suất tương ứng là $P(X_1), P(X_2), \ldots, P(X_n)$, đại lượng Entropy H được định nghĩa là:

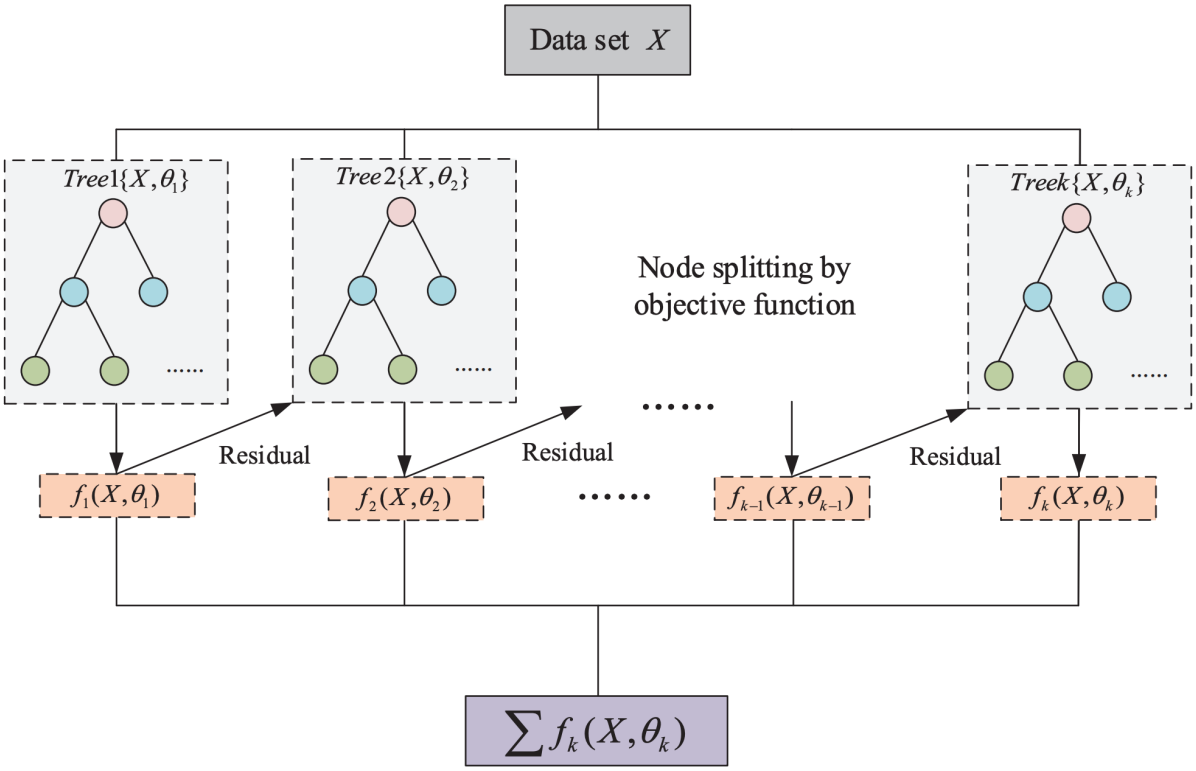
$$H(X) = -\sum_{i=1}^{n} P(X_i) \log P(X_i)$$

Một tính chất quan trọng của entropy là đại lượng này sẽ đạt cực đại khi tất cả các đầu ra có xác suất bằng nhau và ngược lại, cực tiểu hoá khi có một giá trị đầu ra chắc chắn xảy ra, hay xác suất để sự kiện đó xảy ra bằng 1. Theo đó, trong các hàm phân phối xác suất của biến ngẫu nhiên liên tục, Uniform là phân phối xác suất có giá trị Entropy cao nhất, phân phối chuẩn là phân phối của biến ngẫu nhiên liên tục có giá trị Entropy cao nhất với kì vọng μ và phương sai σ cho trước.

Ý tưởng chính của mô hình Maximum Entropy trong lĩnh vực Xử lý ngôn ngữ tự nhiên [4] là sử dụng một mô hình xác suất, trong đó mô hình sẽ chọn ra một phân phối xác suất có chỉ số entropy cao nhất với các ràng buộc cho trước. Các ràng buộc này thường được lấy từ tập dữ liệu huấn luyện. Đối với bài toán phân tích cảm xúc, mô hình MaxEnt sẽ sử dụng các đặc trưng từ các văn bản trong bộ dữ liệu để dự đoán cảm xúc.

3.1.3 Mô hình XGBoost Classifier

XGBoost (eXtreme Gradient Boosting) [6] là một mô hình với hiệu suất và độ linh hoạt cao. XGBoost xây dựng một cách tuần tự các cây quyết định, trong đó các cây sau sẽ sửa lỗi của các cây trước. Quá trình lặp lại này giúp cải thiện khả năng của mô hình, từ đó đưa ra các dự đoán chính xác hơn.



Hình 3: Tổng quan cách hoạt động mô hình XGBoost

Trong bài toán phân tích cảm xúc, mô hình XGBoost rất hiệu quả bằng cách xử lý các dữ liệu văn bản (thông thường được chuyển đổi qua định dạng số bằng các kĩ thuật như TF-IDF hoặc Word Embeddings) để dự đoán nhãn tương ứng.

3.2 Huấn luyện các thuật toán máy học

Sau khi đã thực hiện tiền xử lý và tokenize, ta có thể thực hiện việc huấn luyện mô hình Naive Bayes từ thư viện NLTK như sau:

```
featuresets = [(document_features(d), c) for (d, c) in documents]
train_set, test_set = train_test_split(featuresets, test_size=0.2,
 ↪  random_state=42)

# fit naive bayes classifier to the train set
nb_classifier = NaiveBayesClassifier.train(train_set)
```

Sau đó ta sẽ thực hiện dự đoán trên tập test:

```
true_labels = [label for (_, label) in test_set]
predicted_labels = [nb_classifier.classify(features) for (features, _) in
↪  test_set]
```

Tương tự đối với mô hình Maxent, ta có thể thực hiện huấn luyện và dự đoán dựa vào thư viện NLTK như sau:

```
maxent_classifier = MaxentClassifier.train(train_set, algorithm='gis', trace=0,
↪  max_iter=10)

true_labels = [label for (_, label) in test_set]
predicted_labels = [maxent_classifier.classify(features) for (features, _) in
↪  test_set]
```

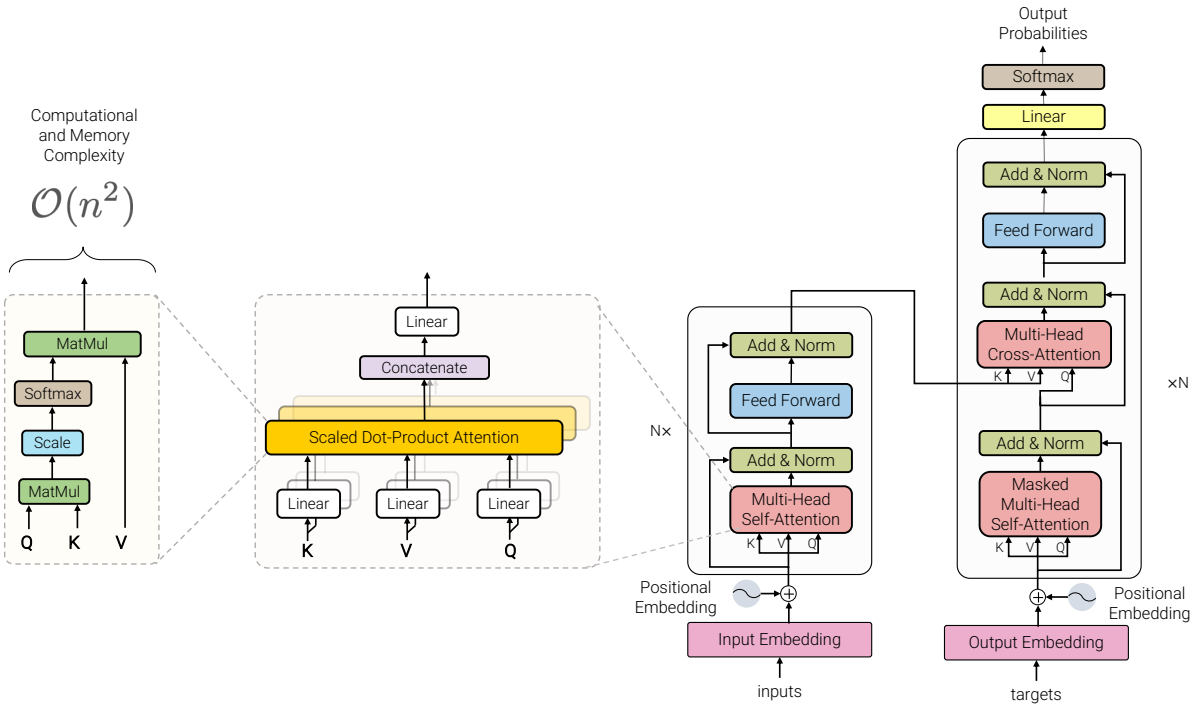
Do thư viện NLTK không hỗ trợ việc huấn luyện mô hình XGBoost, nên việc huấn luyện mô hình và dự đoán có thể được thực hiện như sau:

```
from xgboost import XGBClassifier

from sklearn.feature_extraction import DictVectorizer

X, y = zip(*featuresets)

# Transform the list of dictionaries into a 2D array
vectorizer = DictVectorizer(sparse=False)
X_transformed = vectorizer.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_transformed, y,
↪  test_size=0.2, random_state=42)

xgb_classifier = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss')
xgb_classifier.fit(X_train, y_train)

predicted_labels = xgb_classifier.predict(X_test)
```

4 CHƯƠNG 4: PHÂN TÍCH CẢM XÚC VÀ DỰ ĐOÁN BẰNG MÔ HÌNH ROBERTA

4.1 Các mô hình học sâu dựa trên Transformer

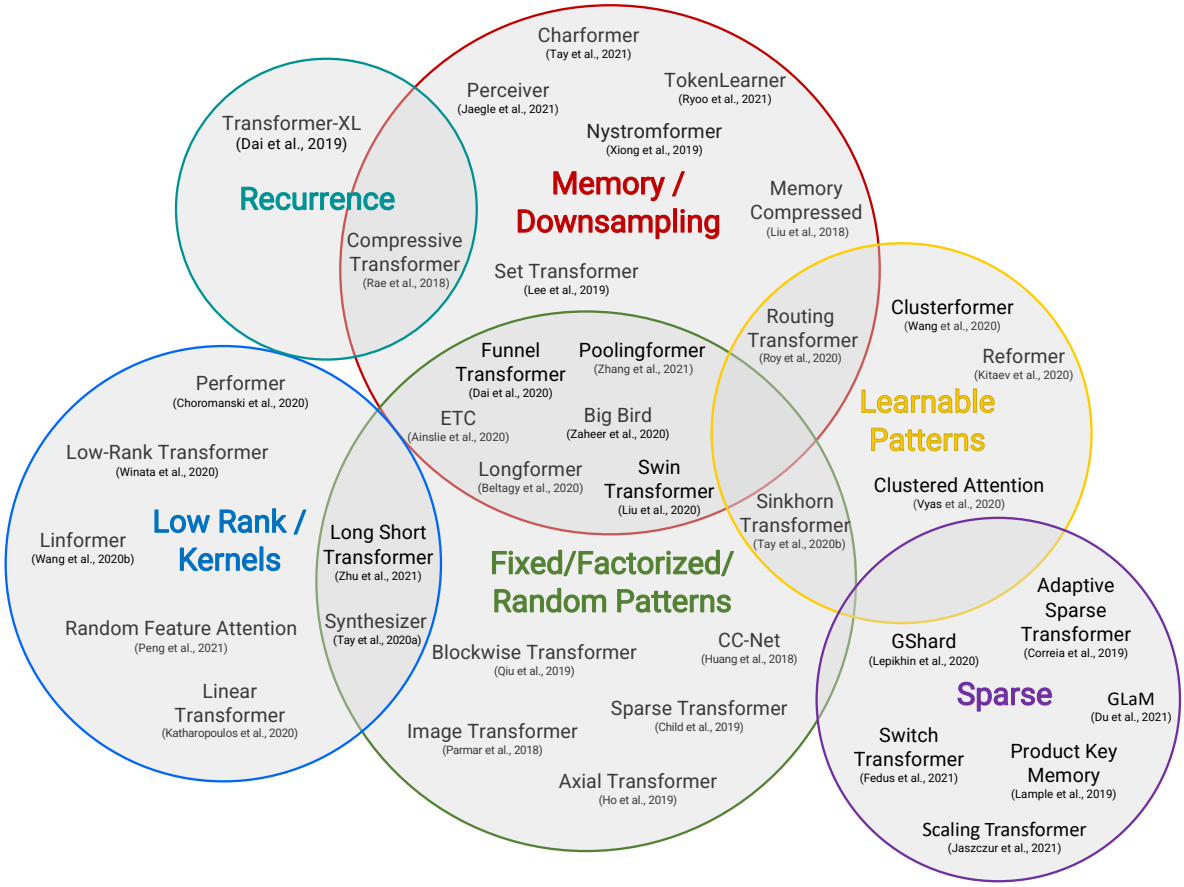
Được giới thiệu trong bài báo *Attention Is All You Need* bởi Vaswani et al. vào năm 2017 [29], Transformer là một loại mạng nơ-ron sử dụng để giải quyết bài toán dịch máy. Transformer đã đánh bại nhiều mô hình truyền thống trong nhiều nhiệm vụ NLP và được sử dụng rộng rãi trong các ứng dụng thực tế.



Hình 4: Tổng quan kiến trúc mô hình Transformer [27]

Phần quan trọng nhất của Transformer là *cơ chế tự chú ý* (self-attention). Tại mỗi bước thời gian, mô hình Transformer xem xét toàn bộ chuỗi đầu vào để tính toán trọng số cho mỗi từ. Các từ quan trọng trong ngữ cảnh được trọng số cao hơn. Cơ chế tự chú ý này cho phép mạng hiểu ngữ cảnh rộng hơn.

Từ khi ra đời, mô hình Transformer đã bùng nổ và được xem như một cuộc cách mạng trong lĩnh vực AI nhờ việc đem lại những cải tiến đáng kể trong hiệu suất xử lý ngôn ngữ tự nhiên và còn được sử dụng trong nhiều loại dữ liệu khác nhau như hình ảnh [11, 12, 14], dữ liệu chuỗi thời gian [1, 22, 32] và nhiều lĩnh vực khác. Từ sau năm 2017, đã có rất nhiều các mô hình khác được phát triển dựa trên mô hình gốc để cải thiện hiệu suất đồng thời giảm bộ nhớ và yêu cầu về khả năng tính toán, bao gồm Linformer [30], Performer [7], Longformer [3], ...



Hình 5: Tổng quan các hướng phát triển mô hình Transformers [27]

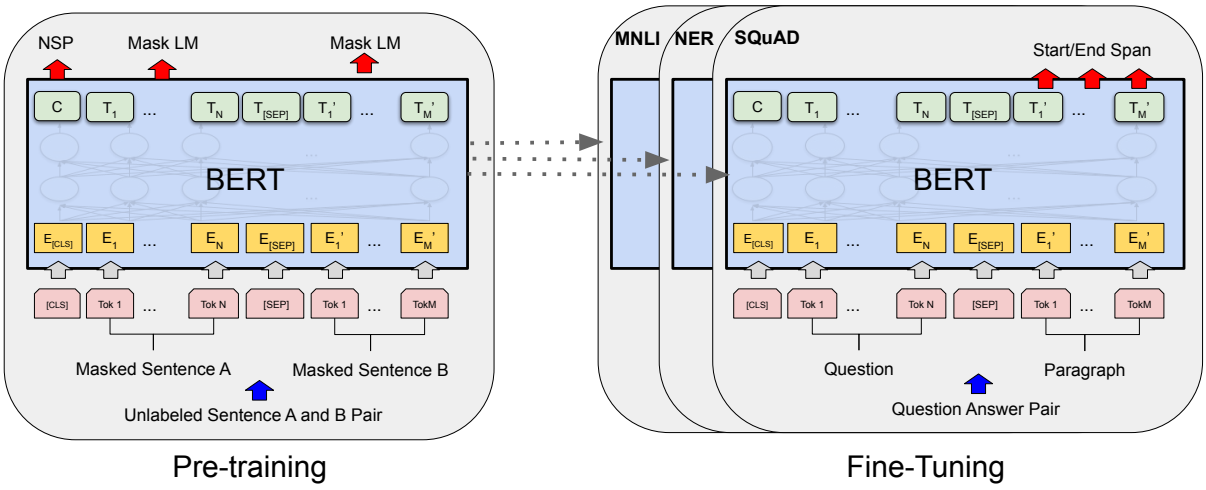
4.2 Tổng quan kiến trúc mô hình BERT

BERT (Bidirectional Encoder Representations from Transformers) [10] là mô hình được huấn luyện dựa trên kiến trúc mạng Transformers [29]. Mô hình BERT được thiết kế để pretrain trên các biểu diễn của các từ một cách đa chiều từ các văn bản không có nhãn bằng cách dựa vào cả ngữ cảnh theo chiều từ trái sang phải và ngược lại trong tất cả các layers. Mô hình pretrained BERT đã đạt được những kết quả ấn tượng khi được tinh chỉnh (fine-tuned) với các tác vụ khác nhau mà không cần phải điều chỉnh kiến trúc mô hình.

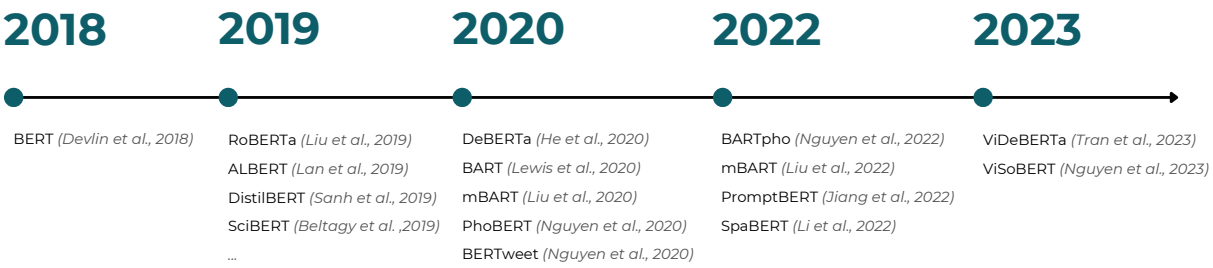
Kể từ sự ra đời của mô hình BERT, rất nhiều các mô hình khác dựa trên BERT đã được giới thiệu nhằm cải thiện hiệu năng của BERT và tinh chỉnh trên các bộ dữ liệu trong nhiều lĩnh vực khác nhau. Hình 7 liệt kê các mô hình nổi bật dựa trên BERT qua các năm.

4.3 Mô hình RoBERTa

RoBERTa (Robustly Optimized BERT Approach) [16] là một mô hình được xây dựng trên nền tảng của BERT với một vài tinh chỉnh giúp cải thiện kết quả và hiệu suất trên



Hình 6: Tổng quan kiến trúc mô hình BERT [10]



Hình 7: Các mô hình nổi bật dựa trên mô hình BERT

nhiều chỉ số và tác vụ khác nhau. Cụ thể, RoBERTa được huấn luyện trong thời gian dài hơn và với batch size lớn hơn, đồng thời các câu trong bộ dữ liệu huấn luyện cũng có độ dài dài hơn. Điều này cho phép mô hình học được các chi tiết phức tạp từ một khối lượng lớn dữ liệu. Thêm vào đó, mô hình RoBERTa không thực hiện huấn luyện dựa trên dự đoán câu tiếp theo (Next Sentence Prediction) như đối với BERT mà thay vào đó chỉ huấn luyện trên một tác vụ duy nhất là masked language model. Cuối cùng, mô hình RoBERTa thực hiện mask các dữ liệu huấn luyện một cách linh hoạt, cho phép mô hình học được nhiều hơn từ đa dạng các biểu diễn. Bằng các tinh chỉnh này, RoBERTa là mô hình đạt được các kết quả tối ưu và tốt nhất vào thời điểm nó được giới thiệu. Trong phạm vi đồ án, mô hình `roberta-base` [3] từ HuggingFace sẽ được sử dụng để huấn luyện cho bài toán phân tích cảm xúc.

4.4 Phân tích và dự đoán cảm xúc bằng mô hình RoBERTa

Đầu tiên ta sẽ tạo một lớp dataset tùy chỉnh trong PyTorch, được thiết kế cho việc huấn luyện mô hình RoBERTa. Lớp ShopeeDataset được tối ưu hóa để xử lý dữ liệu với cấu trúc đặc biệt, giúp quá trình huấn luyện trở nên hiệu quả và linh hoạt hơn.

[3] https://huggingface.co/roberta-base

Trong phương thức khởi tạo `__init__`, lớp này nhận vào encodings - là các vector đặc trưng đã được mã hóa từ văn bản và là kết quả của quá trình tokenization và labels (nhãn tương ứng cho mỗi mẫu dữ liệu).

Phương thức `__getitem__` cho phép truy cập và trả về một mẫu dữ liệu cụ thể từ tập dữ liệu, bao gồm cả mã hóa và nhãn, dựa trên chỉ số được cung cấp giúp đảm bảo rằng mỗi lần lấy mẫu để huấn luyện, mô hình sẽ nhận được đúng dữ liệu và nhãn tương ứng.

Cuối cùng, phương thức `__len__` cung cấp kích thước tổng thể của tập dữ liệu. Thông tin này cần thiết cho việc quản lý vòng lặp huấn luyện và đánh giá mô hình.

```
# Dataset class
class ShopeeDataset(torch.utils.data.Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = labels

    def __getitem__(self, idx):
        item = {key: torch.tensor(val[idx]) for key, val in
        ↪  self.encodings.items()}
        item['labels'] = torch.tensor(self.labels[idx])
        return item

    def __len__(self):
        return len(self.labels)
```

Tiếp theo ta sẽ chuẩn bị và huấn luyện một mô hình RoBERTa cho nhiệm vụ phân loại các đoạn văn bản sử dụng thư viện PyTorch và Transformers của Hugging Face. Đầu tiên, dữ liệu huấn luyện và kiểm thử được mã hóa bằng tokenizer, với việc cắt xén *(truncate)* và đệm *(padding)* để đảm bảo mọi chuỗi có độ dài nhất quán. Sau đó, hai tập dữ liệu train và test sẽ được tạo ra.

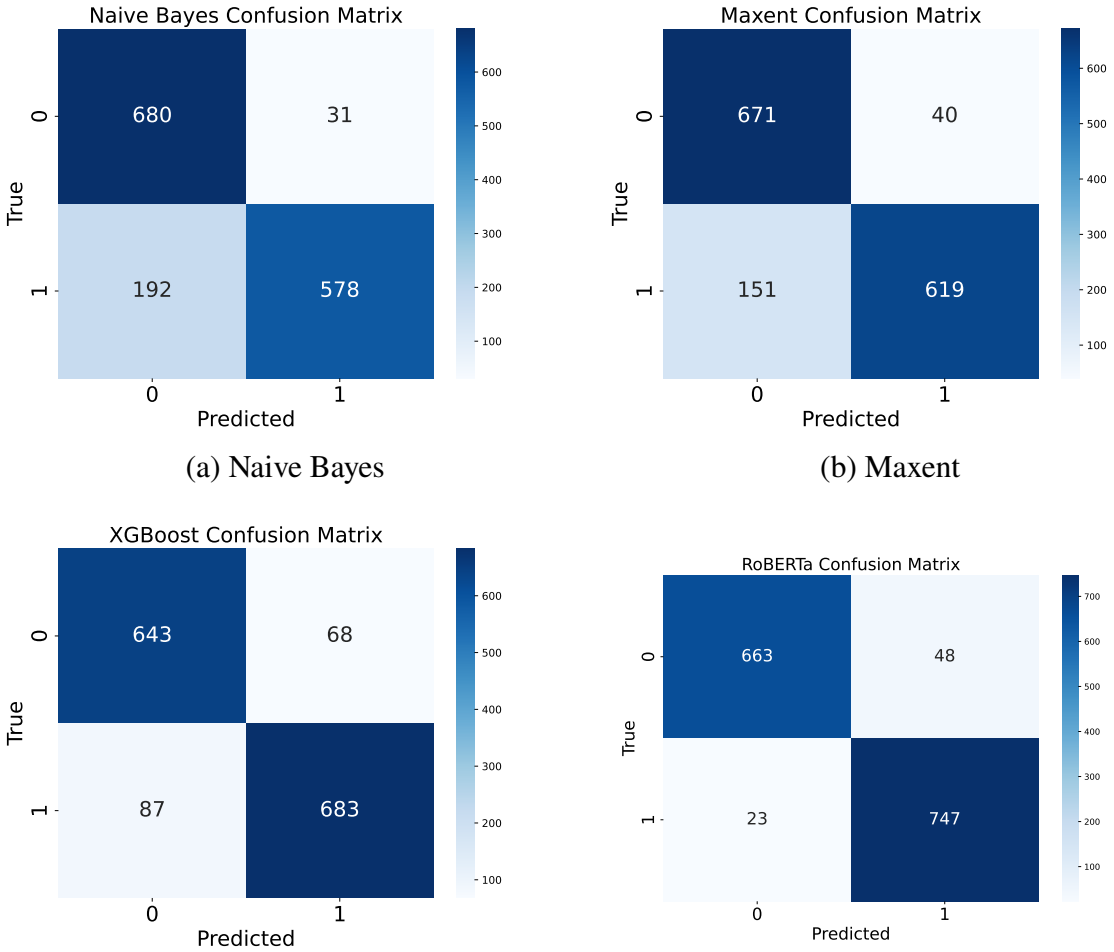
Tiếp theo, mô hình RoBERTa được tải về với các trọng số đã được huấn luyện trước. Các tham số huấn luyện như epoch, batch size, warmup_step và weight_decay được định nghĩa trong TrainingArguments.

Sau đó ta sẽ khởi tạo Trainer với mô hình RoBERTa, các tham số huấn luyện, và tập dữ liệu huấn luyện và đánh giá. Cuối cùng, phương thức trainer.train() được gọi để bắt đầu quá trình huấn luyện mô hình trên tập train với số epoch và các tham số huấn luyện khác đã được xác định.

```
# Tokenize the data
train_encodings = tokenizer(X_train.tolist(), truncation=True, padding=True)
test_encodings = tokenizer(X_test.tolist(), truncation=True, padding=True)

# Create datasets
train_dataset = ShopeeDataset(train_encodings, y_train.tolist())
test_dataset = ShopeeDataset(test_encodings, y_test.tolist())

# Load pre-trained model
model = RobertaForSequenceClassification.from_pretrained('roberta-base')

# Training arguments
training_args = TrainingArguments(
    output_dir='./results',
    num_train_epochs=3,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    warmup_steps=500,
    weight_decay=0.01,
    logging_dir='./logs',
)

# Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset
)

# Train the model
trainer.train()
```

5 CHƯƠNG 5: ĐÁNH GIÁ KẾT QUẢ

Thông qua các ma trận nhầm lẫn, mô hình RoBERTa là mô hình có tỷ lệ dự đoán đúng cao nhất, đặc biệt là trong việc dự đoán các nhãn Positive, xếp sau đó là mô hình XGBoost, Maxent, Naive Bayes.



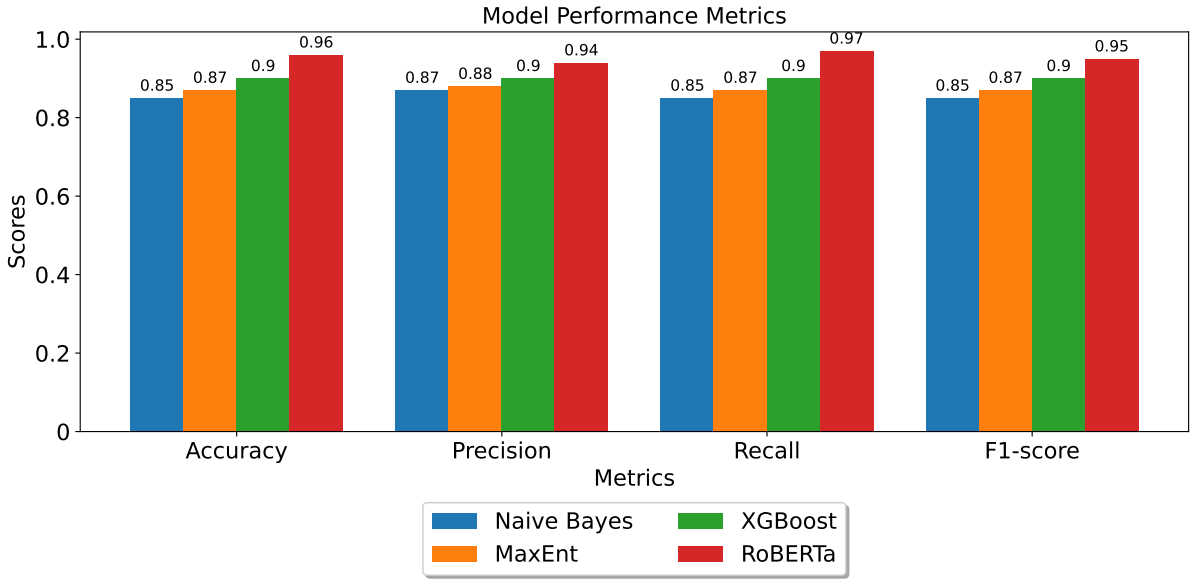
(a) Naive Bayes (b) Maxent

(c) XGBoost (d) RoBERTa

Hình 8: Ma trận nhầm lẫn thể hiện kết quả dự đoán của các mô hình

Dựa trên kết quả đánh giá mô hình thông qua các chỉ số là Accuracy, Precision, Recall, F1-score, nhóm đã có một cái nhìn tổng quan hơn về hiệu suất của các mô hình. Đứng đầu là mô hình RoBERTa với điểm số tốt nhất ở mỗi chỉ số và đều trên mức 0.9, Điều này cho thấy khả năng phân loại cao và hiệu suất ổn định của mô hình.

Tiếp theo mô hình XGBoost cũng ghi điểm với hiệu suất ổn định và đồng đều trong việc phân loại, XGBoost có kết quả gần bằng với RoBERTa, các chỉ số bằng nhau và ở mức 0.9.



Hình 9: Kết quả đánh giá các mô hình trong tác vụ Phân tích cảm xúc trên bộ dữ liệu Shopee Customer Data. Có thể thấy RoBERTa là mô hình có kết quả tốt nhất so với các mô hình máy học còn lại.

Hai mô hình còn lại, Naive Bayes và MaxEnt, cũng đem lại hiệu suất tốt với các chỉ số nằm trong khoảng từ 0.85 đến 0.87. Mặc dù không đạt được mức điểm số cao như RoBERTa và XGBoost, nhưng vẫn cho thấy khả năng phân loại khá tốt.

Model	**Metrics**			
	Accuracy	Precision	Recall	F1 Score
Naive Bayes [17]	0.85	0.87	0.85	0.85
MaxEnt [4]	0.87	0.88	0.87	0.87
XGBoost [6]	0.90	0.90	0.90	0.90
RoBERTa [16]	**0.96**	**0.94**	**0.97**	**0.95**

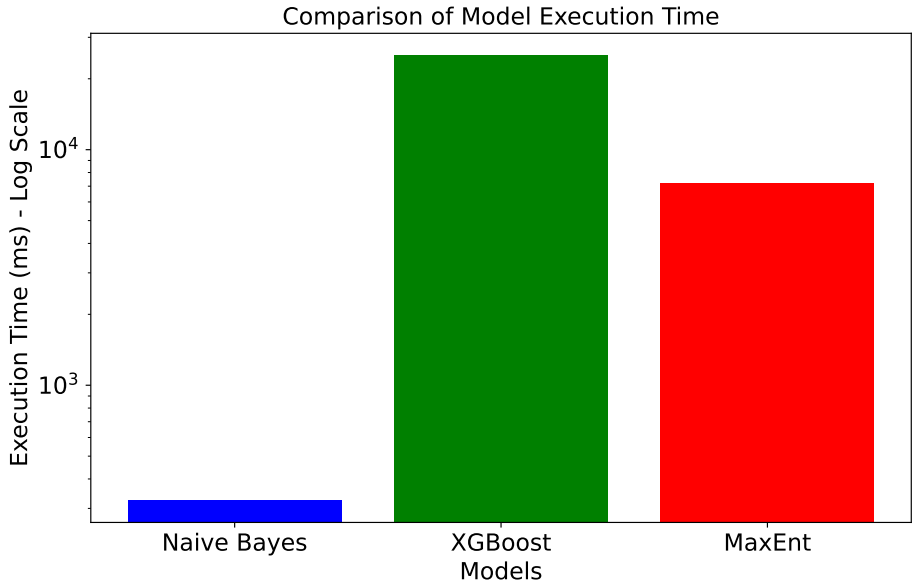
Bảng 2: Đánh giá các mô hình phân lớp

Tổng thể, nhóm nhận thấy rằng mô hình học sâu trong lĩnh vực NLP, như RoBERTa, thường mang lại kết quả tốt hơn so với các mô hình truyền thống như Naive Bayes và MaxEnt, đặc biệt trong việc xử lý và hiểu ngôn ngữ tự nhiên.

So sánh thời gian thực thi của các mô hình

Sau khi tiến hành phân tích và so sánh hiệu suất giữa các mô hình, nhóm tiếp tục đánh giá thời gian thực thi của các mô hình để xem xét mô hình nào cho kết quả tối ưu về cả

thời gian lẫn hiệu suất.



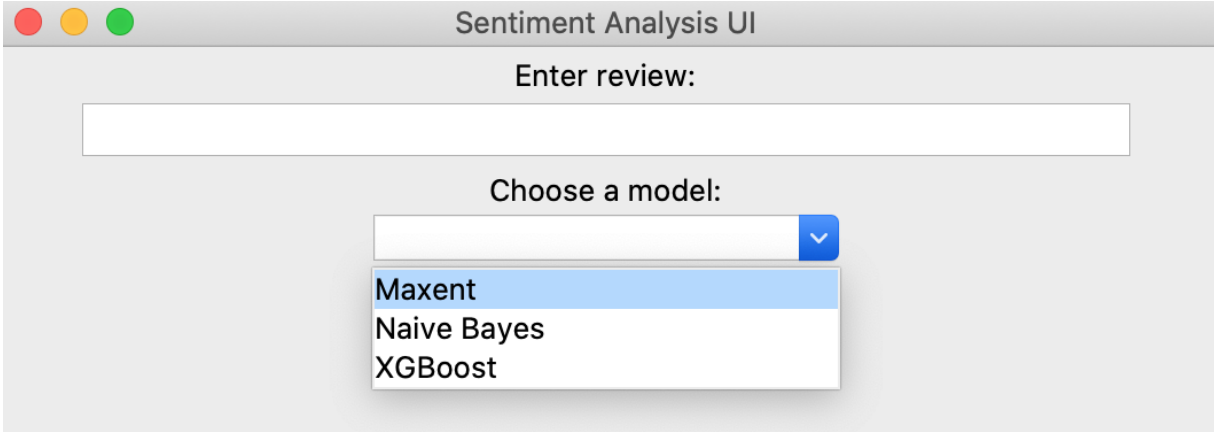
Hình 10: Thời gian thực thi các mô hình

Hình 10 cho thấy mô hình Naive Bayes có thời gian thực thi tối ưu nhất, thấp hơn hẳn so với 2 mô hình còn lại là XGBoost Models và Maxent. Điều này cho thấy mô hình Naive Bayes không chỉ có hiệu suất tốt trong việc nhận diện các đánh giá tiêu cực mà còn mang lại lợi ích về mặt thời gian, làm cho nó trở thành một sự lựa chọn hợp lý cho các ứng dụng yêu cầu cả hiệu suất và tối ưu hóa thời gian thực thi.

6 CHƯƠNG 6: TRIỂN KHAI VÀ ỨNG DỤNG MÔ HÌNH

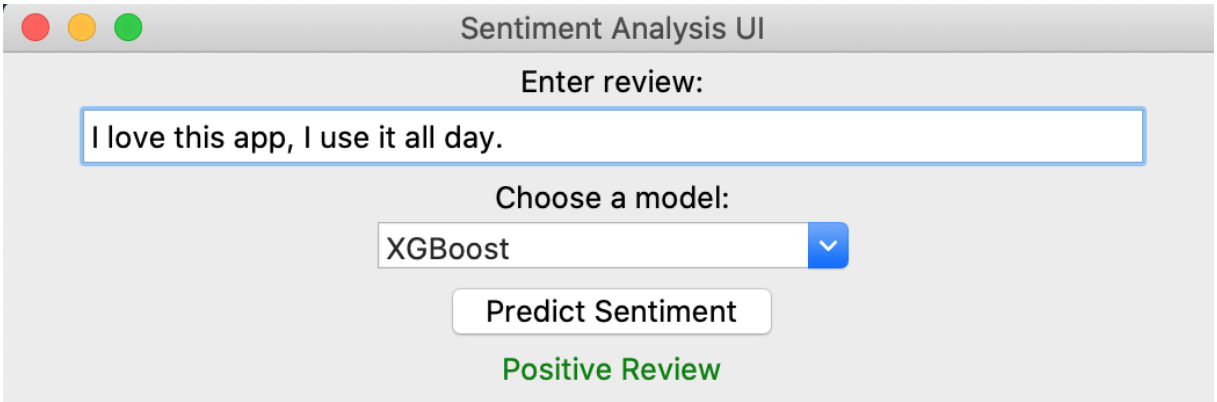
6.1 Triển khai và ứng dụng các mô hình máy học

Các mô hình máy học đã được huấn luyện sẽ được lưu dưới dạng file `pkl` để triển khai ứng dụng. Chi tiết code cài đặt ứng dụng nằm ở phần Phụ Lục A. Ứng dụng để triển khai các mô hình máy học có giao diện như sau:



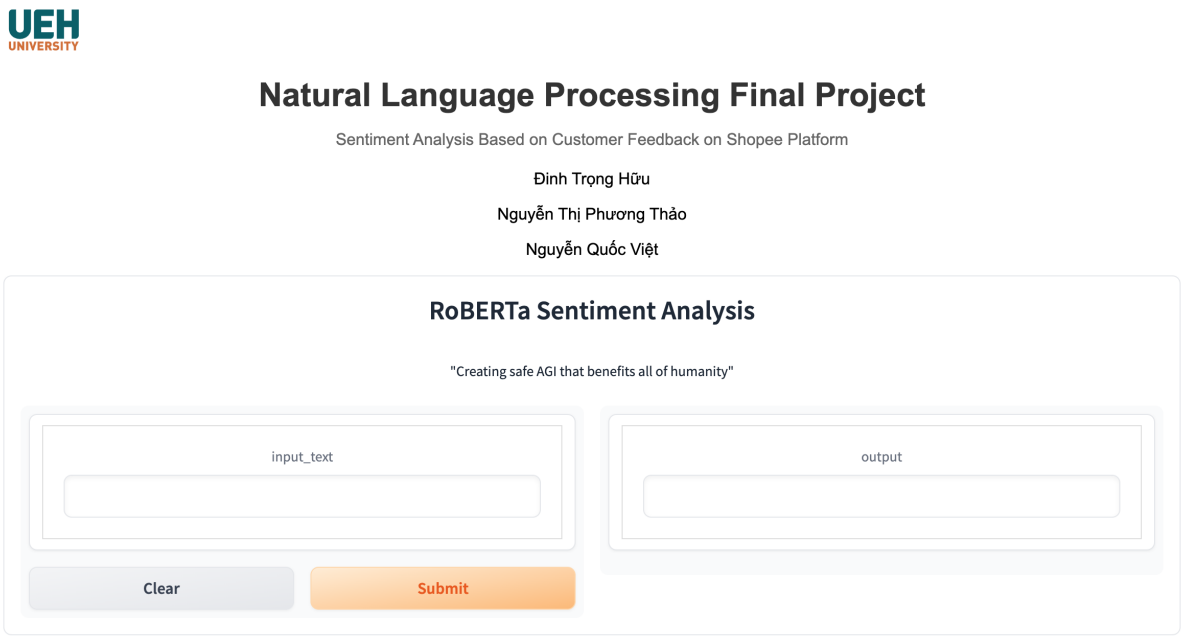
Hình 11: Ứng dụng triển khai các mô hình máy học cho phép người dùng nhập đánh giá trực tiếp và chọn mô hình dự đoán

Sau khi nhập vào câu đánh giá, lựa chọn mô hình, và nhấn nút Predict Sentiment, ứng dụng sẽ trả về kết quả dự đoán như sau:



6.2 Triển khai và ứng dụng các mô hình RoBERTa

Mô hình RoBERTa sau khi đã được huấn luyện trên bộ dữ liệu Shopee App Review sẽ được triển khai trên Hugging Space và tạo một giao diện web trực quan thông qua Gradio:



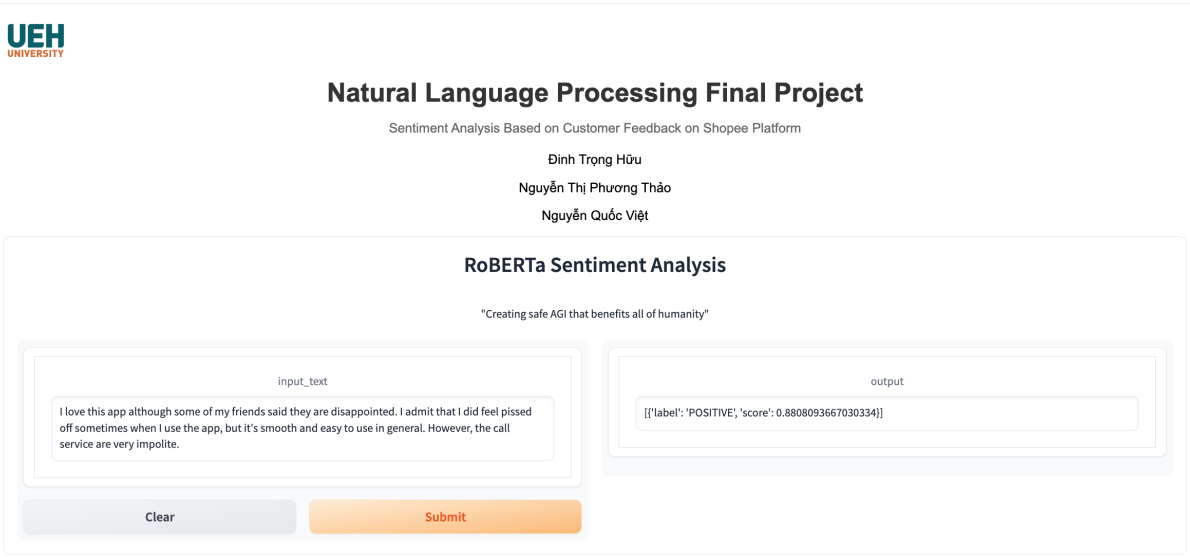
Hình 12: Website triển khai mô hình RoBERTa thông qua Gradio để thực hiện dự đoán từ Review được nhập trực tiếp từ người dùng. Truy cập website tại *https://ueh-nlp.github.io/*

Nhóm sẽ thực hiện thử với nội dung câu đánh giá như sau:

```
I love this app although some of my friends said they are
disappointed. I admit that I did feel pissed off sometimes when I
use the app, but it's smooth and easy to use in general. However, the
call service are very impolite.
```

Tạm dịch:

```
Tôi rất thích ứng dụng này mặc dù một vài người bạn của tôi bảo rằng
họ rất thất vọng. Tôi thừa nhận rằng đôi khi tôi rất tức giận khi sử
dụng ứng dụng, tuy nhiên nhìn chung thì nó vẫn mượt mà và dễ sử dụng.
Mặc dù vậy, đội ngũ hỗ trợ qua điện thoại rất bất lịch sự.
```



Mặc dù câu đánh giá được cố tình tạo ra với sự phức tạp trong ngữ nghĩa ("một vài người bạn của tôi bảo rằng họ rất thất vọng", "thừa nhận rằng đôi khi tôi rất tức giận", "đội ngũ hỗ trợ qua điện thoại rất bất lịch sự"), mô hình RoBERTa vẫn cho ra kết quả dự đoán câu đúng với nhãn `Positive` cho thấy khả năng hiểu được ngữ cảnh của mô hình. Trong khi đó, cũng cùng câu đánh giá trên, mô hình Maxent và Naive Bayes sẽ phân loại sai thành nhãn `Negative`, tuy nhiên, mô hình XGBoost sẽ cho kết quả phân loại đúng với nhãn `Positive`.

7 CHƯƠNG 7: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

7.1 Các kết quả đạt được

Thông qua đồ án, nhóm đã thực hiện việc thu nhập, tiền xử lý, huấn luyện các mô hình máy học cũng như học sâu, đồng thời xây dựng ứng dụng và web để triển khai các mô hình, đem đến một trải nghiệm trực quan cho người dùng.

Các kết quả đạt được dựa vào các chỉ số cũng rất ấn tượng, với tiêu biểu nhất là mô hình RoBERTa với độ chính xác lên đến 96%. Tuy nhiên, đánh đổi lại là thời gian để huấn luyện mô hình RoBERTa còn khá chậm. Các mô hình máy học cũng đạt được kết quả phân loại rất ấn tượng, với độ chính xác dao động trong khoảng 85-90%.

Về phần giao diện, nhóm triển khai xây dựng ứng dụng có thể cài đặt một cách trực tiếp trên máy cục bộ của người dùng đối với các mô hình máy học. Đối với mô hình RoBERTa, nhóm thực hiện triển khai mô hình thông qua Hugging Face Space đồng thời xây dựng website để người dùng trải nghiệm mô hình một cách trực quan tại https://ueh-nlp.github.io/.

7.2 Hướng phát triển

Hướng phát triển cho đề tài này là mở rộng phạm vi nghiên cứu từ Tiếng Anh sang Tiếng Việt, nhằm đáp ứng nhu cầu ngày càng tăng của cộng đồng người dùng Việt Nam. Điều này có thể được thực hiện bằng cách tận dụng các mô hình học sâu hiện đại được huấn luyện riêng trên tiếng Việt như URA-Llama [4], PhoBERT [18], ViSoBERT [19], SeaLLMs [21], VinaLlama [20] để thực hiện việc dịch thuật sang tiếng Anh và đưa vào các mô hình khác để dự đoán.

Ngoài ra, việc bổ sung các thông tin liên quan như các thuộc tính khác và nhóm người dùng là một hướng phát triển khác có thể cung cấp sự đa dạng và độ chi tiết hơn về đánh giá. Mô hình phân loại lúc này sẽ đánh giá dựa trên từng nhóm người dùng hoặc thuộc tính để tạo ra một hệ thống đánh giá phản hồi đa chiều và hiểu rõ hơn về nhu cầu và mong muốn cụ thể của từng đối tượng.

Ngoài ra, đối với mô hình RoBERTa, để tăng tốc độ huấn luyện mô hình cũng như giảm thiểu số tham số cần tối ưu, việc áp dụng các kĩ thuật như LoRA [13], QLoRA [9] cũng là một hướng phát triển tiềm năng. Thêm vào đó, một cách tiếp cận khác đối với bài toán phân tích cảm xúc là tận dụng khả năng học in-context với sự phát triển của các mô hình ngôn ngữ lớn [24, 5, 8, 23, 28]. Điều này giúp tiết kiệm chi phí và thời gian một cách đáng kể vì các mô hình không cần được phải huấn luyện từ đầu.

[4] https://huggingface.co/ura-hcmut/ura-llama-70b

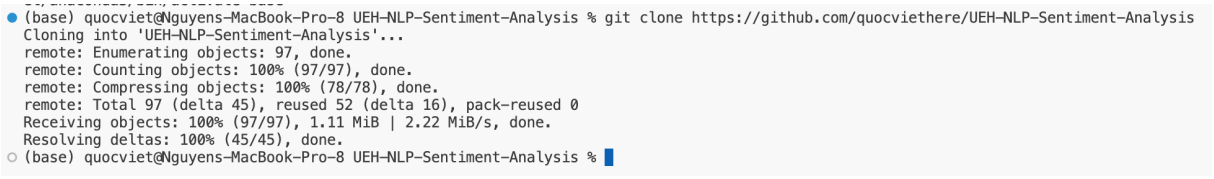
8 PHỤ LỤC

A Cài đặt giao diện ứng dụng các mô hình máy học

Đầu tiên người dùng cần truy cập GitHub và clone về repo tại https://github.com/quocviethere/UEH-NLP-Sentiment-Analysis bằng lệnh sau:

```
git clone https://github.com/quocviethere/UEH-NLP-Sentiment-Analysis
```

Sau khi cài đặt thành công, Terminal sẽ hiển thị như sau:



Sau đó nhập lệnh:

```
python app.py
```

Lúc này giao diện ứng dụng sẽ được tự động hiện ra.

B Mã nguồn Github

Toàn bộ Source Code thực hiện dự án và mô tả được đăng tải tại https://github.com/quocviethere/UEH-NLP-Sentiment-Analysis

C Bảng phân công

Thành viên	Phân công	Đánh giá
Đinh Trọng Hữu	Tiền xử lý và EDA dữ liệu Cài đặt và lý thuyết mô hình Naive Bayes Kết luận và hướng phát triển	100%
Nguyễn Thị Phương Thảo	Tiền xử lý và EDA dữ liệu Cài đặt và lý thuyết mô hình Maxent Đánh giá các thuật toán máy học	100%
Nguyễn Quốc Việt	Cài đặt và lý thuyết mô hình RoBERTa và XGBoost Cài đặt giao diện các mô hình máy học Làm web triển khai mô hình RoBERTa	100%

Tài liệu

[1] Anonymous (2023). itransformer: Inverted transformers are effective for time series forecasting. In *Submitted to The Twelfth International Conference on Learning Representations*. under review.

[2] Bayes, M. and Price, M. (1763). An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions (1683-1775)*, 53:370–418.

[3] Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

[4] Berger, A. L., Della Pietra, S. A., and Della Pietra, V. J. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

[5] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

[6] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

[7] Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., Belanger, D., Colwell, L., and Weller, A. (2022). Rethinking attention with performers.

[8] Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling instruction-finetuned language models.

[9] Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). QLoRA: Ef-

ficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[10] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[11] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[12] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009.

[13] Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

[14] Kim, D., Angelova, A., and Kuo, W. (2023). Region-aware pretraining for open-vocabulary object detection with vision transformers.

[15] Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Blanco, E. and Lu, W., editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

[16] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

[17] Maron, M. E. (1961). Automatic indexing: An experimental inquiry. *J. ACM*, 8(3):404–417.

[18] Nguyen, D. Q. and Tuan Nguyen, A. (2020). PhoBERT: Pre-trained language models for Vietnamese. In Cohn, T., He, Y., and Liu, Y., editors, *Findings of the*

Association for Computational Linguistics: EMNLP 2020, pages 1037–1042, Online. Association for Computational Linguistics.

[19] Nguyen, N., Phan, T., Nguyen, D.-V., and Nguyen, K. (2023a). ViSoBERT: A pre-trained language model for Vietnamese social media text processing. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5191–5207, Singapore. Association for Computational Linguistics.

[20] Nguyen, Q., Pham, H., and Dao, D. (2023b). Vinallama: Llama-based vietnamese foundation model.

[21] Nguyen, X.-P., Zhang, W., Li, X., Aljunied, M., Tan, Q., Cheng, L., Chen, G., Deng, Y., Yang, S., Liu, C., Zhang, H., and Bing, L. (2023c). Seallms – large language models for southeast asia.

[22] Nie, X., Zhou, X., Li, Z., Wang, L., Lin, X., and Tong, T. (2022). Logtrans: Providing efficient local-global fusion with transformer and cnn parallel network for biomedical image segmentation. In *2022 IEEE 24th Int Conf on High Performance Computing Communications; 8th Int Conf on Data Science Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud Big Data Systems Application (HPCC/DSS/SmartCity/DependSys)*, pages 769–776.

[23] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

[24] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners.

[25] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2020). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

[26] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

[27] Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. (2022). Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6).

[28] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models.

[29] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[30] Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. (2020). Linformer: Self-attention with linear complexity.

[31] Weaver, W. (1952). Translation. In *Proceedings of the Conference on Mechanical Translation*.

[32] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115.