

**ĐẠI HỌC UEH
TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ**

**KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH
CHUYÊN NGÀNH: KHOA HỌC DỮ LIỆU**



KHOÁ LUẬN TỐT NGHIỆP

**ĐỀ TÀI: PHÂN TÍCH SENTIMENT BÌNH LUẬN KHÁCH HÀNG
TRÊN GOOGLE MAPS CỦA CHUỖI HIGHLANDS COFFEE**

GIÁO VIÊN HƯỚNG DẪN: TS. Huỳnh Văn Đức
HỌ TÊN SINH VIÊN: Nguyễn Khánh Linh
MSSV: 31201024500
LỚP: DS001
KHOÁ: 46

TP. Hồ Chí Minh, Tháng 10/2023

LỜI CẢM ƠN

Với tất cả lòng biết ơn, trước hết em muốn gửi lời cảm ơn chân thành đến giảng viên hướng dẫn của em, thầy Huỳnh Văn Đức. Cảm ơn thầy vì đã luôn tận tâm và kiên nhẫn hướng dẫn em trong suốt quá trình nghiên cứu và viết khoá luận để em có thể vượt qua những thách thức và phát triển trong việc nghiên cứu.

Ngoài ra, em cũng muốn gửi lời cảm ơn đến tất cả thầy cô giảng dạy, người đã chia sẻ kiến thức và kinh nghiệm của mình cho em trong suốt quá trình học tập tại giảng đại học. Những bài giảng theo buổi học và những buổi thảo luận đã giúp cho em hiểu sâu hơn về lĩnh vực này.

Cuối cùng, em xin gửi lời cảm ơn đến chính bản thân mình, vì đã nỗ lực và làm việc chăm chỉ để hoàn thành khoá luận này. Em học được rất nhiều trong quá trình nghiên cứu này và hy vọng rằng bài nghiên cứu này sẽ mang lại giá trị thực sự.

Một lần nữa, em xin chân thành cảm ơn!

TÓM TẮT

Đề tài "*Phân tích ý kiến (sentiment) trong bình luận của khách hàng trên Google Maps về Highlands Coffee*" nhằm mục đích đánh giá và phân loại ý kiến của khách hàng một cách khách quan và đáng tin cậy. Việc này giúp cho doanh nghiệp hiểu rõ hơn về cách mà khách hàng cảm nhận về dịch vụ, sản phẩm và trải nghiệm tại các cửa hàng của họ.

Vì trang web chính thức của Highlands Coffee không cung cấp mục để khách hàng đánh giá, nên em đã chọn Google Maps là nơi thu thập ý kiến của khách hàng. Google Maps là một ứng dụng phổ biến được sử dụng rộng rãi, cung cấp nhiều thông tin về địa điểm, đánh giá và nhận xét từ người dùng.

Để thu thập dữ liệu trên Google Maps, ta sử dụng các công cụ như Selenium và BeautifulSoup. Sau đó, ta tiến hành xử lý bình luận bằng Vncorenlp và phân loại ý kiến trong những bình luận này. Trong đó, mô hình PhoBERT là mô hình lựa chọn phù hợp nhất cho việc này. PhoBERT là một biến thể của mô hình BERT (Bidirectional Encoder Representations from Transformers) đã được đào tạo trên văn bản tiếng Việt, và nó đã chứng minh khả năng xuất sắc trong việc phân loại Sentiment. Kết quả phân loại ý kiến sẽ giúp cho doanh nghiệp Highland tìm ra được các yếu tố ảnh hưởng đến cảm xúc, cảm nhận của khách hàng khi trải nghiệm sử dụng dịch vụ.

Hy vọng rằng điều này giúp mọi người hiểu rõ hơn về mục tiêu và phương pháp của đề tài "*Phân tích ý kiến Sentiment bình luận của khách hàng trên Google Maps của chuỗi Highlands Coffee.*"

MỤC LỤC

| | |
|--|-----------|
| LỜI CẢM ƠN | 2 |
| TÓM TẮT..... | 3 |
| DANH MỤC HÌNH ẢNH | 6 |
| DANH MỤC BẢNG BIỂU..... | 8 |
| DANH MỤC TỪ VIẾT TẮT | 9 |
| MỞ ĐẦU..... | 10 |
| 1. Giới thiệu: | 10 |
| 2. Mục tiêu và nhiệm vụ nghiên cứu: | 12 |
| 3. Đối tượng - phạm vi nghiên cứu: | 12 |
| 3.1. Đối tượng nghiên cứu:..... | 12 |
| 3.2. Phạm vi nghiên cứu:..... | 12 |
| 3.3. Sơ đồ nghiên cứu:..... | 13 |
| 4. Bố cục khoá luận: | 14 |
| CHƯƠNG 1: CƠ SỞ LÝ THUYẾT | 15 |
| 1.1. Công cụ Selenium: | 15 |
| 1.1.1. Selenium là gì? | 15 |
| 1.1.2. Một số tính năng nổi bật của Selenium: | 15 |
| 1.1.3. Những thành phần quan trọng của Selenium: | 16 |
| 1.2. Thư viện Beautiful Soup:..... | 18 |
| 1.3. VnCoreNLP:..... | 19 |
| 1.3. Mô hình word2vec: | 20 |
| 1.4. Mô hình phoBERT: | 23 |
| 1.4.1. Giới thiệu về Transformers: | 23 |
| 1.4.2. Giới thiệu về BERT:..... | 30 |

| | |
|--|-----------|
| 1.4.3. Giới thiệu về RoBERTa: | 31 |
| 1.4.4. Giới thiệu về phoBERT:..... | 31 |
| CHƯƠNG 2: NỘI DUNG THỰC HIỆN | 33 |
| 2.1. Thu thập dữ liệu:..... | 33 |
| 2.1.1. Cách thu thập dữ liệu:..... | 33 |
| 2.1.2. Dữ liệu thu thập được:..... | 37 |
| 2.2. Tiền xử lý: | 38 |
| 2.2.1. Xử lý Review Time: | 38 |
| 2.2.2. Xử lý Review Rate: | 39 |
| 2.2.3. Xử lý Review Comment:..... | 41 |
| 2.3 Mô hình PhoBERT - SENTIMENT:..... | 42 |
| 2.3.1. Lựa chọn mô hình:..... | 42 |
| 2.3.2. Thực nghiệm mô hình: | 44 |
| 2.4. Khai thác dữ liệu bình luận khách hàng: | 47 |
| 2.4.1. Rút trích các từ khoá dựa trên SETIMENT: | 50 |
| 2.4.2. Đôi chiêu kết hợp từ khoá dựa trên từng loại SENTIMENT: | 56 |
| CHƯƠNG 3: ĐÁNH GIÁ - ĐỀ XUẤT GIẢI PHÁP | 59 |
| 3.1. Đánh giá :..... | 59 |
| 3.2. Đề xuất giải pháp: | 61 |
| KẾT LUẬN..... | 62 |
| TÀI LIỆU THAM KHẢO..... | 63 |

DANH MỤC HÌNH ẢNH

| | |
|---|----|
| Hình 1: Sơ đồ tổng quan phương hướng giải quyết..... | 13 |
| Hình 2: Cùng tìm hiểu về Selenium | 15 |
| Hình 3: Selenium Webdriver | 16 |
| Hình 4: Selenium IDE (Integrated Development Environment) | 17 |
| Hình 5: Selenium Grid..... | 17 |
| Hình 6: Selenium Server..... | 18 |
| Hình 7: Minh họa Thư viện Beautiful Soup | 18 |
| Hình 8: Luồng xử lý của VnCoreNLP | 19 |
| Hình 9: Kết quả của ví dụ Skip-gram | 21 |
| Hình 10: Mô hình Word2Vec trainning trên Gensim | 22 |
| Hình 11: Kiến trúc mô hình Transformer | 24 |
| Hình 12: Biểu diễn nhúng từ..... | 25 |
| Hình 13: Mã hoá vị trí từ nhúng | 25 |
| Hình 14: Cơ chế Self-Attention | 26 |
| Hình 15: Multi-head Attention cho câu | 27 |
| Hình 16: Quá trình concat các Attention heads | 28 |
| Hình 17: Quá trình cộng với attention (z) và thực hiện Layer Normalization | 28 |
| Hình 18: Tính toán song song cho câu | 29 |
| Hình 19: Tính toán song song cho câu | 29 |
| Hình 20: BERT là một trong những thuật toán quan trọng trong việc tìm kiếm..... | 30 |
| Hình 21: Kết quả Selenium thành công điều hướng tới web..... | 34 |
| Hình 22: Sử dụng Selenium click vào khu vực dữ liệu cần lấy | 35 |
| Hình 23: Lấy dữ liệu tên khách hàng bằng thư viện BeautifulSoup dựa trên div class..... | 35 |
| Hình 24: Lấy tên khách hàng dựa vào aria-label | 36 |
| Hình 25: Lấy rating khách hàng | 36 |
| Hình 26: Lấy bình luận khách hàng | 36 |
| Hình 27: Hướng dẫn lấy dữ liệu từ phần đánh giá của tất cả quán Highland quận 10 | 37 |
| Hình 28: Dataset khi thu thập được | 37 |

| | |
|---|----|
| Hình 29: Dữ liệu còn lại khi drop cột "Reviewer Name" | 38 |
| Hình 30: Thông tin hiển thị cột giá trị thời gian | 38 |
| Hình 31: Tổng quan Rating trong 5 năm gần đây của Highland | 39 |
| Hình 32: Biểu đồ tình hình rating trung bình qua từng năm..... | 40 |
| Hình 33: Biểu đồ rating trung bình theo từng tháng từ 11/2022 đến 10/2023 | 41 |
| Hình 34: Xoá bỏ stopword..... | 42 |
| Hình 35: Tần xuất/Phần trăm của từng sentiment | 46 |
| Hình 37: Cột "Review Comment" sau khi được làm sạch và tách từ | 48 |
| Hình 38: Word Cloud các từ khoá từ khách hàng bình luận nhiều | 48 |
| Hình 39: Top 20 từ xuất hiện nhiều trong positive reviews | 51 |
| Hình 40: Các từ liên quan khi nhắc tới từ khoá xuất hiện nhiều trong positive reviews...51 | |
| Hình 41: Top 20 từ xuất hiện nhiều trong negative reviews | 52 |
| Hình 42: Các từ liên quan khi nhắc tới từ khoá xuất hiện nhiều trong negative reviews ..53 | |
| Hình 43: Top 20 từ xuất hiện nhiều trong neutral reviews | 54 |
| Hình 44: Các từ liên quan khi nhắc tới từ khoá xuất hiện nhiều trong neutral reviews55 | |
| Hình 45: Phần trăm xuất hiện của các từ khoá đại diện ở SENTIMENT positive | 57 |
| Hình 46: Phần trăm xuất hiện của các từ khoá đại diện ở SENTIMENT negative | 57 |
| Hình 47: Phần trăm xuất hiện của các từ khoá đại diện ở SENTIMENT neutral.....58 | |
| Hình 36: Biểu đồ cột tròn cho từng Sentiment theo từng Rating | 59 |

DANH MỤC BẢNG BIỂU

| | |
|---|----|
| Bảng 1: Các mô hình PhoBERT đã được đào tạo trước | 32 |
| Bảng 2: Thuộc tính và mô tả thuộc tính của bảng dữ liệu thu thập được | 37 |
| Bảng 3: Đánh giá phương pháp - mô hình SENTIMENT | 43 |
| Bảng 4: Top 20 từ khoá khách hàng nhắc đến nhiều nhất trong dữ liệu | 49 |
| Bảng 5: Giải thích các biến | 49 |
| Bảng 6: Top 10 từ khoá xuất hiện nhiều phân theo bình luận positive | 51 |
| Bảng 7: Top 10 từ khoá xuất hiện nhiều phân theo bình luận negative | 53 |
| Bảng 8: Top 10 từ khoá xuất hiện nhiều phân theo bình luận neutral | 55 |
| Bảng 9: Phân cụm khách hàng theo yếu tố đại diện từng SENTIMENT | 60 |

DANH MỤC TỪ VIẾT TẮT

| Từ viết tắt | Giải thích |
|-------------|---|
| BERT | Bidirectional Encoder Representations from Transformers |
| TXL | Tiền xử lý |
| IDE | Integrated Development Environment |
| VnCoreNLP | Vietnamese Core Natural Language Processing |

MỞ ĐẦU

1. Giới thiệu:

Trong những năm gần đây, sự tiến bộ nhanh chóng của công nghệ và sự phổ cập của Internet đã làm thay đổi cách mà doanh nghiệp tương tác với khách hàng một cách đáng kể. Khách hàng bây giờ không chỉ là người mua hàng, họ còn là những người có giọng nói mạnh mẽ và có khả năng lan truyền thông tin với tốc độ chóng mặt. Điều này trở nên đặc biệt quan trọng trong ngành thực phẩm, dịch vụ ẩm thực, thậm chí là trên các sàn thương mại điện tử - đồ thiết yếu.

Khách hàng ngày càng phụ thuộc vào đánh giá và nhận xét của các khách hàng trước đó để đưa ra quyết định mua hàng hoặc lựa chọn địa điểm ăn uống. Việc này đã tạo nên một hiện tượng mới, khiến cho các ý kiến và sự tương tác của khách hàng trở thành một luồng dữ liệu văn bản không ngừng, từng giờ và từng phút. Mỗi bài đánh giá, mỗi bình luận trên mạng xã hội đều mang trong mình giá trị thông tin đối với doanh nghiệp.

Tuy nhiên, việc xử lý và phân tích lượng lớn dữ liệu văn bản này đòi hỏi sự hỗ trợ mạnh mẽ từ các máy tính và công nghệ thông tin. Đây là nhiệm vụ khó khăn và phức tạp mà các nhà lãnh đạo công ty và các chuyên gia tiếp thị phải đối mặt. Họ cần hiểu rõ những cảm nhận thực sự của khách hàng về sản phẩm, dịch vụ và thương hiệu của họ để có thể điều chỉnh chiến lược kinh doanh và phản hồi nhanh chóng vào phản hồi của khách hàng.

Để đổi mới với thách thức này và tận dụng tối đa luồng dữ liệu văn bản từ khách hàng, chúng ta cần phải dựa vào sự trợ giúp mạnh mẽ từ các hệ thống máy tính và công nghệ phân tích dữ liệu. Điều này sẽ giúp chúng ta rút ra những thông tin quý báu từ những câu chuyện, đánh giá và bình luận của khách hàng, từ đó cải thiện sản phẩm, dịch vụ, và tạo ra trải nghiệm mua sắm tốt hơn cho họ.

Trên thực tế, việc thu thập và phân tích dữ liệu đánh giá và bình luận từ khách hàng trên các nền tảng như Google Maps là một bước quan trọng trong quá trình cải thiện mối quan hệ với khách hàng. Lấy ví dụ về Highland Coffee, một chuỗi cà phê nổi tiếng, việc theo dõi những đánh giá và nhận xét từ người tiêu dùng có thể giúp họ hiểu rõ hơn về cảm nhận thực sự của khách hàng về sản phẩm và dịch vụ của mình.

Nhưng vấn đề đặt ra là làm thế nào để hiệu quả trong việc thu thập và phân tích những ý kiến này? Làm thế nào để biết được khách hàng đang hài lòng và không hài lòng về sản phẩm và dịch vụ của Highland? Và quan trọng hơn, làm thế nào để tìm ra những insight từ dữ liệu này để cải thiện chất lượng và điểm rating?

Để giải quyết vấn đề đó em đề xuất áp dụng mô hình ngôn ngữ **PhoBERT** và phân tích dữ liệu để khai thác và phân loại các ý kiến của khách hàng về Highland trên Google Maps. Trong quá trình này, em sẽ tập trung vào dữ liệu từ các đánh giá, nhận xét của khách hàng trên **Google Maps** để hiểu rõ hơn về những điểm yếu cũng như điểm mạnh của chuỗi cà phê Highland tại quận 10. Mục tiêu không chỉ tập trung vào việc xác định các ý kiến tích cực và tiêu cực mà còn tìm hiểu xem khách hàng đang gặp những vấn đề gì, các yếu tố ảnh hưởng đến việc sử dụng dịch vụ của khách hàng để từ đó doanh nghiệp có thể cải thiện. Chính vì những lý do đó, em chọn đề tài “**Phân tích Sentiment bình luận khách hàng trên Google của chuỗi Highlands coffee**”.

2. Phát biểu bài toán:

Phân tích ý kiến (Sentiment Analysis), còn gọi là khai phá quan điểm người dùng, đang là một lĩnh vực đầy triển vọng và thu hút sự quan tâm rất lớn từ cộng đồng nghiên cứu và cả các nhà phát triển ứng dụng. Trong bối cảnh này, việc phân tích cảm tính từ thông tin chứa trong các bình luận và đánh giá của người tiêu dùng trở nên vô cùng quan trọng. Điều này giúp cho các doanh nghiệp như Highland có cái nhìn sâu hơn về cách khách hàng đánh giá và thể hiện cảm xúc về sản phẩm, dịch vụ, sự kiện, và nhiều khía cạnh khác của cuộc sống.

Cụ thể, bài toán phân loại quan điểm và đánh giá từ các khách hàng Highland có thể được coi như một nhiệm vụ phân tích ngữ nghĩa và cảm tính của văn bản. Mục tiêu chính ở đây là xác định các cảm xúc, như tích cực, tiêu cực hoặc trung tính, trong các bình luận của khách hàng. Máy học có thể đóng một vai trò quan trọng trong việc giải quyết bài toán này bằng cách xử lý đánh giá và bình luận từ khách hàng về việc sử dụng dịch vụ của Highland và trả về kết quả phân tích sentiment tương ứng cho mỗi đánh giá và bình luận.

Phân tích ý kiến của "tiếng nói của khách hàng" (Voice of Customer - VOC) có thể giải quyết nhiều vấn đề quan trọng cho doanh nghiệp như HighLand. Nó giúp HighLand nhận biết được những điểm mạnh và yếu của sản phẩm của họ, từ đó có cơ hội cải thiện và tối ưu hóa chất lượng sản phẩm để đáp ứng sự mong đợi của khách hàng. Nếu xuất hiện các đánh giá tiêu cực thường xuyên, HighLand có thể nhanh chóng đưa ra các biện pháp để giảm thiểu vấn đề và tạo sự hài lòng cho khách hàng. Đồng thời, phân loại cảm tính trong các bình luận cũng giúp HighLand nắm bắt xu hướng và sự phản ánh của thị trường. Điều này giúp họ nhận thấy sự thay đổi trong quan điểm của khách hàng theo thời gian và phát hiện sớm các vấn đề tiềm ẩn hoặc cơ hội mới. Nó cũng giúp HighLand xác định mức độ hài lòng của khách hàng và đo lường hiệu suất của chiến lược quản lý cảm tính và dịch vụ khách hàng của họ.

2. Mục tiêu và nhiệm vụ nghiên cứu:

Để phân tích bài toán, ta sẽ đi sâu vào những nhiệm vụ nghiên cứu sau:

- Xác định ý kiến tích cực, trung tính và tiêu cực từ khách hàng về Highland Coffee để hiểu rõ điểm mạnh và điểm yếu của sản phẩm và dịch vụ của họ.
- Tìm hiểu về những vấn đề mà khách hàng đang gặp phải khi sử dụng dịch vụ của Highland, bao gồm cả các yếu tố ảnh hưởng đến trải nghiệm của họ.
- Đề xuất các biện pháp cải thiện dựa trên kết quả phân tích, nhằm tăng cường mối quan hệ với khách hàng và cải thiện điểm rating của Highland trên nền tảng Google Maps.

3. Đối tượng - phạm vi nghiên cứu:

3.1. Đối tượng nghiên cứu:

Khách hàng của chuỗi cà phê Highland: Khách hàng là nguồn thông tin quan trọng trong nghiên cứu này. Những ý kiến và đánh giá của họ trên Google Maps sẽ được thu thập và phân tích để đánh giá sự thỏa mãn và đề xuất cải tiến.

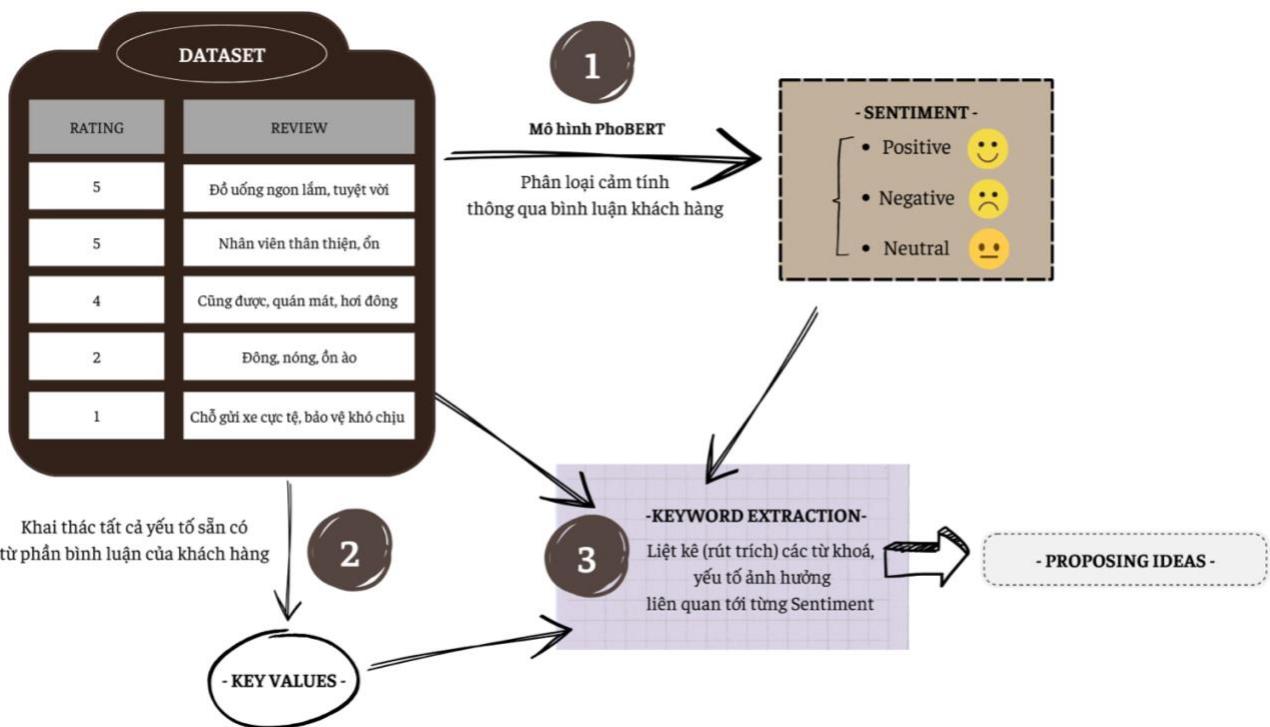
3.2. Phạm vi nghiên cứu:

1. Về không gian: Chuỗi cà phê Highland tại quận 10: Các cửa hàng cà phê Highland trong phạm vi quận 10 của Thành Phố Chí Minh (tổng cộng 8 quán).

2. *Về thời gian*: Từ tháng 9 năm 2018 đến tháng 9 năm 2023. (Trong 5 năm)

3.3. Sơ đồ nghiên cứu:

Để giải quyết bài toán này, ta cần phải chia nhỏ bài toán thành 3 phần như hình dưới đây. Đây là một mô hình tổng quan về quá trình nghiên cứu, cũng như quá trình giải quyết của bài toán.



Hình 1: Sơ đồ tổng quan phương hướng giải quyết

Bước 1 của quá trình này, ta sẽ thực hiện việc tách từ dữ liệu bình luận của khách hàng bằng **VnCoreNLP**, sau đó đưa chúng vào mô hình **PhoBERT** để tiến hành **phân loại cảm tính (tích cực - tiêu cực - trung tính)**. Kết quả từ mô hình sẽ cung cấp một đánh giá cảm xúc phù hợp cho mỗi bình luận.

Bước tiếp theo (2), là phân tích dữ liệu từ các bình luận của khách hàng để **tìm ra các yếu tố và từ khóa phổ biến** mà họ thường sử dụng khi đánh giá sản phẩm hoặc dịch vụ của mình. Quá trình này giúp ta có thể xác định rõ các vấn đề mà đa số khách hàng quan tâm.

Tiếp đến (3), ta sẽ kết hợp kết quả Sentiment thu được từ bước 1 với tất cả yếu tố ta xác định được ở bước 2. Chúng ta tiến hành **liệt kê và xác định các yếu tố ảnh hưởng tới từng cảm tính của khách hàng.** Dựa vào những thông tin này, chúng ta có thể đưa ra các gợi ý đề xuất cũng như phương hướng cải thiện dịch vụ tại Highland.

4. Bố cục khoá luận:

Ngoài các phần như lời cảm ơn, phần mở đầu, phần kết luận gồm các chương:

Chương 1. Cơ sở lý thuyết.

Hệ thống hoá kiến thức được sử dụng trong đề tài.

Chương 2. Nội dung thực hiện.

Trình bày cách thu thập dữ liệu, phương pháp giải quyết bài toán. Trong đó, tiến hành phân tích và phân loại các bình luận và đánh giá của khách hàng về Highland, đồng thời xác định ra những yếu tố đã tạo nên điểm mạnh và điểm yếu của Highland từ góc nhìn của khách hàng.

Chương 3. Đánh giá - Đề xuất giải pháp phát triển.

Dựa vào các kết quả phân tích được ở chương 2, đề xuất một số giải pháp gợi ý phương hướng phát triển.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. Công cụ Selenium:

1.1.1. Selenium là gì?



Hình 2: Cùng tìm hiểu về Selenium

Selenium là một bộ công cụ kiểm thử tự động mã nguồn mở, chuyên dành cho kiểm thử ứng dụng web. Nó hỗ trợ nhiều trình duyệt và hệ điều hành khác nhau như Windows, Mac, Linux và nhiều nền tảng khác. Selenium cho phép bạn viết các test script bằng nhiều ngôn ngữ lập trình khác nhau như Java, PHP, C#, Ruby, Python, hoặc thậm chí là Perl.

1.1.2. Một số tính năng nổi bật của Selenium:

Trình duyệt đa nền tảng: Selenium hỗ trợ nhiều trình duyệt phổ biến như Chrome, Firefox, Safari, và Edge, cho phép bạn kiểm tra tính tương thích của ứng dụng web trên nhiều nền tảng.

Tương tác đa dạng: Bạn có thể lập trình Selenium để thực hiện các thao tác như mở trình duyệt, truy cập các liên kết, điền dữ liệu vào các trường nhập, lấy thông tin từ trang web, tải lên hoặc tải xuống dữ liệu từ trang web, và thậm chí tương tác với các thành phần web như nút, biểu mẫu, và hộp thoại.

Kiểm thử tự động: Selenium thường được sử dụng cho kiểm thử tự động ứng dụng web, giúp đảm bảo rằng ứng dụng hoạt động đúng cách trên mọi trình duyệt và nền tảng.

Tùy chỉnh linh hoạt: Selenium cho phép bạn tùy chỉnh và mở rộng nó để đáp ứng các yêu cầu cụ thể của dự án. Bạn có thể thêm các tiện ích mở rộng và plugin để mở rộng khả năng của Selenium.

Tự động hóa công việc đơn điệu: Ngoài kiểm thử, bạn cũng có thể xây dựng các dự án để tự động hóa các công việc đơn điệu và lặp đi lặp lại, như việc tự động đăng nhập vào các trang web hàng ngày hoặc thực hiện các tác vụ lặp đi lặp lại trên trình duyệt.

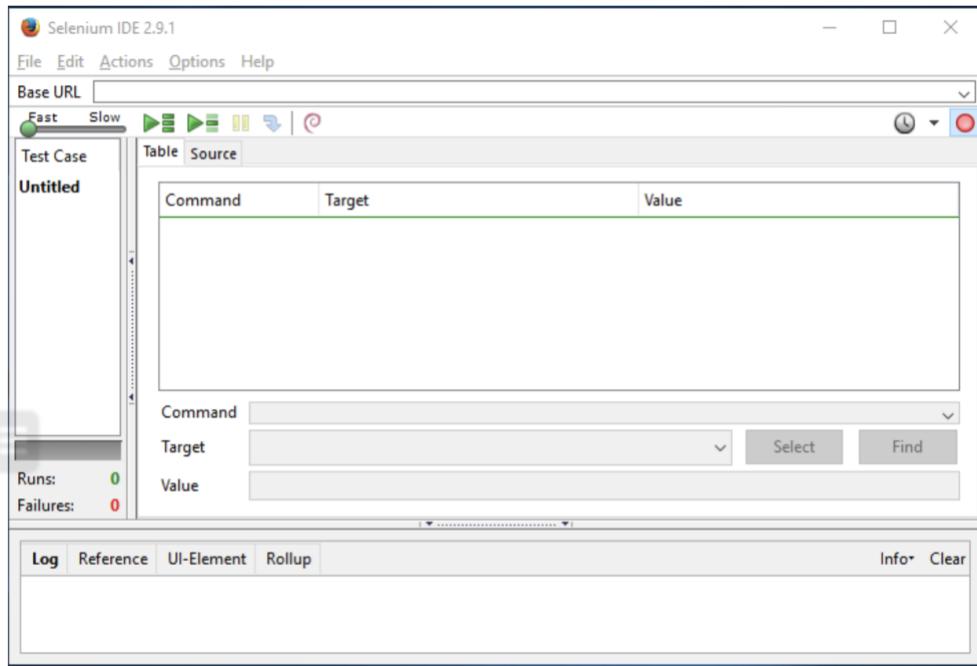
1.1.3. Những thành phần quan trọng của Selenium:

1. Selenium WebDriver: là một API cho phép bạn tương tác với trình duyệt web. Nó cung cấp các phương thức để điều khiển các hoạt động trên trình duyệt như mở một trang web, điều hướng, điền thông tin vào các trường văn bản, và kiểm tra nội dung trang web.



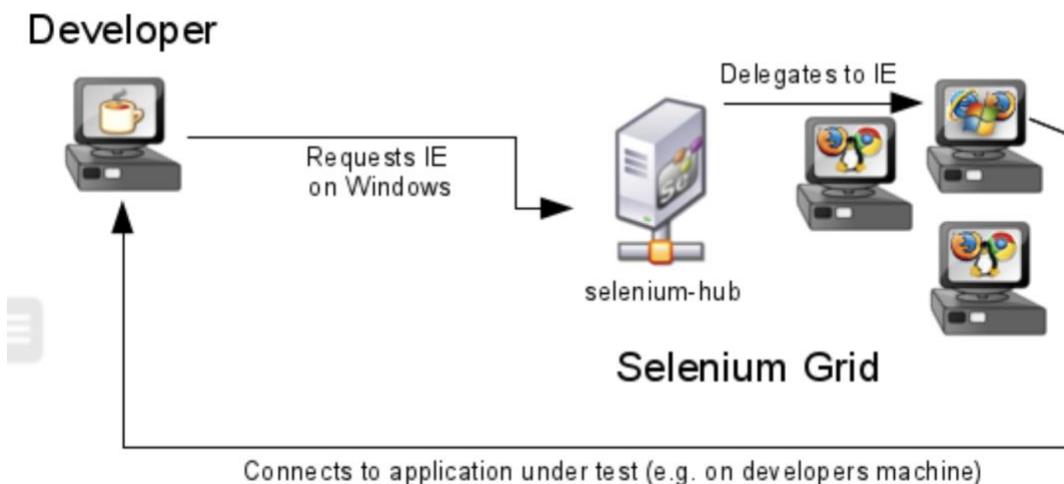
Hình 3: Selenium Webdriver

2. Selenium IDE: Selenium IDE (Integrated Development Environment) là một tiện ích mở rộng cho trình duyệt web, cho phép bạn ghi và chạy các tác vụ kiểm thử trên trình duyệt một cách dễ dàng. Nó thường được sử dụng cho kiểm thử thử nghiệm nhanh chóng và ghi lại các tương tác người dùng.



Hình 4: Selenium IDE (Integrated Development Environment)

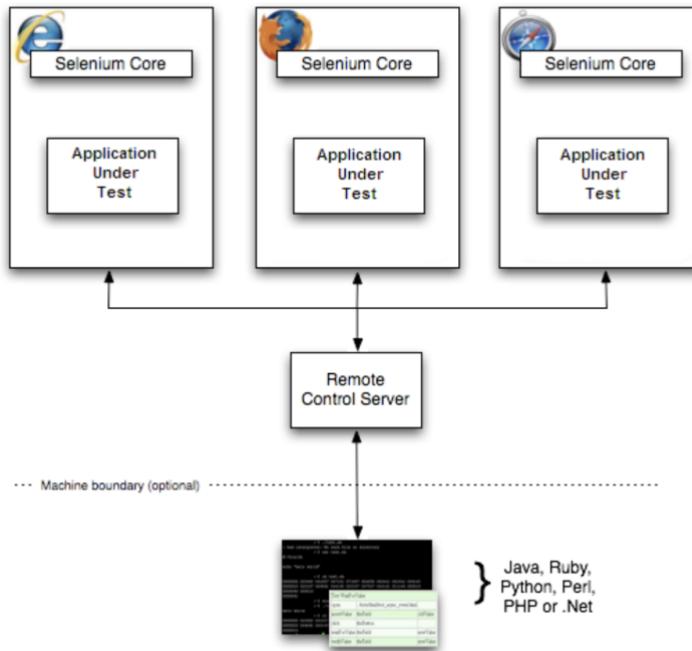
3. Selenium Grid: Selenium Grid cho phép bạn chạy các kiểm thử đồng thời trên nhiều trình duyệt và máy tính từ xa. Điều này giúp tăng hiệu suất và tiết kiệm thời gian trong việc kiểm thử trên nhiều môi trường khác nhau.



Hình 5: Selenium Grid

4. Selenium Server: (hay còn gọi là Selenium RC - Remote Control) là một máy chủ trung tâm cho việc kiểm thử tự động. Nó giúp quản lý các yêu cầu từ các phiên bản Selenium WebDriver khác nhau và điều khiển các trình duyệt từ xa.

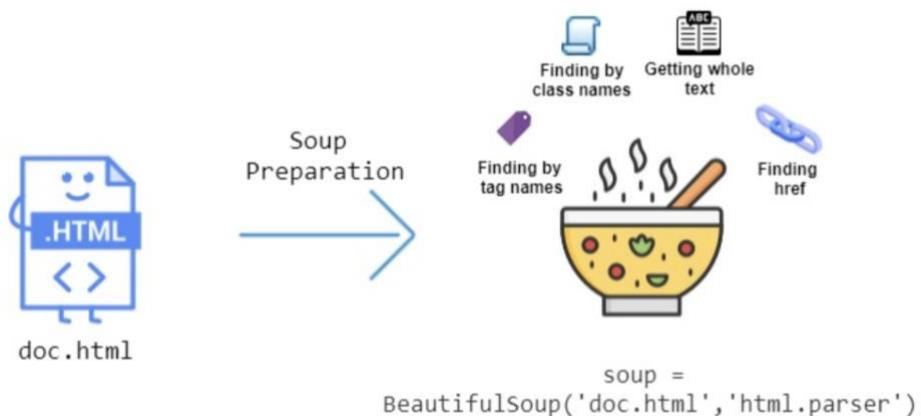
Windows, Linux, or Mac (as appropriate)...



Hình 6: Selenium Server

1.2. Thư viện Beautiful Soup:

BeautifulSoup là một thư viện Python mạnh mẽ được sử dụng để trích xuất thông tin từ các tài liệu HTML và XML. Thư viện này hoạt động phối hợp với các trình phân tích cú pháp (parser), giúp bạn dễ dàng duyệt, tìm kiếm và chỉnh sửa dữ liệu trong cây phân tích (parse tree) được tạo từ tài liệu ban đầu. Nhờ vào tích hợp với các parser này, BeautifulSoup đã giúp các nhà phát triển tiết kiệm rất nhiều thời gian và công sức trong công việc xử lý và trích xuất thông tin từ các tài liệu HTML và XML.

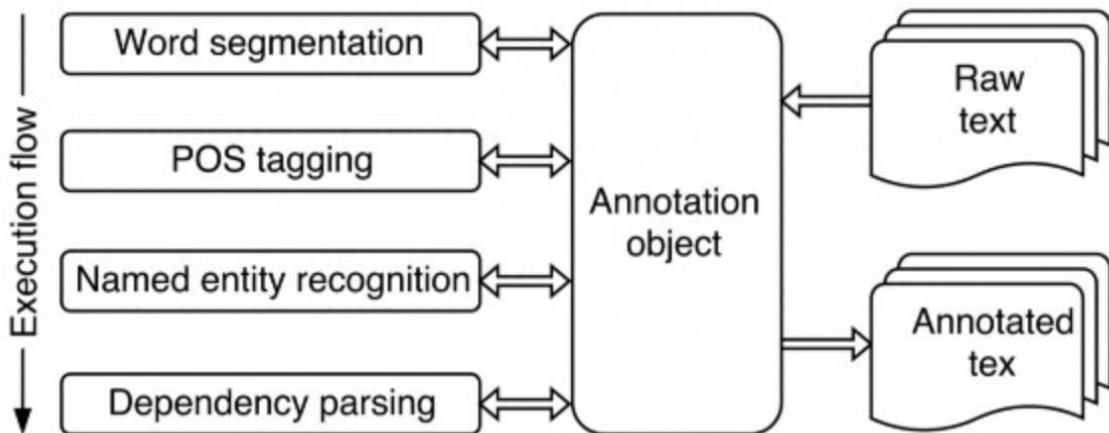


Hình 7: Minh họa Thư viện Beautiful Soup

1.3. VnCoreNLP:

VNCorenLP (Vietnamese Core Natural Language Processing) là một thư viện xử lý ngôn ngữ tự nhiên cho tiếng Việt được phát triển bởi các nhà nghiên cứu tại trường Đại học Khoa học Tự nhiên, Đại học Quốc gia TP.HCM và FPT Software. Thư viện này cung cấp các công cụ và tài liệu để thực hiện các nhiệm vụ liên quan đến xử lý ngôn ngữ tự nhiên cho tiếng Việt, bao gồm:

- Phân loại từ loại.
- Phân tích cú pháp.
- Phân đoạn văn bản.
- Gán thực thể và gán thời gian.



Hình 8: *Luồng xử lý của VnCoreNLP*

Đây là thư viện quan trọng trong mô hình phoBERT, trong tài liệu hướng dẫn sử dụng mô hình PhoBERT từ `wonrax/phobert-base-vietnamese-sentiment`*, người tạo mô hình đã khuyên nghị rằng **nếu văn bản đầu vào là dữ liệu thô, thì nên qua bước TXL (tách từ) bằng VnCoreNLP để đảm bảo tính nhất quán và hiệu suất tốt nhất**. Điều này là do PhoBERT đã sử dụng RDRSegmenter từ VnCoreNLP trong quá trình tiền xử lý dữ liệu huấn luyện, bao gồm chuẩn hóa âm điệu tiếng Việt và phân đoạn từ và câu. Để sử dụng VnCoreNLP tách từ tiếng Việt trước khi mã hoá, ta sẽ phải cài đặt môi trường và gói mô hình ngôn ngữ của nó (<https://github.com/vncorenlp/VnCoreNLP>) và giải nén vào một thư mục.

1.3. Mô hình word2vec:

Word2Vec là một mô hình học máy phổ biến được sử dụng để biểu diễn từ vựng của ngôn ngữ tự nhiên trong không gian vector có chiều thấp. Mô hình này được giới thiệu bởi Tomas Mikolov và đồng nghiệp tại Google vào năm 2013. Word2Vec đã thay đổi cách chúng ta làm việc với xử lý ngôn ngữ tự nhiên (NLP) bằng cách tạo ra các biểu diễn vector cho từng từ trong văn bản. Điều này có nhiều ứng dụng hữu ích trong NLP, bao gồm:

1. Tìm kiếm từ vựng tương tự: Word2Vec cho phép tìm kiếm các từ có ý nghĩa tương tự bằng cách tính khoảng cách vector giữa chúng. Ví dụ, "vua" và "nữ hoàng" có thể có biểu diễn vector gần nhau hơn so với "vua" và "bàn ăn".

2. Xác định các mối quan hệ từ vựng: Word2Vec có khả năng bắt các mối quan hệ ngữ nghĩa giữa các từ, ví dụ như "vua - nam + nữ = nữ hoàng." Bằng cách thực hiện các phép tính tương tự, bạn có thể tạo ra các mối quan hệ ngữ nghĩa như "hoàng tử" - "nam" + "nữ" = "công chúa."

3. Cải thiện hiệu suất của các tác vụ NLP: Word2Vec thường được sử dụng để cải thiện hiệu suất cho nhiều tác vụ NLP, bao gồm phân loại văn bản, gom cụm, phân tích ý kiến, và nhiều tác vụ khác.

❖ Word2Vec có hai phương pháp chính:

1. Skip-gram: Mô hình này cố gắng dự đoán các từ liên quan đến một từ mục tiêu bằng cách sử dụng nó làm ngữ cảnh. Điều này giúp tạo ra biểu diễn tốt cho các từ bằng cách học từ các từ xung quanh. Thường hoạt động tốt hơn trong việc học vector từ cho các từ hiếm và hiệu quả hơn với các **tập dữ liệu nhỏ**.

Giả sử chúng ta có một đoạn văn như sau: "*Tôi muốn một chiếc cốc màu xanh đựng hoa quả đậm.*" Để tạo một phép nhúng từ tốt hơn, chúng ta sẽ chọn một tập hợp các từ làm bối cảnh (context) thông qua việc xác định các từ mục tiêu (target) dựa trên từ bối cảnh. Chẳng hạn, nếu chúng ta chọn từ "cốc" làm từ bối cảnh, thì các từ lân cận được sẽ là như sau:

| Bối cảnh (context) | Mục tiêu (target) |
|--------------------|-------------------|
| cốc | màu_xanh |
| cốc | chiếc |
| cốc | một |
| cốc | muốn |

Hình 9: Kết quả của ví dụ Skip-gram

Tương tự như các phương pháp tiếp cận thông thường khác, mô hình của chúng ta biểu diễn từ vựng dưới dạng một vector one-hot. Vector này sẽ được sử dụng làm đầu vào cho một mạng nơ-ron với một tầng ẩn có 300 đơn vị. Tại tầng output, chúng ta sử dụng một hàm softmax để tính xác suất phân phối từ mục tiêu cho tất cả các từ trong từ vựng (tổng cộng 10,000 từ). Thông qua quá trình feed forward và back propagation, mô hình tối ưu hóa các tham số để dự đoán từ mục tiêu một cách chính xác nhất có thể. Khi quay trở lại tầng ẩn, chúng ta thu được đầu ra tại tầng này, đó là ma trận nhúng $\mathbf{E} \in \mathbb{R}^{n \times 300}$.

$$\mathbf{o}_c \rightarrow \mathbf{E} \rightarrow \mathbf{e}_c \rightarrow \text{softmax} \rightarrow \hat{\mathbf{y}}$$

$\mathbf{e}_c \in \mathbb{R}^{300}$ là véc tơ nhúng trích xuất từ ma trận \mathbf{E} tương ứng với từ bối cảnh c . $\hat{\mathbf{y}}$ là xác suất được dự báo của từ mục tiêu.

Ma trận nhúng là một tập hợp các vector nhúng, mỗi vector tương ứng với một từ trong ngữ cảnh. Xác suất được dự đoán cho từ mục tiêu được biểu diễn bởi một vector xác suất. Khi áp dụng hàm softmax, các giá trị xác suất ở đầu ra được tính theo công thức:

$$P(t = v_i | c) = \frac{e^{\theta_i^T e_c}}{\sum_{j=1}^{10000} e^{\theta_j^T e_c}}$$

Ở đây, $\theta_i \in \mathbb{R}^{300}$ là một vector chứa các tham số, biểu thị sự kết nối giữa các đơn vị ở tầng ẩn và tầng output.

Kết quả dự đoán của mô hình mạng nơ-ron càng chính xác, thì ma trận nhúng càng phản ánh mối quan hệ thực tế giữa từ trong ngữ cảnh và từ mục tiêu. Điều quan trọng là chúng ta quan tâm đặc biệt đến các hàng trong ma trận nhúng \mathbf{E} . Chúng là các vector nhúng \mathbf{e}_c đại diện cho từng từ trong ngữ cảnh c .

2. **CBOW** (Continuous Bag of Words): Mô hình này ngược lại, cố gắng dự đoán từ mục tiêu dựa trên các từ ngữ cảnh xung quanh nó. Nó thường nhanh hơn trong việc học. trên các **tập dữ liệu lớn** và hoạt động tốt khi bạn có ít dữ liệu.

Chúng ta có thể nhận thấy rằng mô hình Skip-grams có thể trở nên rất tốn chi phí tính toán vì phải tính toán xác suất cho nhiều từ mục tiêu khác nhau, và điều này dẫn đến sự phức tạp. Để giảm bớt chi phí tính toán, chúng ta có thể sử dụng mô hình CBOW (Continuous Bag of Words). CBOW hoạt động ngược lại so với Skip-grams. Trong CBOW, các từ trong ngữ cảnh xung quanh từ mục tiêu được sử dụng để dự đoán từ mục tiêu, và ngược lại với Skip-grams.

Kiến trúc mạng nơ-ron của CBOW bao gồm 3 lớp chính:

1. Lớp Input: Các từ xung quanh từ mục tiêu trong ngữ cảnh.
2. Lớp Projection: Lớp này tính trung bình của các vectơ biểu diễn của các từ đầu vào để tạo ra một vectơ đặc trưng.
3. Lớp Output: Lớp này sử dụng hàm softmax để dự đoán xác suất xuất hiện của từ mục tiêu.

❖ Sử dụng Gensim cho mô hình Word2Vec:

Ngoài ra, chúng ta có thể sử dụng thư viện Gensim để dễ dàng huấn luyện mô hình Word2Vec với chỉ vài dòng mã đơn giản.

Dưới đây là một ví dụ về cách huấn luyện mô hình Word2Vec sử dụng Gensim, và có một số tham số quan trọng cần lưu ý:

```
from gensim.models import Word2Vec
# Training model với 1000 câu đầu tiên trong kinh thánh
sentences = [[item.lower() for item in doc.split()] for doc in norm_bible[:1000]]
model = Word2Vec(sentences, min_count = 1, size = 150, window = 10, sg = 1, workers = 8)
model.train(sentences, total_examples = model.corpus_count, epochs = 10)

(210070, 336740)
```

Hình 10: Mô hình Word2Vec training trên Gensim

Trong đó, tham số quan trọng trong Word2Vec như sau:

- **size**: Tham số này xác định kích thước của ma trận nhúng, tức là số chiều của vector biểu diễn cho mỗi từ.

- **window**: Kích thước của cửa sổ sử dụng để xác định ngữ cảnh của các từ trong văn bản. Nó quyết định các từ n-gram mà mô hình sẽ sử dụng.

- **sg**: Tham số này nhận hai giá trị {0, 1}. Nếu sg là 0, mô hình sử dụng phương pháp Continuous Bag of Words (CBOW), trong khi nếu sg là 1, mô hình sử dụng phương pháp Skip-grams.

- **workers**: Số lượng core CPU mà bạn muốn sử dụng để huấn luyện. Số lượng core này có thể tăng tốc độ huấn luyện, đặc biệt là trên các tập dữ liệu lớn.

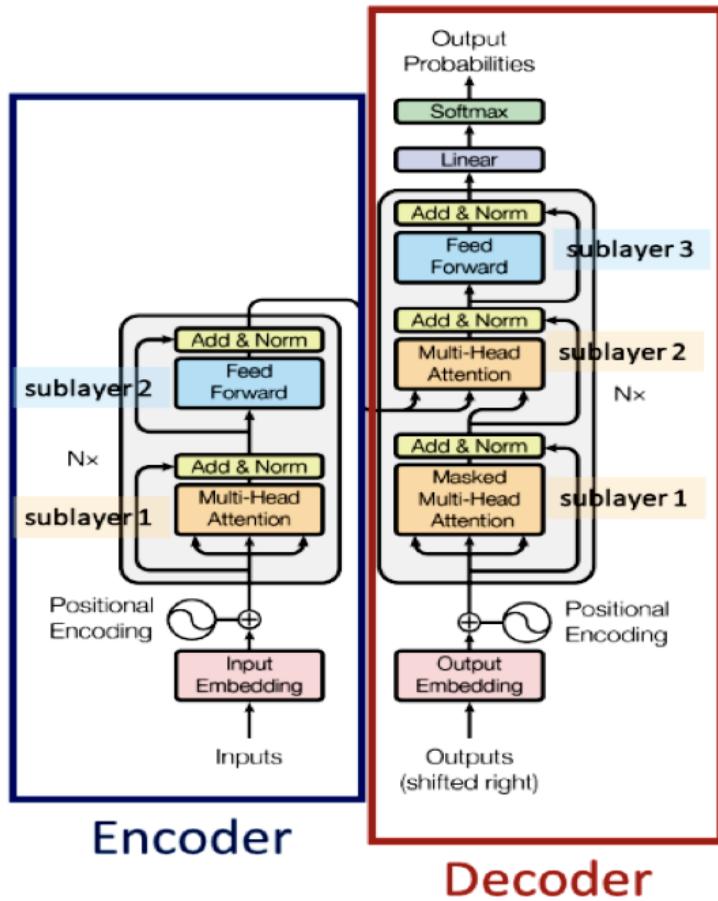
1.4. Mô hình phoBERT:

PhoBERT là một biến thể của BERT, được tạo ra dành riêng cho ngôn ngữ tiếng Việt. Mô hình BERT gốc đã được đào tạo trên dữ liệu tiếng Anh và thường không hoạt động tốt trên các ngôn ngữ khác. Để giải quyết vấn đề này, các nhà nghiên cứu và các nhóm phát triển đã **tạo ra các biến thể của BERT** được điều chỉnh cho từng ngôn ngữ cụ thể.

BERT dựa trên kiến trúc Transformers. Transformers là một kiến trúc mạng nơ-ron sâu rất mạnh mẽ cho xử lý ngôn ngữ tự nhiên và nhiều nhiệm vụ khác. BERT (và các biến thể của nó, bao gồm PhoBERT) sử dụng kiến trúc này để học biểu diễn từ và văn bản thông qua việc đào tạo trên dữ liệu lớn, giúp cải thiện hiểu biết và hiệu suất trong nhiều ứng dụng NLP.

1.4.1. Giới thiệu về Transformers:

Trong kiến trúc Transformer, việc trích xuất đặc trưng của các đối tượng được thực hiện thông qua cơ chế tự-chú ý (self-attention), giúp mã hóa thông tin về ngữ cảnh. Kiến trúc này chia thành hai phần chính: **bộ mã hóa (encoder)** ở bên trái và **bộ giải mã (decoder)** ở bên phải. Bộ mã hóa dùng để xử lý dữ liệu đầu vào, trong khi bộ giải mã sử dụng để tạo ra dự đoán. Mỗi bộ mã hóa gồm nhiều lớp con.



Hình 11: Kiến trúc mô hình Transformer

1.4.1.1. Encoder layer:

- Input Embedding:

Trong lĩnh vực học sâu, việc biểu diễn dữ liệu đầu vào dưới dạng số thực hoặc phức, cũng như sử dụng các cấu trúc toán học như vector và ma trận là rất quan trọng. Do đó, với sự phát triển của deep learning, phương pháp học biểu diễn đã thu hút sự quan tâm đáng kể. Mới đây, đã xuất hiện một số mô hình học biểu diễn từ đáng chú ý như GloVe, Fasttext, và gensim Word2Vec.

Input Embedding



Hình 12: Biểu diễn nhúng từ

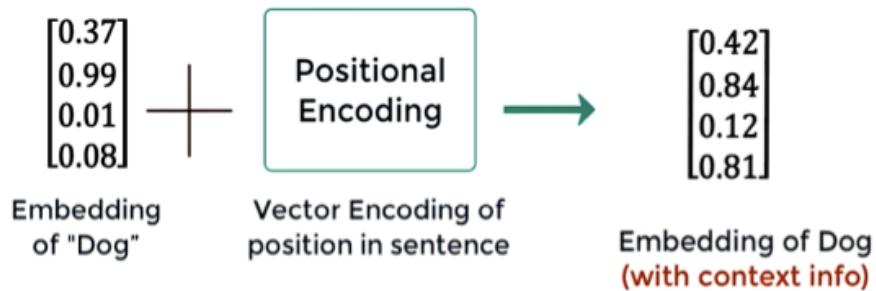
- Positional Encoding

Word embeddings giúp biểu diễn ý nghĩa của từ, nhưng ý nghĩa của cùng một từ có thể thay đổi khi nó xuất hiện ở các vị trí khác nhau trong một câu. Đó là lý do tại sao Transformers bổ sung Positional Encoding để cung cấp thông tin về vị trí của các từ trong câu.

$$PE_{(pos, 2i)} = \sin\left(\text{pos}/1000^{2i/d_{\text{model}}}\right)$$

$$PE_{(pos, 2i+1)} = \cos\left(\text{pos}/10000^{2i/d_{\text{model}}}\right)$$

Trong trường hợp này, "pos" thể hiện vị trí của từ trong câu, "PE" biểu thị giá trị của phần tử thứ i trong embeddings có độ dài d_{model} . Sau đó, chúng ta thực hiện phép cộng giữa vector PE và vector Embedding.



$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

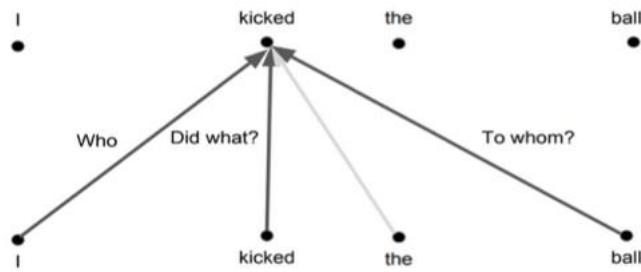
$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

Hình 13: Mã hóa vị trí nhúng

- Self-Attention

Self-Attention, trong Transformers, là một cơ chế quan trọng giúp mô hình "nhận biết" mối quan hệ giữa các từ trong một câu. Ví dụ, xét từ "kicked" trong câu "I kicked the ball" (Tôi đã đá quả bóng). Từ "kicked" có mối quan hệ mật thiết với "I" (chủ ngữ) và "ball" (vị ngữ), do đó, trong cơ chế Self-Attention, sự quan tâm đến những từ này sẽ "liên quan mạnh". Trong khi đó, từ "the," là một giới từ, thường không tạo ra sự kết nối mạnh mẽ với từ "kicked" trong bộ não của mô hình..

Self-Attention



Hình 14: Cơ chế Self-Attention

Cấu trúc tổng quan của mô-đun Multi-head Attention bao gồm ba vector đầu vào, được ký hiệu là Q (Querys), K (Keys) và V (Values). Từ ba vector này, vector attention Z cho một từ được tính toán bằng cách sử dụng công thức dưới đây:

$$Z = \text{softmax} \left(\frac{Q \cdot K^T}{\sqrt{\text{Dimension of vector } Q, K \text{ or } V}} \right) \cdot V$$

Thực hiện tính như sau:

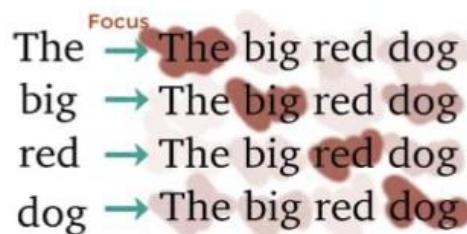
- Bước 1: Tính ba vector Q, K, V, input embeddings được nhân với ba ma trận trọng số tương ứng WQ, WK, WV.
- Bước 2: Vector K đóng vai trò như một "khóa" đại diện cho từng từ, trong khi vector Q sẽ truy vấn các vector K của các từ khác trong câu bằng phép nhân chập, để tính toán độ tương quan giữa chúng (điểm số "Score"). Bước này có mục tiêu chuẩn hóa các giá trị "Score" bằng cách chia cho căn bậc hai của số chiều của vector Q/K/V (trong ví dụ này, chia cho 8)

vì kích thước của Q/K/V là 64-D), để đảm bảo rằng giá trị "Score" không phụ thuộc vào độ dài của các vectơ Q/K/V.

- Bước 3: Áp dụng hàm Softmax để tạo ra một phân bố xác suất trên các từ, dựa trên các giá trị "Score" thu được từ bước trước.
- Bước 4: Nhân phân bố xác suất này với vectơ V để tạo ra một vectơ attention, loại bỏ các từ không quan trọng (có xác suất thấp) và giữ lại các từ quan trọng (có xác suất cao).
- Bước 5: Cuối cùng, cộng các vectơ attention (đã được nhân với đầu ra của hàm Softmax) lại với nhau để tạo ra ma trận attention cho toàn bộ câu. Quá trình này được lặp lại cho tất cả các từ trong câu để tạo ra ma trận attention hoàn chỉnh.

- Multi-head Attention

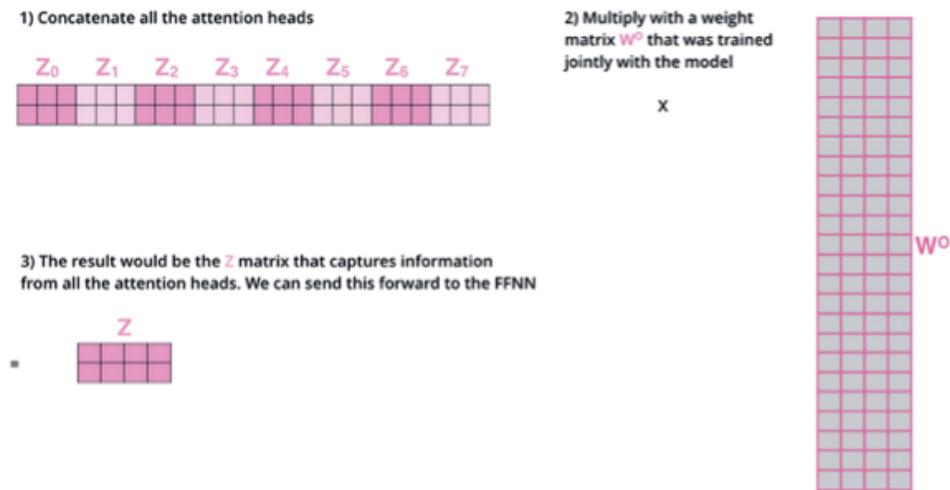
Vấn đề của Self-attention là attention của một từ sẽ luôn "chú ý" vào chính nó vì "nó" phải liên quan đến "nó" nhiều nhất. Ví dụ như sau



Hình 15: Multi-head Attention cho câu

Sự tương tác giữa các từ KHÁC NHAU trong câu được thực hiện bởi Multi-head attention: thay vì sử dụng 1 Self-attention (1 head) bằng cách sử dụng nhiều Attention khác nhau (multi-head), mỗi Attention sẽ chú ý đến một phần khác nhau trong câu.

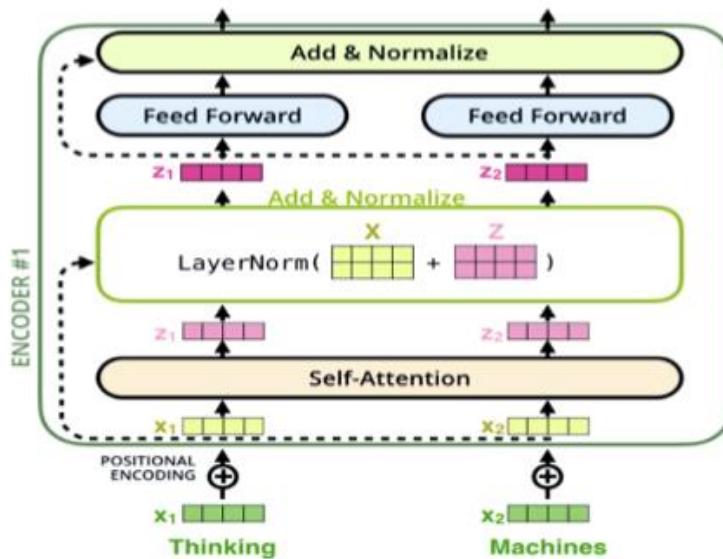
Mỗi "head" cho ra một ma trận attention riêng. Việc concat các ma trận này và nhân với ma trận trọng số WO sinh ra một ma trận attention duy nhất (weighted sum). Ma trận trọng số này được tune trong khi training.



Hình 16: Quá trình concat các Attention heads

- Residuals

Trong cấu trúc tổng quát (hình 11), mỗi sub-layer đều được thiết kế dưới dạng một khối dư thừa (residual block). Các kết nối bỏ qua (skip connections) trong kiến trúc Transformers cho phép thông tin chuyển qua sub-layer một cách trực tiếp. Cụ thể, thông tin này (x) được thêm vào attention (z) của nó và sau đó tiến hành quá trình Layer Normalization.



Hình 17: Quá trình cộng với attention (z) và thực hiện Layer Normalization

- Feed Forward

Khi đã thực hiện việc chuẩn hóa (Normalize) các vectơ z , chúng sẽ được truyền qua một mạng fully connected trước khi đưa vào bộ giải mã (Decoder). Bởi vì sự độc lập giữa các vectơ này, việc tính toán có thể được tận dụng một cách song song cho toàn bộ câu.



Hình 18: Tính toán song song cho câu

1.4.1.2. Decoder layer

- Masked Multi-head Attention

Trong quá trình thực hiện bài toán dịch từ tiếng Anh sang tiếng Pháp sử dụng mô hình Transformers, nhiệm vụ của Decoder là giải mã thông tin từ Encoder và tạo ra các từ tiếng Pháp dựa trên thông tin từ các từ trước đó. Nếu ta áp dụng Multi-head attention cho toàn bộ câu như ở phần Encoder, thì Decoder sẽ "nhìn thấy" toàn bộ phần còn lại của câu tiếng Pháp, bao gồm cả các từ mà nó sẽ dịch tiếp theo. Để ngăn chặn điều này, khi Decoder dịch đến từ thứ i , phần phía sau của câu tiếng Pháp sẽ được che (masked), và Decoder chỉ có thể "nhìn" thấy những phần mà nó đã dịch trước đó.



Hình 19: Tính toán song song cho câu

- Quá trình decode

Quá trình giải mã (Decode) cơ bản tương tự quá trình mã hóa (Encode) với sự khác biệt là Decoder thực hiện giải mã từng từ một và sử dụng dữ liệu đầu vào của mình (câu tiếng Pháp) sau khi đã thực hiện mask. Trong quá trình này, dữ liệu đã được mask được đưa qua sub-layer #1 của Decoder và nhân với một ma trận trọng số WQ. Ma trận K và V được lấy từ Encoder, cùng với ma trận Q từ quá trình Attention đa đầu (Masked Multi-Head Attention) để tiếp tục đưa vào sub-layer #2 và #3, tương tự như quá trình mã hóa. Cuối cùng, các vector thu được sau các sub-layer này được đưa qua một lớp Linear (mạng Fully Connected) và sau đó qua lớp Softmax để tính toán xác suất của từ tiếp theo trong chuỗi đầu ra.

1.4.2. Giới thiệu về BERT:

BERT, là từ viết tắt của Bidirectional Encoder Representations from Transformers có nghĩa là “**Mô hình Mã hoá Hai chiều dữ liệu từ các khối Transformers**” được giới thiệu bởi Google AI vào năm 2018. Đây là một mô hình ngôn ngữ tự học sâu (**Deep Learning**) được xây dựng dựa trên mô hình mạng mô phỏng theo hệ thống nơ-ron thần kinh của con người (neural network) được dùng để đào tạo trước (pre-train) quá trình xử lý ngôn ngữ tự nhiên. Hay nói cách khác, BERT được thiết kế để giúp cho Google có thể phân biệt được rõ chính xác các ý nghĩa, sắc thái, ngữ cảnh của các từ xuất hiện trong truy vấn tìm kiếm.

BERT sử dụng kiến thức từ mô hình Transformer. Transformer bao gồm hai phần quan trọng: Encoder và Decoder, nhưng BERT **chỉ sử dụng phần Encoder**. Mô hình BERT là một mô hình mã hóa sử dụng nhiều lớp Transformer hai chiều để hiểu và biểu diễn ngôn ngữ.



Hình 20: BERT là một trong những thuật toán quan trọng trong việc tìm kiếm.

Ví dụ: nếu chúng ta tìm kiếm từ “bank” tại Google, nếu không có ngữ cảnh, từ ghép xung quanh từ “bank” thì kết quả có thể trả về “bank account” (tài khoản ngân hàng) và “bank of the river” (bờ sông). Các mô hình ngữ cảnh của Google thường là mô hình ngữ cảnh đơn chiều, tức là khi nhập vô “i accessed the bank account”, mô hình sẽ diễn giải từ “bank” dựa trên từ xuất hiện trước nó là “i accessed the” mà không tính tới từ “account” ở phía sau. Vậy nên BERT được tạo ra để diễn giải cả hai chiều dữ liệu để giúp cho hệ thống hiểu đúng với từ thông qua ngữ cảnh “trước” và “sau” của từ đó.

1.4.3. Giới thiệu về RoBERTa:

Sự ra đời của BERT đã mang lại những cải tiến đáng kể cho những bài toán Question Answering, Sentiment Analysis,... Tuy nhiên, **mô hình BERT lại không mang lại kết quả thực hiện tốt đối với ngôn ngữ ít phổ biến**. Điều này dẫn đến RoBERTa ra đời, đây là một dự án của Facebook, nó kế thừa các kiến trúc và thuật toán từ mô hình BERT và được triển khai trên framework PyTorch, một framework được phát triển bởi Facebook và được yêu thích bởi cộng đồng AI. Dự án này hỗ trợ việc huấn luyện lại các mô hình BERT trên các bộ dữ liệu mới cho các ngôn ngữ khác ngoài các ngôn ngữ phổ biến. Kể từ khi được ra mắt, RoBERTa đã đóng góp một loạt các mô hình pretrain cho nhiều ngôn ngữ khác nhau.

1.4.4. Giới thiệu về phoBERT:

Mô hình ngôn ngữ đã được tiền huấn luyện trước cho tiếng Việt, gọi là PhoBERT, đang được coi là những mô hình ngôn ngữ hàng đầu cho tiếng Việt (**Pho**, tức là "Phở", là một món ăn phổ biến tại Việt Nam + với tên gốc BERT):

+ Có hai phiên bản PhoBERT là "**base**" và "**large**" được công khai, đây là những mô hình ngôn ngữ tiền huấn luyện trước lớn đầu tiên cho tiếng Việt. Phương pháp tiền huấn luyện của PhoBERT dựa trên RoBERTa, tối ưu hóa quy trình tiền huấn luyện BERT để đạt hiệu suất mạnh mẽ hơn.

Bảng 1: Các mô hình PhoBERT đã được đào tạo trước

| Model | #params | Arch. | Max-length | Pre-training data | License |
|-----------------------|---------|-------|------------|---|------------------------------------|
| vinai/phobert-base | 135M | base | 256 | 20GB dữ liệu từ Wikipedia và các văn bản tin tức | MIT License |
| vinai/phobert-large | 370M | Large | 256 | 20GB dữ liệu từ Wikipedia và các văn bản tin tức | MIT License |
| vinai/phobert-base-v2 | 135M | base | 256 | 20GB văn bản Wikipedia và Tin tức + 120GB văn bản từ OSCAR-2301 | GNU Afferro GPL v3 |

+ PhoBERT vượt trội hơn so với các phương pháp tiền huấn luyện trước đối với tiếng Việt và nhiều ngôn ngữ khác, đạt được hiệu suất tiên tiến mới trên bốn nhiệm vụ xử lý ngôn ngữ tự nhiên bao gồm gán loại từ, phân tích phụ thuộc, nhận dạng thực thể định danh và suy luận ngôn ngữ tự nhiên.

CHƯƠNG 2: NỘI DUNG THỰC HIỆN

Định luật 80-20, hay còn gọi là Nguyên tắc Pareto, là một trong những nguyên tắc quan trọng trong cuộc sống và kinh doanh. Nguyên tắc này cho rằng "80% kết quả thường đến từ 20% nguyên nhân". Áp dụng lý thuyết này vào việc cải thiện chất lượng dịch vụ của HighLand trên nền tảng Google Maps, chúng ta cần **tập trung vào việc xử lý những vấn đề của 20% khách hàng đang không hài lòng** để làm cho trải nghiệm chung của tất cả khách hàng trở nên tốt hơn. Điều này đòi hỏi ta phải lắng nghe "*tiếng nói của khách hàng*" (Voice of Customer) thông qua việc thu thập phản hồi và nhận xét từ họ để xác định những vấn đề cụ thể mà họ đang gặp phải. Việc này sẽ khiến Highland có thể duy trì và củng cố danh tiếng thương hiệu của doanh nghiệp. Ngoài ra, ta cũng sẽ **đồng thời thu thập ý kiến từ 80% khách hàng còn lại để tối ưu hóa dịch vụ và sản phẩm**, đảm bảo rằng Highland có thể duy trì sự hài lòng và trung thực của đối tượng lớn này.

2.1. Thu thập dữ liệu:

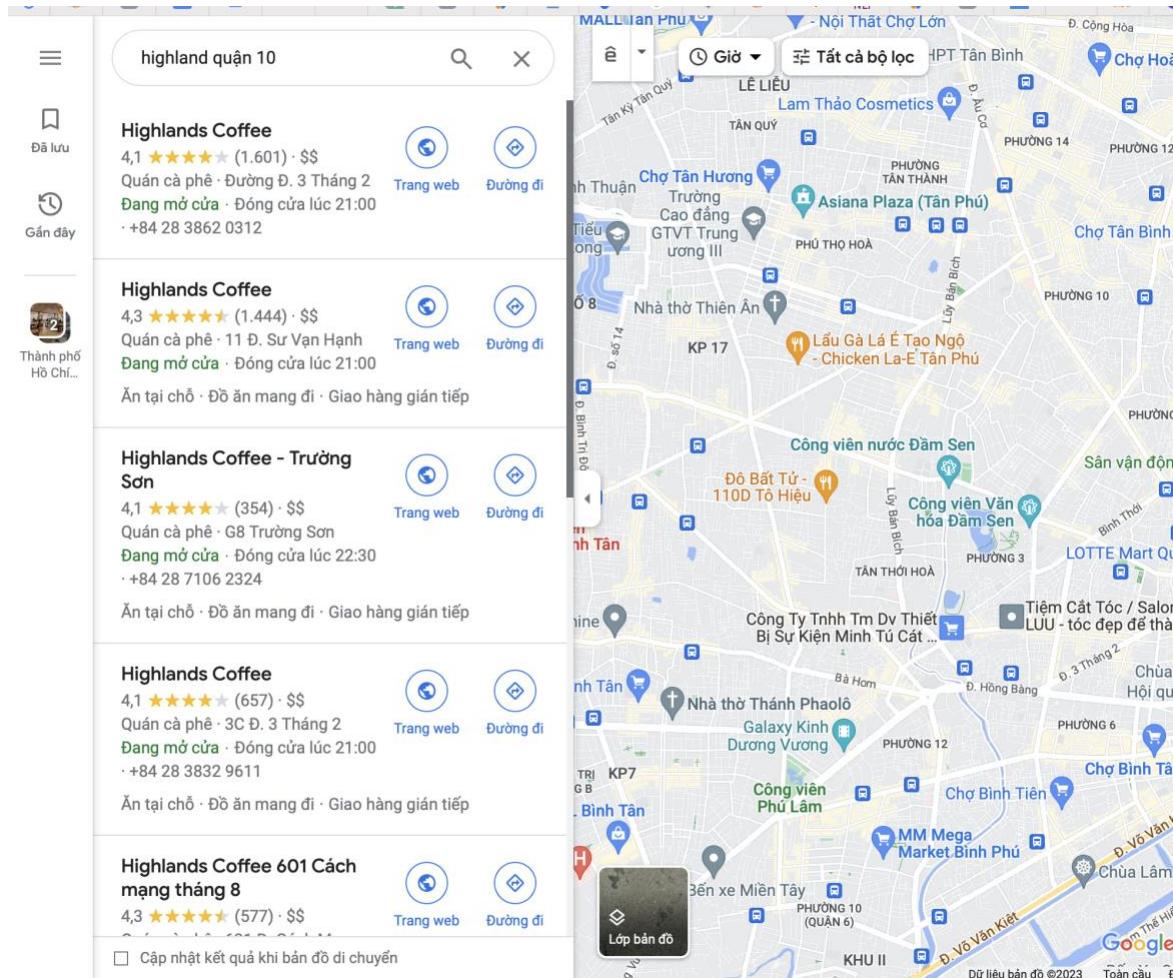
2.1.1. Cách thu thập dữ liệu:

Để có thể thu thập được ý kiến, quan điểm, cảm nhận của khách hàng về chất lượng dịch vụ HighLand trên Google maps ta cần phải sử dụng thư viện: **BeautifulSoup** và **công cụ Selenium**. Đây là 2 công cụ phổ biến trong ngôn ngữ python dùng để hỗ trợ lẫn nhau trong việc lấy dữ liệu từ các trang web.

Đối với công cụ Selenium, ta sẽ **chọn Selenium WebDriver với bộ trình duyệt Google Chrome** để phục vụ bài toán vì là Selenium có công cụ mở rộng được sử dụng đặc biệt với trình duyệt Chrome là ChromeDriver, đây là một giao diện để kết nối và điều khiển trình duyệt Chrome từ Selenium. Nó cung cấp tích hợp tốt với trình duyệt này và hỗ trợ nhiều tính năng mạnh mẽ. Vì **Selenium WebDriver có các tín năng tự động hóa**, nên ta sẽ sử dụng tín năng này để thực hiện thao tác chọn các tín năng web, cuộc review ở google map và phần danh sách các quán HighLand.

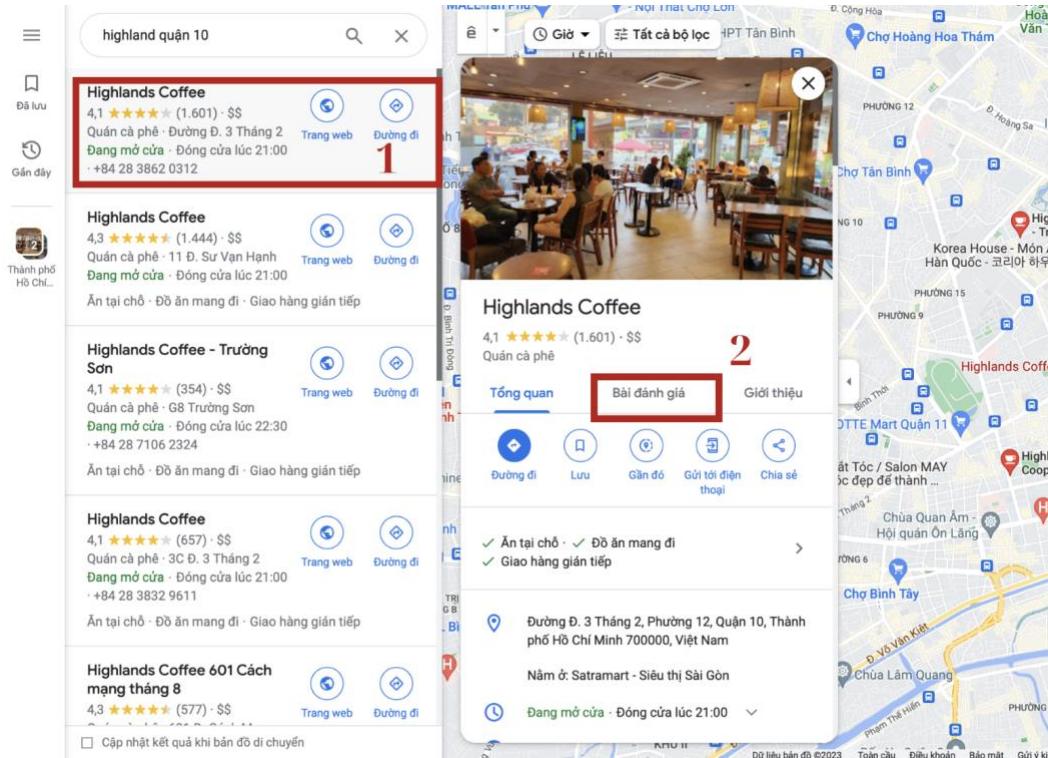
Những dữ liệu ta muốn lấy là những thông tin của khách hàng, bao gồm: **Tên khách hàng, số sao khách hàng đánh giá, thời gian khách hàng đánh giá và bình luận của khách hàng**. Để lấy những dữ liệu này, đầu tiên ta cần cài đặt Selenium WebDriver tương ứng với trình duyệt muốn sử dụng (ở đây sẽ cài **ChromeDriver**), lúc này Selenium cho

phép ta tương tác với trình duyệt Google Chrome thông qua mã code của mình. Sau đó ta sẽ thông qua công cụ này thực hiện các hành động điều hướng mở trang web Google maps của HighLand quận 10 đã được cho trước thông qua câu lệnh **driver.get()**.



Hình 21: Kết quả Selenium thành công điều hướng tới web

Tiếp đến, ta sẽ sử dụng câu lệnh **driver.find_element()** để xác định vị trí quán đầu tiên sau đó thực hiện hành động click vào quán đầu tiên (1), sau đó tương tự ta sẽ tìm vị trí click mục “Bài đánh giá” (2) để đến với mục đánh giá của khách hàng như **hình 22**.

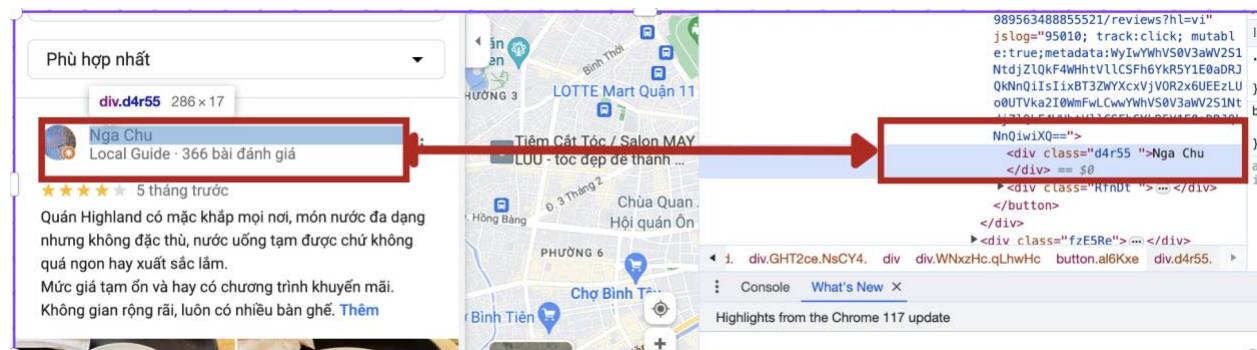


Hình 22: Sử dụng Selenium click vào khu vực dữ liệu cần lấy

Sau khi đến với mục “Bài đánh giá” (3) như trong **hình 27**, dựa vào thư viện BeautifulSoup, ta sẽ lấy được trong phần thông tin khách hàng (4).

Để có thể lấy dữ liệu thì ta phải tiến hành truy xuất dữ liệu tương ứng theo thẻ HTML chứa các thông tin đó.

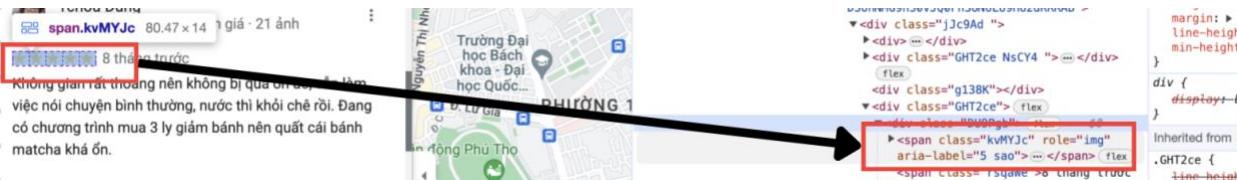
+ Để lấy tên của khách hàng, ta sẽ dò tới một div có class là "d4r55" để lấy tên khách hàng hoặc dò đến “aria-label” để lấy tên.



Hình 23: Lấy dữ liệu tên khách hàng bằng thư viện BeautifulSoup dựa trên div class

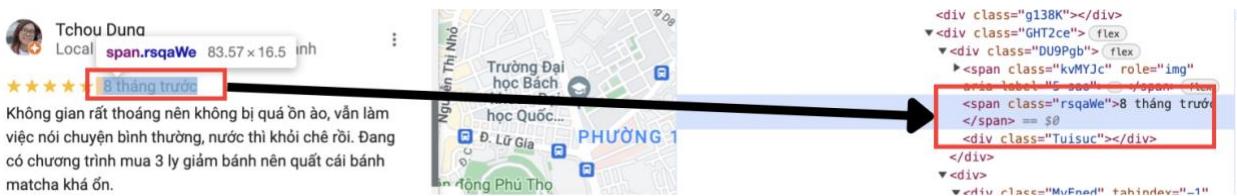
Hình 24: Lấy tên khách hàng dựa vào aria-label

+ Để lấy rating khách hàng, ta dò đến class “kvMYJc”



Hình 25: Lấy rating khách hàng

+ Để lấy thời gian khách hàng đã bình luận, ta dò đến class “rsqaWe”



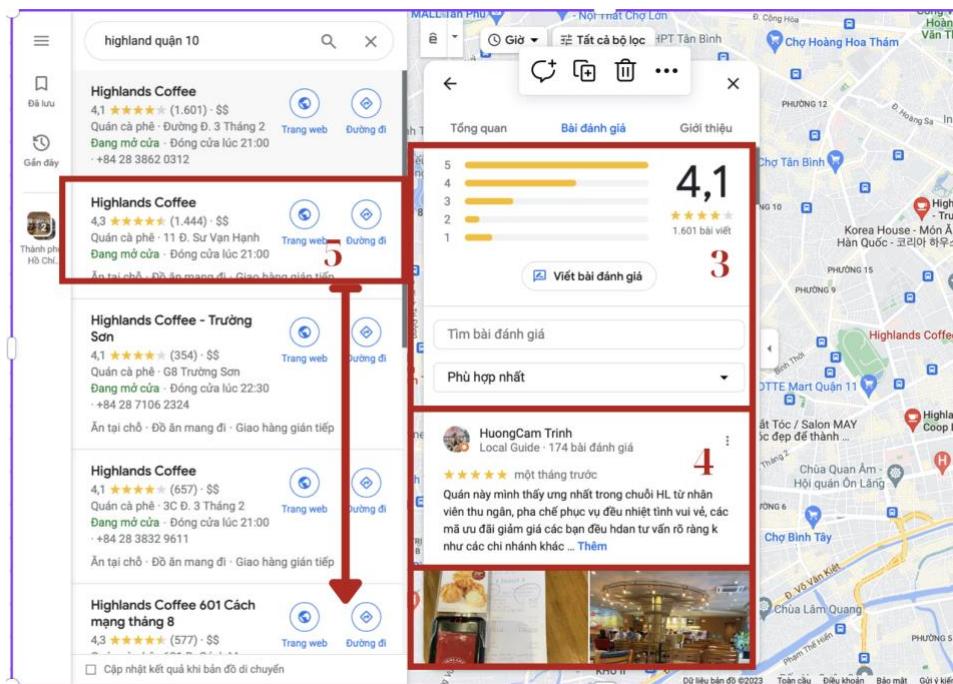
+ Để lấy bình luận khách hàng, ta dò tới class “wiI7pd”



Hình 26: Lấy bình luận khách hàng

Tuy nhiên để lấy được nhiều đánh giá của khách hàng nhất có thể, ta phải dùng thêm selenium với chức năng “cuộn thanh” đánh giá, nếu không tối đa mục “bài đánh giá” ta có thể lấy được **chỉ có 10 bình luận**. (Vì google maps hiện thị mặc định là 10 bình luận nếu ta không lướt xuống để tải thêm bình luận)

Tương tự với các quán Highland khác (5), ta sẽ áp dụng thêm vòng lặp lấy thông tin như vậy đối với các quán còn lại, đồng thời cũng dùng Selenium với chức năng cuộn để có thể cuộn được hết quán cà phê quận 10.



Hình 27: Hướng lấy dữ liệu từ phần đánh giá của tất cả quán Highland quận 10

2.1.2. Dữ liệu thu thập được:

Sau khi thu thập xong, ta sẽ có được bộ dữ liệu là 3739 dòng với các thuộc tính sau đây:

| | Reviewer Name | Review Rate | Review Time | Review Comment |
|---|------------------|-------------|-----------------|--|
| 0 | HuongCam Trinh | 5 sao | một tháng trước | Quán này mình thấy ưng nhất trong chuỗi HL từ ... |
| 1 | Chin Chin | 5 sao | 4 tháng trước | Không gian thoáng mát nhiều chỗ ngồi. Mình ngồi... |
| 2 | Tchou Dung | 5 sao | 6 tháng trước | Không gian rất thoáng nên không bị quá ồn ào, ... |
| 3 | nguyen trinh | 4 sao | 2 tháng trước | Quán gần nhà, không gian thoáng view nhìn ra đ... |
| 4 | Anh Nguyễn Hoàng | 5 sao | 2 tháng trước | Ở trong khu satsa, có bãi xe, kê bóng cây lớn ... |

Hình 28: Dataset khi thu thập được

Bảng 2: Thuộc tính và mô tả thuộc tính của bảng dữ liệu thu thập được

| Thuộc tính | Mô tả thuộc tính |
|----------------|--|
| Reviewer Name | Tên khách hàng |
| Review Rate | Điểm đánh giá của khách hàng |
| Review Time | Khoảng thời gian khách hàng đánh giá đến bây giờ |
| Review Comment | Lời đánh giá từ khách hàng |

2.2. Tiền xử lý:

Trước hết, ta nên xoá bỏ cột “**Reviewer Name**”, vì cột này không có giá trị thông tin và không đóng góp cho mục tiêu phân loại của chúng ta. Vì vậy, chúng ta còn 3 cột dữ liệu như sau: “**Review Rate**” và “**Review Time**” sẽ được tiền xử lý để hỗ trợ việc tìm hiểu về dữ liệu - tổng quan tình hình khách hàng Highland đánh giá chất lượng dịch vụ thông qua số sao. Còn “**Review Comment**” sẽ được sử dụng để phân tích ý kiến (Sentiment).

| Review Rate | Review Time | Review Comment |
|-------------|-------------|---|
| 0 | 5 sao | một tháng trước Quán này mình thấy ưng nhất trong chuỗi HL từ ... |
| 1 | 5 sao | 4 tháng trước Không gian thoáng mát nhiều chỗ ngồi. Mình ngồi... |
| 2 | 5 sao | 6 tháng trước Không gian rất thoáng nên không bị quá ồn ào, ... |
| 3 | 4 sao | 2 tháng trước Quán gần nhà, không gian thoáng view nhìn ra đ... |
| 4 | 5 sao | 2 tháng trước Ở trong khu satra, có bãi xe, kê bóng cây lớn ... |
| 5 | 1 sao | 6 tháng trước Trải nghiệm tệ hại chưa từng có. Tôi vào order... |
| 6 | 5 sao | một tháng trước 😊 Về không gian quán\n\nĐến Highlands review th... |
| 7 | 1 sao | một tháng trước Lè mè không lo làm món, toàn đứng chơi nạnh nhau. |
| 8 | 1 sao | 4 tháng trước Bảo vệ ca chiều tối bãi gửi xe kém. Bắt khách ... |
| 9 | 5 sao | một năm trước Quán nằm gần siêu thị Satramart.\nMỗi lần đi s... |

Hình 29: Dữ liệu còn lại khi drop cột "Reviewer Name"

2.2.1. Xử lý Review Time:

Đầu tiên, ta thấy được rằng dữ liệu cột “**Review Time**” khi thu thập được đang không theo quy luật thời gian:

```
array(['1 tháng trước', '4 tháng trước', '6 tháng trước', '2 tháng trước',
       '1 năm trước', '8 tháng trước', '3 năm trước', '5 năm trước',
       '5 tháng trước', '4 năm trước', '2 năm trước', '9 tháng trước',
       '6 năm trước', '11 tháng trước', '10 tháng trước', '7 tháng trước',
       '2 tuần trước', '6 ngày trước', '1 tuần trước', '3 tháng trước',
       '4 ngày trước', '3 tuần trước', '4 tuần trước', '5 ngày trước',
       '1 ngày trước', '3 giờ trước', '7 năm trước', '3 ngày trước'],
      dtype=object)
```

Hình 30: Thông tin hiển thị cột giá trị thời gian

Ta có thể thấy được rằng, **Google maps ghi nhận khoảng thời gian kể từ thời điểm khách hàng đánh giá**, chứ không ghi nhận thời điểm ngày tháng khách hàng đánh giá. Vậy nên ta sẽ chia dữ liệu thành 2 phần như sau:

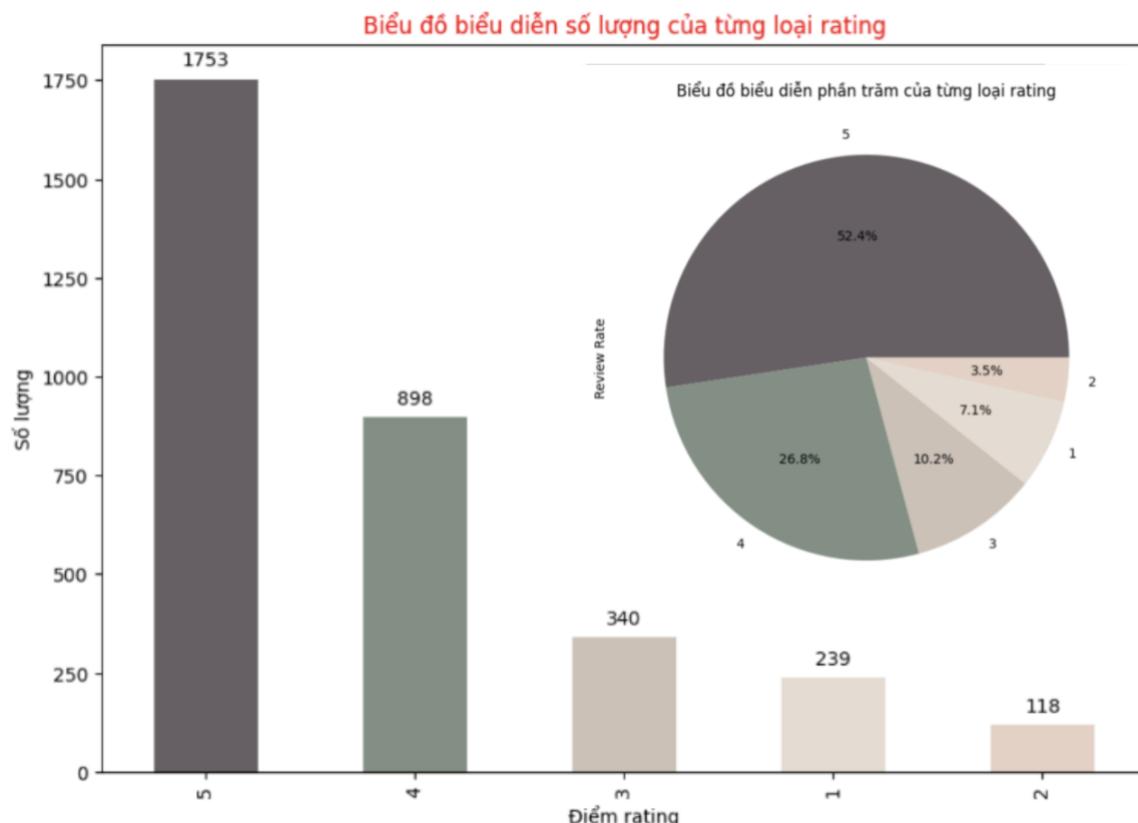
+ Vì thu thập dữ liệu từ đầu tháng 10, nên dữ liệu từ tháng 11/2022 đến 10/2023 sẽ tính là dữ liệu trong 1 năm trở lại đây. (2022-2023) (Đây là những dữ liệu có cột giá trị **không dính tới năm**, vì nếu tính theo năm thì tức đã trên 1 năm kể từ lần đánh giá đó)

+ Tương tự với những năm khác, ta đẩy lùi giá trị với những dữ liệu có năm. (Tức là dữ liệu hiện “1 năm trước” ta sẽ chuyển về 2021-2022).

+ Ngoài ra, ta sẽ **loại bỏ tất cả dữ liệu trên 5 năm** (Vì dữ liệu bình luận thay đổi theo thời gian do sự phát triển và sự thay đổi trong quan điểm của người dùng. Nếu thu thập dữ liệu quá lâu, nó có thể không còn đại diện cho tình hình hiện tại.)

2.2.2. Xử lý Review Rate:

Loại bỏ chữ “sao” ở cuối mỗi dòng trong cột để cột có giá trị dạng số. Điều này giúp cho cột mang giá trị dễ biểu diễn trực quan hơn việc tìm hiểu dữ liệu.

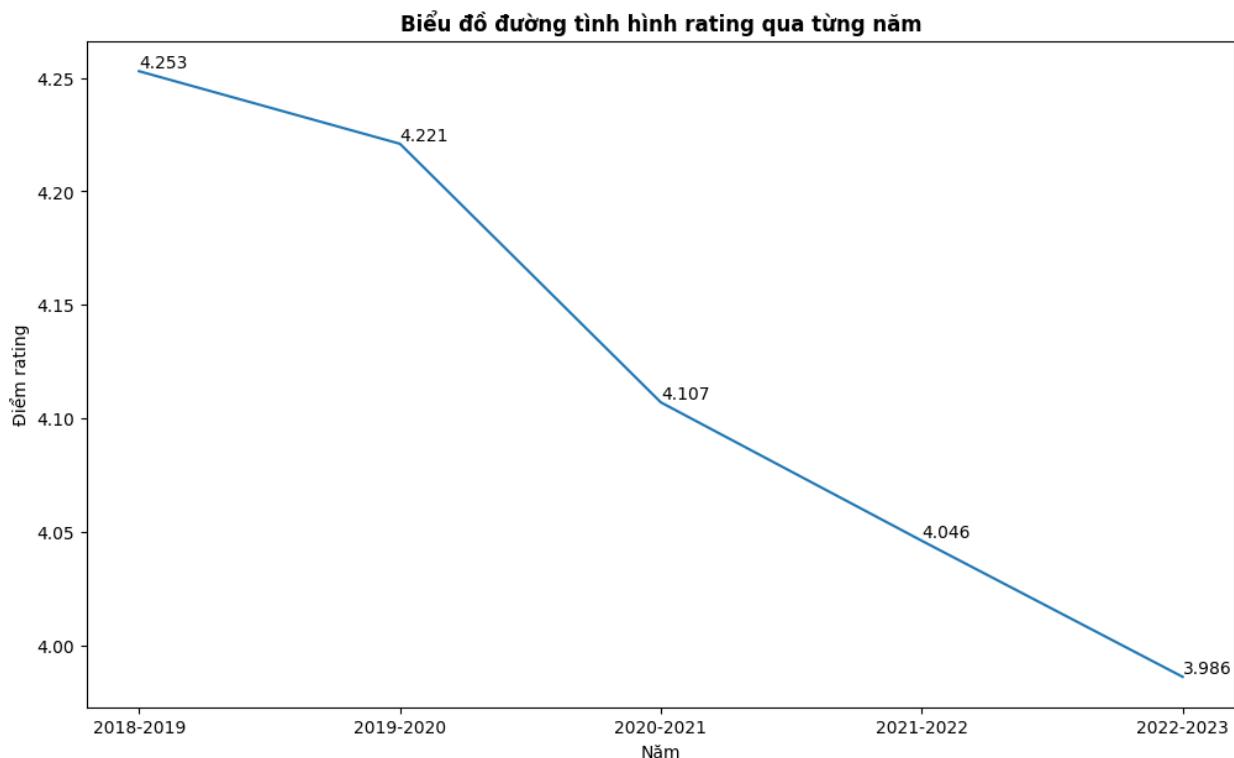


Hình 31: Tổng quan Rating trong 5 năm gần đây của Highland

Điểm rating trung bình của tất cả quán cà phê trong 5 năm trở lại đây : 4.137

Nhìn chung, thông qua biểu đồ **hình 31** và **điểm rating trung bình**, ta có thể thấy được rằng khách hàng đánh giá chất lượng dịch vụ tương đối cao, phần lớn được đánh giá ở mức 5 sao và 4 sao với tỉ lệ là (**52,4%** và **26,8%**). Điều này chắc chắn là một dấu hiệu tích cực về sự hài lòng của khách hàng với sản phẩm và dịch vụ của Highland.

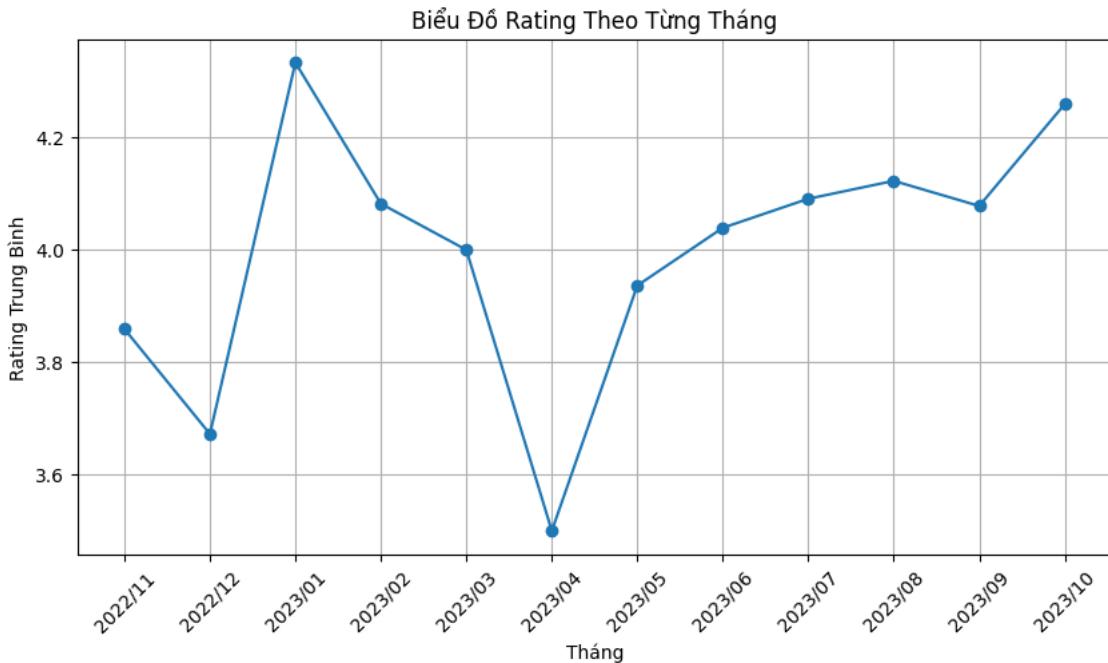
Tuy nhiên, khi xét theo từng năm, kể từ năm 2018 đến nay, ta có thể thấy điểm rating trung bình đang **ngày càng có xu hướng tụt dốc** từ 4.253 (2018-2019) xuống 3.986 (2022-2023). Điều này có thể là một tín hiệu đáng quan ngại về sự thay đổi trong cách khách hàng đánh giá chất lượng dịch vụ của chúng tôi.



Hình 32: Biểu đồ tình hình rating trung bình qua từng năm

Điều đáng quan tâm hơn, nếu ta xem xét những thay đổi cách đánh giá của khách hàng trong khoảng thời gian gần đây (2022-2023), thì có một số tháng thể hiện mức đánh giá thấp hơn mức trung bình, với thậm chí có tháng nằm dưới 3.5 sao, như trong trường hợp của tháng 4/2023. Điều này có thể là dấu hiệu cho thấy có vấn đề cụ thể xảy ra trong

khoảng thời gian đó và ta cần phải nghiên cứu kỹ lưỡng để xác định nguyên nhân cụ thể và tiến hành các biện pháp sửa đổi để nâng cao chất lượng dịch vụ.



Hình 33: Biểu đồ rating trung bình theo từng tháng từ 11/2022 đến 10/2023

2.2.3. Xử lý Review Comment:

Vậy nên trước khi đi vào phân tích dữ liệu ta sẽ tiến hành một số bước làm sạch dữ liệu, tách từ thông qua 2 thư viện:

+ **Gensim** (gensim.utils.simple_preprocess) để thực hiện tiền xử lý văn bản trên mỗi dòng trong cột '**Review Comment**'. Cụ thể, nó thực hiện việc chuyển đổi dữ liệu ở cột thành các từ viết thường và loại bỏ dấu câu, ký tự đặc biệt.

+ **VnCoreNLP**: Sau đó ta khởi động bộ công cụ phân đoạn, tách từ (word segmentation) với đoạn mã dưới đây:

```
rdrsegmenter = VnCoreNLP("/content/drive/MyDrive/VnCoreNLP/VnCoreNLP-1.1.1.jar", annotators="wseg", max_heap_size='-Xmx500m')
```

Trong đó:

+ Đối tượng VnCoreNLP được khởi tạo với đường dẫn đến tệp JAR của VnCoreNLP ("*/content/drive/MyDrive/VnCoreNLP/VnCoreNLP-1.1.1.jar*").

+ annotators="wseg" chỉ định rằng muốn sử dụng VnCoreNLP cho phân đoạn từ (word segmentation).

+ max_heap_size=''-Xmx500m' là đối số để cấu hình dung lượng heap tối đa sẽ sử dụng cho VnCoreNLP. Trong trường hợp này, kích thước heap được thiết lập là 500MB. (Kích thước heap 500MB có thể được xem xét là một giá trị cân nhắc giữa đảm bảo hiệu suất và không tạo áp lực quá lớn lên tài nguyên hệ thống.)

Tiếp theo trong quá trình tiền xử lý dữ liệu văn bản tiếng Việt, chúng ta cần loại bỏ các từ dừng, những từ như "bị," "bởi," "cả," "các," và "cái," vì chúng thường không mang nhiều thông tin hữu ích và thường xuất hiện thường xuyên trong văn bản. Loại bỏ các từ dừng này giúp giảm kích thước của văn bản và cải thiện hiệu suất xử lý ngôn ngữ tự nhiên. Hiện tại, chúng ta có sẵn hai tệp dữ liệu stopword tiếng Việt từ GitHub ([GitHub - stopwords/vietnamese-stopwords: Vietnamese stopwords](#)). Chúng ta sẽ sử dụng danh sách từ dừng từ các tệp này để tiến hành loại bỏ chúng khỏi cột "**Review comment**" của dữ liệu.

| Review Comment | tokens |
|--|--|
| Quán này mình thấy ưng nhất trong chuỗi HL từ ... | [quán, ưng, chuỗi, hl, nhân_viện, thu_ngân, ph... |
| Không gian thoáng mát nhiều chỗ ngồi. Mình ngồi... | [không_gian, thoáng, mát, chỗ, hơi, muỗi, chíc... |
| Không gian rất thoáng nên không bị quá ồn ào, ... | [không_gian, thoáng, ồn_ào, làm_việc, nói_chuy... |
| Quán gần nhà, không gian thoáng view nhìn ra đ... | [quán, không_gian, thoáng, view, đường, thoải_... |
| Ở trong khu satra, có bãi xe, kê bóng cây lớn ... | [khu, satra, bãi, xe, kê, bóng, mát_mè, phòng,... |
| ... | ... |
| Đây là một trong những quán cà phê nhộn nhịp n... | [quán, cà_phê, nhộn_nhip, tháp, viettel, nhân_... |
| Cà phê ngon trong môi trường xung quanh thoảii ... | [cà_phê, ngon, môi_trường, xung_quanh, thoảii_mái] |
| Địa điểm đẹp, view đẹp, nhiều quán cà p ... | [địa_diểm, đẹp, view, đẹp, quán, cà] |
| Cùng một tiêu chuẩn cao sau 7 năm hoặc lâu h ... | [tiêu_chuẩn] |
| Quá đông đú ... | [đông, đú] |

Hình 34: Xoá bỏ stopword

2.3 Mô hình PhoBERT - SENTIMENT:

2.3.1. Lựa chọn mô hình:

Để có thể phân tích Sentiment cho bình luận khách hàng ta thu thập được từ Google maps thực chất là việc phân loại ý kiến (Sentiment) **dữ liệu văn bản không có nhãn**. Và việc để xây dựng mô hình sentiment cho dữ liệu tiếng việt không có nhãn là một vấn đề

phức tạp và có thể đòi hỏi sự tốn tài nguyên lớn và chưa hoàn hảo. Sau đây là một số phương pháp cũng như mô hình để hỗ trợ cho phân tích Sentiment:

Bảng 3: Đánh giá phương pháp - mô hình SENTIMENT

| PHƯƠNG PHÁP | ĐÁNH GIÁ |
|--|---|
| Word Embedding và Text Classification: Sử dụng các phương pháp nhúng từ (word embeddings) như Word2Vec hoặc FastText để biểu diễn tiếng Việt trong không gian vecto, sau đó sử dụng các mô hình phân loại văn bản như Logistic Regression, Support Vector Machines (SVM), Naive Bayes để thực hiện sentiment. | <p>→ Đây là cách tiếp cận truyền thống, đơn giản. Tuy nhiên dữ liệu tiếng Việt là một ngôn ngữ phức tạp, khi tiếp cận các từ thông qua biểu diễn vecto sẽ khiến chúng không nắm bắt được ngữ cảnh trong câu.</p> |
| Sử dụng Mạng Nơ-ron Học sâu (Deep Learning): Convolutional Neural Networks (CNN) hoặc Recurrent Neural Networks (RNN) để xây dựng mô hình phân tích cảm tính | <p>→ Tuy nhiên để xây dựng ta cần có một bộ dữ liệu rất lớn có nhãn để huấn luyện mô hình, sau đó áp dụng nó cho dữ liệu không có nhãn. Vậy nên đối với bộ dữ liệu từ đầu không có nhãn của ta thì ta không sử dụng cách này.</p> |
| Dựa trên từ vựng (Lexicon-base approach): Phương pháp này dựa vào từ điển chứa các từ hoặc cụm từ tích cực và tiêu cực. Có các thư viện từ điển cảm xúc tiếng việt như VnSentiment, VADER (Valence Aware Dictionary and sEntiment Reassoner), SentiVN, TextBlob, ... | <p>→ Phương pháp này hữu ích trong những trường hợp từ vựng đơn giản, điều này dẫn tới phương pháp không phân tích thông tin sâu về lý do hoặc ngữ cảnh mà cảm xúc được thể hiện.</p> |

| | |
|--|--|
| <p>Transfer Learning: Bắt đầu bằng việc sử dụng một mô hình tiền huấn luyện đã có sẵn cho sentiment analysis bằng tiếng Anh hoặc một ngôn ngữ khác, sau đó điều chỉnh lại cho ngôn ngữ tiếng Việt. Phương pháp này cho phép mô hình học cách áp dụng kiến thức từ một ngôn ngữ sang tiếng Việt, giúp tận dụng sự thông tin có sẵn và tiết kiệm thời gian huấn luyện.</p> | <p>→ Nếu chọn phương pháp này mô hình sẽ phù thuộc vào bộ dữ liệu huấn luyện: Bộ dữ liệu về cảm xúc mà bạn muốn phân tích nên thuộc cùng một lĩnh vực (ví dụ: nếu bạn sử dụng dữ liệu tiếng Anh về phim ảnh Mỹ để xây dựng mô hình, thì tệp kiểm tra không nên là dữ liệu tiếng Việt về phim tài liệu Việt Nam). Bộ dữ liệu chúng ta không có bộ huấn luyện khác cùng một lĩnh vực khác phù hợp.</p> |
| <p>Pre-trained Language Models: Sử dụng các mô hình đã được huấn luyện trước và có thể fine-tuning (tinh chỉnh mô hình) cho việc sentiment trên dữ liệu tiếng việt khác một cách tốt mà không yêu cầu dữ liệu đó có gắn nhãn cụ thể. Trong đó, Hugging Face Transformers đã cung cấp cho ta các mô hình BERT tiền huấn luyện sẵn như PhoBERT là mô hình đã được huấn luyện đặc biệt cho ngôn ngữ tiếng Việt , giúp nó hiểu rõ hơn ngữ cảnh và ngữ nghĩa của từ ngữ tiếng Việt.</p> | <p>➔ Chọn mô hình: PhoBERT do đây là mô hình ngôn ngữ được đào tạo trên một lượng lớn văn bản tiếng việt nên khả năng hiểu biết ngôn ngữ tiếng Việt của nó là rất cao, giúp nó có khả năng hiểu và phân tích cảm xúc một cách chính xác hơn so với các mô hình không được đào tạo trước.</p> |

2.3.2. Thực nghiệm mô hình:

- Ta tạo một **tokenizer** để xử lý và chuyển đổi văn bản thành chuỗi mã hoá, và một mô hình dựa trên PhoBERT để thực hiện phân loại cảm xúc trên các đoạn văn bản tiếng Việt sử dụng mô hình đã được đào tạo trước như đoạn mã dưới đây:

```
# Khởi tạo tokenizer và model
tokenizer = AutoTokenizer.from_pretrained("vinai/phobert-base-v2")
model = AutoModelForSequenceClassification.from_pretrained("wonrax/phobert-base-vietnamese-sentiment")
```

- Trong đó :

+**AutoTokenizer.from_pretrained("vinai/phobert-base-v2")**: Tokenizer được sử dụng để **chia văn bản thành các từ** (hoặc phần tử) và **chuyển chúng thành các mã số hóa** để đưa vào mô hình.

- AutoTokenizer là một lớp trong thư viện Transformers của Hugging Face, được sử dụng để tạo một tokenizer từ mô hình đã được đào tạo trước.
- “vinai/phobert-base-v2” là tên của mô hình mà muốn sử dụng (Đây là phiên bản nhẹ của mô hình PhoBERT. Nó có kích thước nhỏ hơn so với vinai/phobert-large và vinai/phobert-base, do đó hoạt động nhanh hơn, hiệu suất tốt và yêu cầu ít tài nguyên tích hợp hơn. Vậy nên vinai/phobert-base-v2 là sự lựa chọn tốt cho bộ dữ liệu ít của chúng ta – 1k8 dòng)

+**AutoModelForSequenceClassification.from_pretrained("wonrax/phobert-base-vietnamese-sentiment")**:

- “wonrax/phobert-base-vietnamese-sentiment” là tên của mô hình được sử dụng cho phân loại cảm tính cho văn bản tiếng Việt.
- AutoModelForSequenceClassification cho phép tự động chọn mô hình phù hợp với tác vụ phân loại chuỗi (sequence classification), trong bài toán này là phân loại cảm tính ý kiến.

```
# Dữ liệu không gắn nhãn
unlabeled_data = df_sen['Review Comment'].tolist()

# Tiền xử lý và tách từ với VnCoreNLP
unlabeled_data = [' '.join(rdrsegmenter.tokenize(u)) for u in unlabeled_data]

# Mã hóa dữ liệu
encoded_input = tokenizer(unlabeled_data, padding=True, truncation=True, max_length=128, return_tensors='pt')

# Đưa dữ liệu qua model
with torch.no_grad():
    outputs = model(**encoded_input)

# Lấy kết quả phân loại cảm xúc từ outputs
sentiments = torch.argmax(outputs.logits, dim=-1)

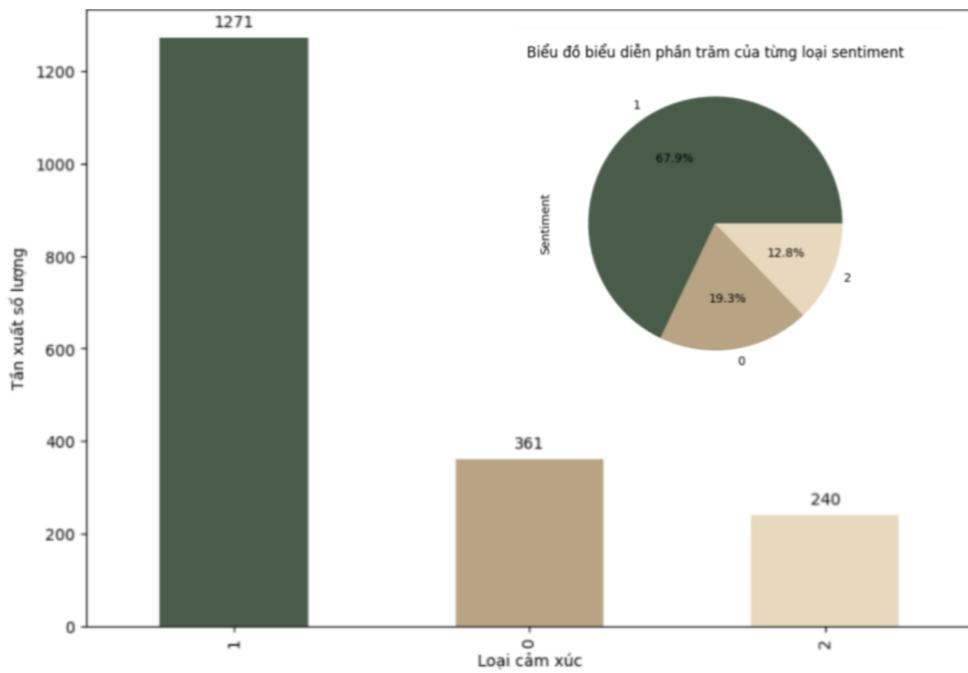
# `sentiments` bây giờ chứa kết quả phân loại cảm xúc cho dữ liệu không gắn nhãn
```

- Cột “**Review Comment**” là dữ liệu chúng ta cần phân loại ý kiến (Sentiment), đây là **dữ liệu không có nhãn** nên ta sẽ đặt tên cho nó là **unlabeled_data**, ta thực hiện việc tách từ cho các đoạn bình luận trong danh sách **unlabeled_data** bằng cách sử dụng VnCoreNLP thông qua **rdrsegmenter** (được tạo ở TXL) sau đó nối các từ lại thành một đoạn văn bản được phân đoạn.

- Sau đó, ta tiến hành mã hóa dữ liệu văn bản đã được tách từ thành định dạng **PyTorch tensors** (Tensors là một loại dữ liệu chính trong PyTorch và các thư viện khác sử dụng máy học sâu. Tensors là một phiên bản tổng quát của mảng đa chiều (nhiều chiều) mà mình có thể sử dụng để lưu trữ và xử lý dữ liệu số học, chẳng hạn như ma trận hoặc các mảng nhiều chiều khác.)

- Tiếp đến ta sẽ chuyển dữ liệu đã được mã hóa vào mô hình model để lấy kết quả đầu ra. Kết quả đầu ra này lớp cảm xúc dự đoán cho mỗi đoạn văn bản. Ta sử dụng hàm **torch.argmax** để tìm lớp có xác suất cao nhất trong các lớp đầu ra của mô hình (các lớp tượng trưng cho các cảm xúc khác nhau).

Sau khi áp dụng mô hình PhoBERT cho việc phân loại cảm xúc, chúng ta đã thu được kết quả như sau:



Hình 35: *Tần xuất/Phản trâm của từng sentiment*

Chú thích: 0 : negative; 1: positive; 2: neutral.

Nhìn chung ta có thể thấy được rằng:

- Điểm mạnh: Quán ta có tỉ lệ bình luận positive rất cao, 67.9% (1271 bình luận). Điều này cho thấy khách hàng hài lòng với dịch vụ của Highland. Đây là một lợi thế cạnh tranh và một động lực để quán ta duy trì và nâng cao chất lượng.

- Điểm yếu: Quán ta còn có tỉ lệ bình luận negative và neutral khá cao, 32.1% (616 bình luận). Điều này cho thấy khách hàng còn có nhiều ý kiến không tốt về quán ta. Quán ta cần tập trung vào những khách hàng này để cải thiện sự hài lòng của họ.

2.4. Khai thác dữ liệu bình luận khách hàng:

Bình luận của khách hàng là một nguồn thông tin quý báu để hiểu rõ hơn về quan điểm và mong ước của họ. Điều này trở nên đặc biệt quan trọng đối với các doanh nghiệp Highlands Coffee, nơi mà sự hài lòng của khách hàng chính là yếu tố then chốt để đạt thành công.

Qua việc phân tích bình luận của khách hàng, chúng ta có khả năng xác định được những yếu tố nào đã đóng góp vào sự hấp dẫn của thương hiệu, cũng như những điểm mạnh mà khách hàng đánh giá cao. Đồng thời, việc này giúp doanh nghiệp nắm bắt được những vấn đề mà khách hàng quan tâm nhất, từ đó có thể phát triển chiến lược và hướng đi phù hợp để cải thiện chất lượng sản phẩm và dịch vụ.

Vì vậy, để khai thác thông tin từ dữ liệu và xác định các yếu tố, từ khóa mà khách hàng thường nhắc đến khi nói về Highlands Coffee, chúng ta cần tập trung vào việc phân tích cột "**Review Comment**" (bình luận của khách hàng). Sau khi hoàn thành các bước của quá trình xử lý dữ liệu, chúng ta sẽ có kết quả như được thể hiện trong cột "tokens." Bây giờ, hãy khám phá những từ khóa mà khách hàng thường sử dụng để mô tả thương hiệu và sản phẩm.

| tokens |
|--|
| [quán, ưng, chuỗi, hl, nhân_viện, thu_ngân, ph...] |
| [không_gian, thoảng, mát, chõ, hơi, muõi, chíc...] |
| [không_gian, thoảng, òn_ào, làm_viec, nói_chuy...] |
| [quán, không_gian, thoảng, view, đường, thoái_...] |
| [khu, satra, bãi, xe, kê, bóng, mát_mè, phòng,...] |
| ... |
| [quán, cà_phê, nhộn_nhip, tháp, viettel, nhân_...] |
| [cà_phê, ngon, môi_trường, xung_quanh, thoái_mái] |
| [địa_diểm, đẹp, view, đẹp, quán, cà] |
| [tiêu_chuẩn] |
| [đóng, đú] |

Hình 36: Cột "Review Comment" sau khi được làm sạch và tách từ

Trước hết, chúng ta sẽ sử dụng một biểu đồ Word Cloud để trực quan hóa các từ khóa mà khách hàng đã sử dụng nhiều nhất trong bình luận của họ. Đồng thời, chúng ta cũng sẽ tính toán và xếp hạng các từ khóa mà khách hàng nhắc đến nhiều nhất như sau:



Hình 37: Word Cloud các từ khóa từ khách hàng bình luận nhiều

Bảng 4: Top 20 từ khoá khách hàng nhắc đến nhiều nhất trong dữ liệu

| | word | appears_in_docs | count | rank | fraction_of_total | appears_in_fraction_of_docs |
|-----|------------|-----------------|-------|------|-------------------|-----------------------------|
| 1 | quán | 335 | 388 | 1.0 | 3.566504 | 17.895299 |
| 153 | ngon | 341 | 345 | 2.0 | 3.171247 | 18.215812 |
| 8 | nhân_vịen | 279 | 315 | 3.0 | 2.895487 | 14.903846 |
| 67 | uống | 280 | 299 | 4.0 | 2.748414 | 14.957265 |
| 36 | không_gian | 261 | 269 | 5.0 | 2.472654 | 13.942308 |
| 40 | xe | 124 | 178 | 6.0 | 1.636180 | 6.623932 |
| 223 | đẹp | 165 | 170 | 7.0 | 1.562644 | 8.814103 |
| 4 | phục_vụ | 140 | 147 | 8.0 | 1.351227 | 7.478632 |
| 31 | đông | 133 | 140 | 9.0 | 1.286883 | 7.104701 |
| 29 | hở | 126 | 135 | 10.0 | 1.240923 | 6.730769 |
| 74 | cafe | 115 | 130 | 11.0 | 1.194963 | 6.143162 |
| 17 | chỗ | 114 | 126 | 12.0 | 1.158195 | 6.089744 |
| 43 | ko | 93 | 116 | 13.0 | 1.066274 | 4.967949 |
| 124 | cà_phê | 103 | 114 | 14.0 | 1.047890 | 5.502137 |
| 151 | đồ | 103 | 106 | 15.0 | 0.974354 | 5.502137 |
| 32 | đi | 89 | 103 | 16.0 | 0.946778 | 4.754274 |
| 133 | ok | 93 | 95 | 17.0 | 0.873242 | 4.967949 |
| 228 | thái_deg | 83 | 92 | 18.0 | 0.845666 | 4.433761 |
| 73 | trà | 75 | 90 | 19.0 | 0.827282 | 4.006410 |
| 47 | ổn | 85 | 88 | 20.0 | 0.808898 | 4.540598 |

Bảng 5: Giải thích các biến

| BIẾN | MÔ TẢ |
|----------------------------|---|
| word | Từ khoá |
| appears_in_doc | Đếm số bình luận mà từ khoá xuất hiện |
| count | Đếm số lượng từ khoá đó xuất hiện |
| rank | Xếp hạng từ khoá theo số lượng từ khoá |
| fraction_of_total | Phần trăm của tổng số lần xuất hiện của mỗi từ khoá đó so với tổng số lần xuất hiện của tất cả các từ khoá. |
| Appears_in_fraction_of_doc | Phần trăm bình luận mà mỗi từ khoá xuất hiện. |

Nhìn chung, ta có thể thấy được rằng các từ khóa "quán," "ngon," "nhân_vịen", "uống" và "không_gian" là top 5 từ khóa xuất hiện nhiều nhất trong bình luận của khách hàng, điều này chứng tỏ rằng khách hàng đánh giá quán với một số yếu tố quan trọng như vị trí, chất lượng thực phẩm, dịch vụ của nhân viên, menu uống, và không gian tổ chức.

2.4.1. Rút trích các từ khoá dựa trên SENTIMENT:

Để xác định các từ khóa liên quan đến một chủ đề cụ thể, chúng ta có thể sử dụng mô hình Word2Vec để phân tích quan hệ giữa các từ trong văn bản. Điều này giúp chúng ta hiểu được từ khóa đó đang liên quan đến chủ đề nào. Để thực hiện điều này, ta có thể sử dụng thư viện Gensim để huấn luyện một mô hình Word2Vec trên các dữ liệu đào tạo tương ứng với từng loại SENTIMENT. Mô hình Word2Vec được tạo bằng cách sử dụng thư viện Gensim.

```
model_pos = Word2Vec(train_data0, vector_size = 100, window = 5, min_count = 1, workers = 4, sg = 1)
```

- Trong đó:

+*vector_size*: Kích thước của các vector biểu diễn từ (100 chiều – giá trị trung bình và thường hoạt động tốt).

+*window*: Số từ xung quanh một từ mà mô hình xem xét trong quá trình học (5 từ - Một giá trị nhỏ như 5 thường tạo ra các biểu diễn từ tốt cho các từ gần nhau trong ngữ cảnh)

+*min_count*: Số lần xuất hiện tối thiểu của một từ trong dữ liệu để nó được coi là từ vựng (1 lần - bao gồm tất cả các từ trong từ vựng, bất kể xem xét một từ một lần duy nhất có thể không mang lại nhiều thông tin hữu ích.).

+*workers*: Số tiến trình được sử dụng trong quá trình huấn luyện (4 tiến trình).

+*sg*: Sử dụng Skip-gram model (*sg*=1) thay vì Continuous Bag of Words (CBOW) model (*sg*=0). (Ta dùng *sg*=1 vì dữ liệu phân theo từng Sentiment rất ít, vậy nên skip-gram sẽ hoạt động tốt hơn và hiệu quả hơn)

▪ Dựa theo theo SENTIMENT - POSITIVE:



Hình 38: Top 20 từ xuất hiện nhiều trong positive reviews

Bảng 6: Top 10 từ khoá xuất hiện nhiều phân theo bình luận positive

| | word | appears_in_docs | count | rank | fraction_of_total | appears_in_fraction_of_docs |
|-----|------------|-----------------|-------|------|-------------------|-----------------------------|
| 109 | ngon | 322 | 325 | 1.0 | 4.939960 | 25.334382 |
| 1 | quán | 216 | 246 | 2.0 | 3.739170 | 16.994493 |
| 36 | không_gian | 225 | 232 | 3.0 | 3.526372 | 17.702596 |
| 84 | uống | 217 | 229 | 4.0 | 3.480772 | 17.073171 |
| 151 | đẹp | 162 | 167 | 5.0 | 2.538380 | 12.745869 |
| 8 | nhân_viện | 139 | 148 | 6.0 | 2.249582 | 10.936271 |
| 140 | cafe | 89 | 98 | 7.0 | 1.489588 | 7.002360 |
| 4 | phục_vụ | 89 | 92 | 8.0 | 1.398389 | 7.002360 |
| 92 | cà_phê | 79 | 88 | 9.0 | 1.337589 | 6.215578 |
| 107 | đồ | 84 | 87 | 10.0 | 1.322389 | 6.608969 |

| | | |
|--|---|---|
| Các từ tương tự của 'ngon': ko: 0.9980214238166809 đi: 0.997794508934021 highland: 0.9977702498435974 quán: 0.9977090358734131 bàn: 0.9976317882537842 chỗ: 0.9975363612174988 nóng: 0.9975297451019287 đông: 0.9974833130836487 trà: 0.9974507093429565 ng: 0.997441291809082 | Các từ tương tự của 'quán': highland: 0.998117983341217 ly: 0.9980405569076538 hơi: 0.9980229139328003 ko: 0.9979332685470581 ng: 0.9979103207588196 đi: 0.9979032874107361 bàn: 0.9978480339050293 bảo_vệ: 0.9978015422821045 ghé: 0.9977902770042419 đồng: 0.9977487921714783 | Các từ tương tự của 'không_gian': siêu_thị: 0.997891366481781 đi: 0.9978626370429993 cafe: 0.9978322386741638 cà_phê: 0.9976934790611267 làm_việc: 0.997594952583313 đông: 0.9975860714912415 nhân_viện: 0.9975646734237671 ko: 0.997515857219696 highland: 0.9974660873413086 ly: 0.9974228739738464 |
|--|---|---|

Hình 39: Các từ liên quan khi nhắc tới từ khoá xuất hiện nhiều trong positive reviews

Dựa trên kết quả phân tích từ khóa trong các bình luận tích cực, ta có thể thấy mối liên hệ giữa các từ khóa và ngữ cảnh mà chúng xuất hiện:

- Từ “**ngon**” thường được sử dụng trong ngữ cảnh liên quan đến trải nghiệm ẩm thực tại các quán cà phê hoặc nhà hàng. Các từ như “**đi**” “**highland**” “**bàn**” và “**chỗ**” có thể chỉ đến việc khách hàng đi đến một quán nước, ngồi tại một bàn, và tận hưởng không gian thoải mái tại đó.
- Từ “**quán**” thường xuất hiện trong ngữ cảnh liên quan đến việc tiêu dùng thức uống tại các quán cà phê. Các từ như “**highland**”, “**ly**,” “**hơi**,” “**đi**” “**bàn**” và “**bảo_vệ**” có thể chỉ đến việc khách hàng đến quán, uống một ly cà phê, tận hưởng không gian yên tĩnh, và cảm nhận sự an toàn do bảo vệ đảm bảo.
- Từ “**không gian**” thường được sử dụng để miêu tả không gian sống, làm việc, hoặc giải trí. Các từ như “**cafe**” “**cà_phê**” “**đi**” và “**đóng**” có thể ám chỉ đến việc khách hàng đi đến một không gian cafe để làm việc hoặc giải trí, và cảm nhận sự đồng đúc, sôi động tại đó.

Nhìn chung, kết quả này cho thấy rằng các từ khóa trong các bình luận tích cực thường liên quan đến trải nghiệm thức uống và không gian tại các quán cà phê Highland. Điều này cho thấy rằng **chất lượng đồ uống** và **không gian quán** là những yếu tố quan trọng để tạo ra trải nghiệm tích cực cho khách hàng.

▪ **Dựa theo theo SENTIMENT - NEGATIVE:**



Hình 40: Top 20 từ xuất hiện nhiều trong negative reviews

Bảng 7: Top 10 từ khoá xuất hiện nhiều phân theo bình luận negative

| | word | appears_in_docs | count | rank | fraction_of_total | appears_in_fraction_of_docs |
|-----|-----------|-----------------|-------|------|-------------------|-----------------------------|
| 7 | nhân_viên | 118 | 139 | 1.0 | 4.839833 | 32.686981 |
| 31 | quán | 81 | 97 | 2.0 | 3.377437 | 22.437673 |
| 78 | thái_độ | 63 | 72 | 3.0 | 2.506964 | 17.451524 |
| 22 | xe | 39 | 60 | 4.0 | 2.089136 | 10.803324 |
| 35 | tệ | 49 | 51 | 5.0 | 1.775766 | 13.573407 |
| 91 | ko | 38 | 51 | 6.0 | 1.775766 | 10.526316 |
| 93 | phục_vụ | 46 | 49 | 7.0 | 1.706128 | 12.742382 |
| 131 | đi | 36 | 43 | 8.0 | 1.497214 | 9.972299 |
| 0 | uống | 36 | 39 | 9.0 | 1.357939 | 9.972299 |
| 193 | đông | 33 | 35 | 10.0 | 1.218663 | 9.141274 |

```
///////////
Các từ tương tự của 'nhân_viên': Các từ tương tự của 'quán':
ly: 0.9464205503463745 thái_độ: 0.9425156116485596
thái_độ: 0.9450235962867737 nhân_viên: 0.9421160221099854
quán: 0.9421160221099854 ly: 0.9415447115898132
đi: 0.9305326342582703 mua: 0.9340441823005676
ko: 0.9280268549919128 xe: 0.9293044805526733
uống: 0.9260905981063843 ko: 0.9269528985023499
xong: 0.9221639037132263 uống: 0.9260505437850952
mua: 0.922062337398529 món: 0.9184523224830627
xe: 0.9194949269294739 đi: 0.9138686656951904
hôm_nay: 0.9178779721260071 hôm_nay: 0.9098653197288513
///////////
Các từ tương tự của 'thái_độ': Các từ tương tự của 'tệ':
ly: 0.9530677795410156 ly: 0.90805584192276
nhân_viên: 0.9450235962867737 thái_độ: 0.9044150114059448
quán: 0.9425155520439148 ko: 0.8970740437507629
ko: 0.941523551940918 quán: 0.8952729105949402
uống: 0.9396374225616455 xe: 0.8948020935058594
mua: 0.9325522780418396 nhân_viên: 0.8911041021347046
xe: 0.9271981716156006 mua: 0.8889247179031372
đi: 0.9253494739532471 uống: 0.8833213448524475
quầy: 0.9223091006278992 bánh: 0.8819595575332642
món: 0.919443666934967 quầy: 0.8803312182426453
/////////
```

Hình 41: Các từ liên quan khi nhắc tới từ khoá xuất hiện nhiều trong negative reviews

Dựa trên kết quả phân tích từ khóa trong các bình luận tiêu cực, ta có thể thấy mối liên hệ giữa các từ khóa và ngữ cảnh mà chúng xuất hiện:

- Từ “tệ” Các từ tương tự như ‘tệ’, chẳng hạn như ‘ly’, ‘thái_độ’, ‘quán’, ‘ko’ (không), ‘mua’, ‘uống’, ‘bánh’, ‘quầy’, đều có tiềm năng tiêu cực hoặc liên quan đến các trải nghiệm

không tốt. Ví dụ, 'ly' và 'thái_độ' có thể liên quan đến phục vụ không tốt hoặc thái độ không thân thiện.

• Từ “nhân_viện” : Các từ tương tự như “ly”, “thái_độ”, “quán”, “đi”, “ko”, “uống”, “xong”, “mua”, “xe”, “hôm_nay” thường liên quan đến nhân viên hoặc dịch vụ khách hàng. Khi kết hợp với từ 'tê' xuất hiện nhiều, chúng cho thấy **thái độ cung nhu cách phục vụ của nhân viên đang gặp vấn đề**.

• Từ “quán” thường được dùng để miêu tả không gian, vị trí, hoặc thương hiệu của một quán cà phê hoặc nhà hàng. Các từ tương tự như “thái_độ”, “nhân_viện”, “ly”, “mua”, “xe”, “ko”, “uống”, “món”, “đi”, “hôm_nay” có thể liên quan đến **trải nghiệm tại quán, đặc biệt về thái độ và dịch vụ từ nhân viên**.

• Từ “thái_độ” Các từ tương tự 'ly', 'nhân_viện', 'quán', 'ko', 'uống', 'mua', 'xe', 'đi', 'quầy', 'món', cho thấy có thể liên quan đến việc khách hàng gặp phải những sự cố, bất tiện, hoặc bất mãn khi đến quán.

Nhìn chung, thông qua việc phân tích các từ khóa trong các bình luận tiêu cực, chúng ta có thể thấy một mối liên hệ rõ ràng giữa những từ này và các khía cạnh tiêu cực của trải nghiệm khách hàng, như **thái độ không tốt** của nhân viên và **dịch vụ không hài lòng** tại quán. Điều này có thể cung cấp thông tin quan trọng cho việc cải thiện chất lượng và hài lòng của khách hàng.

▪ Dựa theo theo SENTIMENT - NEUTRAL:



Hình 42: Top 20 từ xuất hiện nhiều trong neutral reviews

Bảng 8: Top 10 từ khoá xuất hiện nhiều phân theo bình luận neutral

| | word | appears_in_docs | count | rank | fraction_of_total | appears_in_fraction_of_docs |
|-----|------------|-----------------|-------|------|-------------------|-----------------------------|
| 12 | quán | 38 | 45 | 1.0 | 3.151261 | 15.833333 |
| 14 | xe | 28 | 44 | 2.0 | 3.081232 | 11.666667 |
| 41 | uống | 27 | 31 | 3.0 | 2.170868 | 11.250000 |
| 108 | tạm | 30 | 31 | 4.0 | 2.170868 | 12.500000 |
| 5 | nhân_vịen | 22 | 28 | 5.0 | 1.960784 | 9.166667 |
| 17 | chỗ | 25 | 28 | 6.0 | 1.960784 | 10.416667 |
| 92 | hở | 25 | 28 | 7.0 | 1.960784 | 10.416667 |
| 78 | không_gian | 24 | 25 | 8.0 | 1.750700 | 10.000000 |
| 54 | đông | 23 | 24 | 9.0 | 1.680672 | 9.583333 |
| 73 | ko | 18 | 24 | 10.0 | 1.680672 | 7.500000 |

```
//////////  
Các từ tương tự của 'quán':  
báo: 0.4602504074573517  
ko: 0.4509839713573456  
sách_sẽ: 0.39934781193733215  
món: 0.38250428438186646  
đi: 0.3675096333026886  
bố_trí: 0.3631821274757385  
bảo_vệ: 0.3477112650871277  
chi_nhánh: 0.3465645909309387  
bình_thường: 0.336308985948562  
lich_sự: 0.33086156845092773  
/////////  
Các từ tương tự của 'uống':  
hở: 0.41636866331100464  
báo: 0.40253859758377075  
bánh_mì: 0.3728558123111725  
bảo_vệ: 0.36870047450065613  
trà: 0.3576696515083313  
đi: 0.351126104593277  
cf: 0.34776076674461365  
xíu: 0.34269580245018005  
trù: 0.3414020240306854  
tiện: 0.3364580273628235  
/////////  
Các từ tương tự của 'xe':  
hiện_đại: 0.3648311495780945  
món: 0.3552665114402771  
trà: 0.3384890556335449  
kiểu: 0.32587185502052307  
nóng: 0.3241053521633148  
uống: 0.32241013646125793  
sen: 0.31666767597198486  
order: 0.30687645077705383  
highlands: 0.3023349940776825  
bảo_vệ: 0.3017154633998871  
/////////  
Các từ tương tự của 'nhân_vịen':  
hở: 0.3975997865200043  
đá: 0.3919389843940735  
chi_nhánh: 0.37409690022468567  
ổn: 0.3587806820869446  
gửi: 0.3584589660167694  
vị: 0.3537778854370117  
thu: 0.3521116077899933  
xanh: 0.33243849873542786  
bánh_mì: 0.31881147623062134  
lâu: 0.31657880544662476
```

Hình 43: Các từ liên quan khi nhắc tới từ khoá xuất hiện nhiều trong neutral reviews

Dựa trên kết quả phân tích từ khóa trong các bình luận trung lập, ta có thể nhận xét rằng có mối liên hệ và sự liên quan giữa các từ khóa và ngữ cảnh mà chúng xuất hiện. Dưới đây là một số điểm quan trọng:

- Từ "quán" có các từ tương tự như "báo", "ko" (viết tắt của "không"), "sách_sẽ" và "món." Điều này cho thấy những bình luận trung lập liên quan đến **sự sạch sẽ của quán, thực đơn và có thể có một số ý kiến hoặc phản hồi về quán.**

- Từ "xe" có các từ tương tự như "**hiện đại**," "**món**" và "**kiểu**." Điều này có thể ám chỉ đến liên quan đến việc chở gửi xe của quán.

- Từ "**uống**" có các từ tương tự như "**hơi**" "**báo**" và "**bánh mì**." Điều này cho thấy bình luận trung lập có thể liên quan đến việc uống nước, thức uống, hoặc cảm nhận về món ăn, có thể là bánh mì, mà quán cung cấp.

- Từ "**nhân viên**" có các từ tương tự như "**hơi**" "**đá**" và "**chi nhánh**." Điều này có thể chỉ ra sự đánh giá về nhân viên tại chi nhánh cụ thể và có thể liên quan đến dịch vụ hoặc kinh nghiệm tại đó.

Kết luận chung: Kết quả phân tích từ khóa cho thấy các từ khóa chính ("quán" "xe" "uống" và "nhân viên") có **sự liên quan đến các khía cạnh khác nhau của trải nghiệm tại quán** như sạch sẽ, thực đơn, việc sử dụng xe, thức uống, nhân viên, và dịch vụ tại chi nhánh.

2.4.2. Đối chiếu kết hợp từ khoá dựa trên từng loại SENTIMENT:

Ở đây, chúng ta sẽ thử so sánh các từ khoá phổ biến của quán, tức là chọn ra top 3 từ khoá xuất hiện nhiều nhất sẽ làm đại diện cho mỗi sentiment, sau đó kiểm tra xem qua mỗi tình cảm này, các từ khoá đó vẫn có còn đáng là từ khoá đại diện hay không.

+ Đối với bình luận tích cực: ta có từ “ngon”, “quán”, “không gian” .

| | word | appears_in_docs | count | rank | fraction_of_total | appears_in_fraction_of_docs |
|-----|------------|-----------------|-------|------|-------------------|-----------------------------|
| 110 | ngon | 322 | 325 | 1.0 | 4.939960 | 25.334382 |
| 3 | quán | 216 | 246 | 2.0 | 3.739170 | 16.994493 |
| 40 | không_gian | 225 | 232 | 3.0 | 3.526372 | 17.702596 |

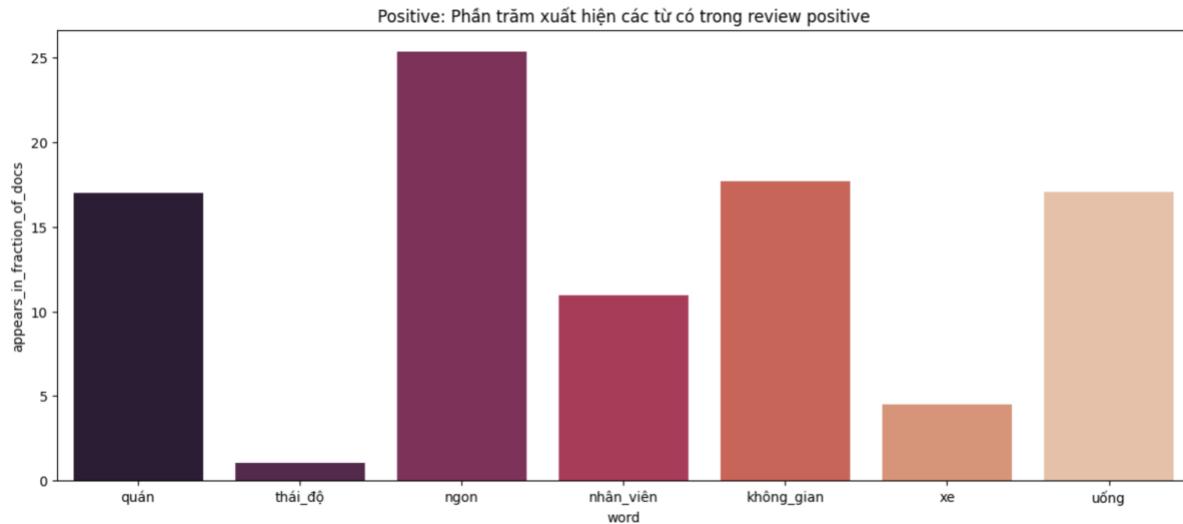
+ Đối với bình luận tiêu cực: ta có từ “nhân viên”, “quán”, “thái độ”.

| | word | appears_in_docs | count | rank | fraction_of_total | appears_in_fraction_of_docs |
|----|-----------|-----------------|-------|------|-------------------|-----------------------------|
| 12 | nhân_vien | 118 | 139 | 1.0 | 4.839833 | 32.686981 |
| 32 | quán | 81 | 97 | 2.0 | 3.377437 | 22.437673 |
| 80 | thái_deg | 63 | 72 | 3.0 | 2.506964 | 17.451524 |

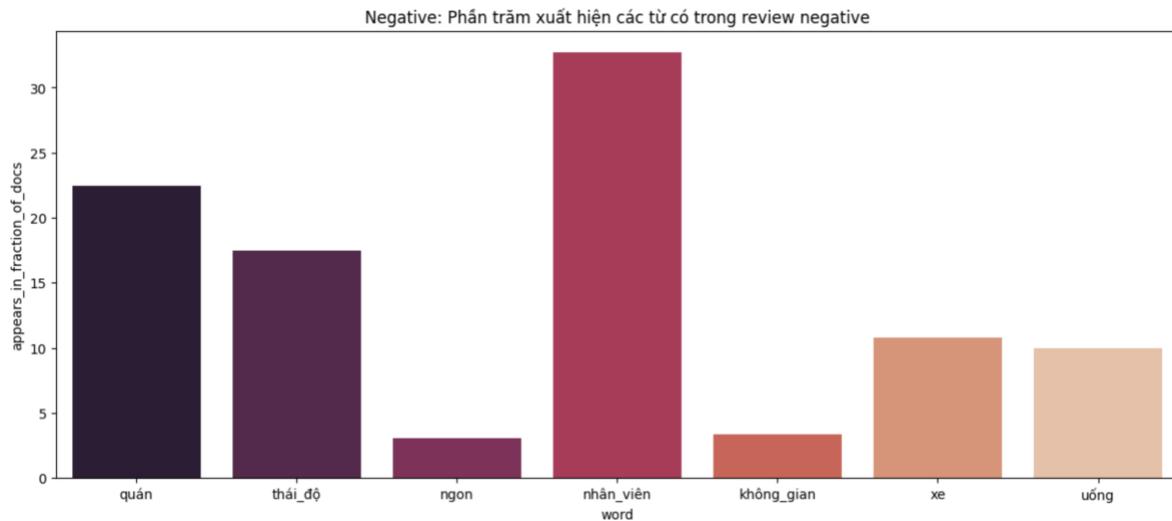
+ Đối với bình luận trung lập ta có từ “quán”, “xe”, “uống”

| | word | appears_in_docs | count | rank | fraction_of_total | appears_in_fraction_of_docs |
|----|------|-----------------|-------|------|-------------------|-----------------------------|
| 12 | quán | 38 | 45 | 1.0 | 3.151261 | 15.833333 |
| 21 | xe | 28 | 44 | 2.0 | 3.081232 | 11.666667 |
| 54 | uống | 27 | 31 | 3.0 | 2.170868 | 11.250000 |

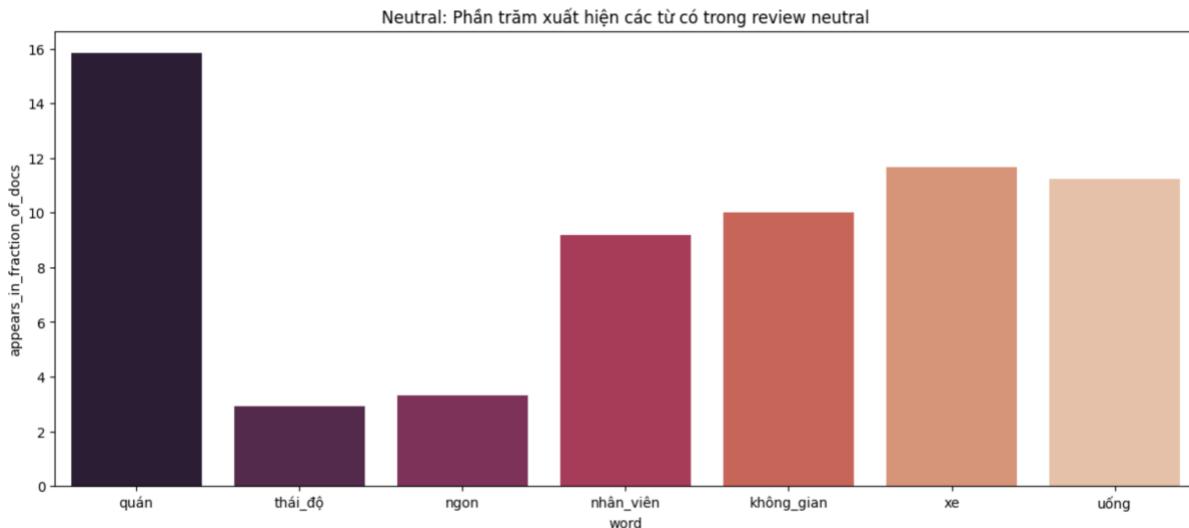
Bây giờ, chúng ta sẽ tiếp tục bằng việc kiểm tra sự xuất hiện của các từ đại diện cho mỗi sentiment trong các sentiment khác nhau.



Hình 44: Phản trǎm xuất hiện của các từ khoá đại diện ở SENTIMENT positive



Hình 45: Phản trǎm xuất hiện của các từ khoá đại diện ở SENTIMENT negative



Hình 46: Phần trăm xuất hiện của các từ khoá đại diện ở SENTIMENT neutral

Qua những biểu đồ trên, chúng ta có thể rút ra một số nhận định quan trọng về các yếu tố trong đánh giá quán:

+Từ "**quán**" không phản ánh sentiment cụ thể nào: Từ "quán" không thể kết nối với một ý kiến cụ thể, bởi vì nó xuất hiện ở mức cao trong tất cả các loại bình luận, bất kể tích cực, tiêu cực hoặc trung lập.

+"**Ngon**" và "**không gian**" đại diện cho các bình luận tích cực. Điều này cho thấy rằng chất lượng đồ uống và không gian quán là điểm mạnh đáng chú ý, và đánh giá tích cực của khách hàng thể hiện điều này.

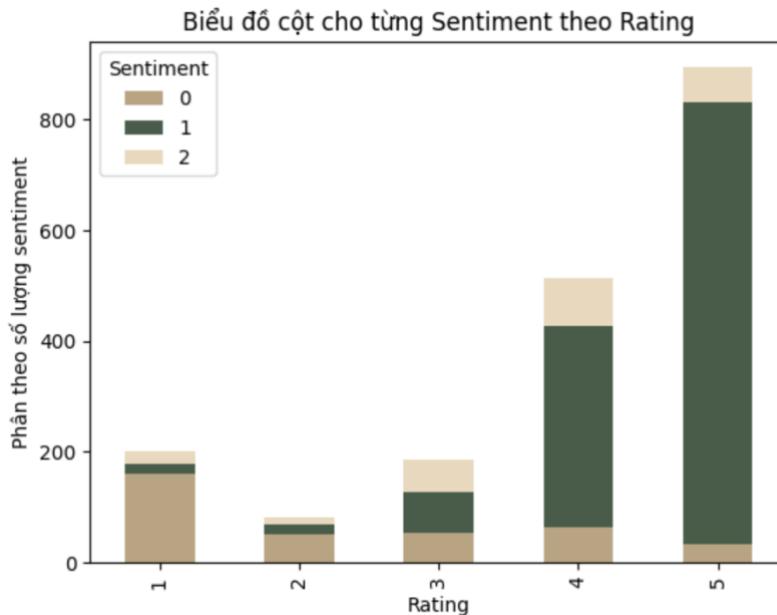
+"**Thái độ**" và "**nhân viên**" thể hiện tiêu cực: 2 từ khoá này xuất hiện nhiều trong các bình luận tiêu cực, chỉ ra rằng thái độ của nhân viên **đang gây ra vấn đề và gây ảnh hưởng tiêu cực đối với trải nghiệm của khách hàng**.

+Bình luận trung lập thể hiện sự cân bằng: Các bình luận trung lập chứa một sự kết hợp của từ khóa đại diện cho các cảm xúc khác nhau. Điều này cho thấy rằng các khía cạnh khác nhau của trải nghiệm tại quán, chẳng hạn như "quán," "xe," "uống," và "nhân viên," **không ảnh hưởng quá mạnh bởi một yếu tố cụ thể nào**.

CHƯƠNG 3: ĐÁNH GIÁ - ĐỀ XUẤT GIẢI PHÁP

3.1. Đánh giá :

♦ Đánh giá mô hình dựa theo kết quả Sentiment với Rating:



Hình 47: Biểu đồ cột tròn cho từng Sentiment theo từng Rating

Dựa vào biểu đồ và sự quan sát, có một số điểm quan trọng cần được bổ sung để phân tích sâu hơn về tình hình phân loại Sentiment của PhoBERT trong các tình huống khác nhau:

+ **Sự mâu thuẫn giữa đánh giá và sentiment ở rating 1-2 sao:** Mâu thuẫn này có thể xuất phát từ việc khách hàng có thể có những quan điểm tích cực hoặc trung lập về một số yếu tố cụ thể trong trải nghiệm của họ, mặc dù tổng thể họ cảm thấy không hài lòng. Ví dụ, họ có thể nhấn mạnh điểm tích cực như dịch vụ của nhân viên, nhưng khi tổng hợp lại, họ vẫn cảm thấy trải nghiệm tổng thể không tốt.

+ **Sự mâu thuẫn giữa đánh giá và sentiment ở rating 4-5 sao:** Sự mâu thuẫn ở đây có thể phản ánh một số hạn chế của hệ thống phân loại Sentiment. Có thể có trường hợp khi PhoBERT không phân biệt được sự sâu rộng của ý kiến trong bình luận, và do đó, một bình luận chứa cả các yếu tố tích cực và tiêu cực có thể được phân loại là tích cực vì đánh giá cuối cùng của khách hàng là 4 hoặc 5 sao.

+ **Phản ánh sự phức tạp của đánh giá khách hàng:** Bên cạnh các trường hợp mâu thuẫn, biểu đồ cũng cho thấy rằng đánh giá của khách hàng có thể phản ánh nhiều yếu tố khác nhau. Có thể có mối tương quan giữa đánh giá và các yếu tố như dịch vụ, giá cả, chất lượng sản phẩm, vị trí cửa hàng, và nhiều yếu tố khác. Điều này đề xuất rằng để hiểu rõ hơn cảm nhận của khách hàng, cần phải xem xét cẩn thận các nội dung cụ thể trong bình luận thay vì chỉ dựa vào điểm số.

=> Tóm lại, việc phân tích Sentiment từ bình luận và điểm đánh giá của khách hàng không phải lúc nào cũng dễ dàng do tính phức tạp và thay đổi của ngôn ngữ và ý kiến. Sự mâu thuẫn trong kết quả phân loại Sentiment có thể xuất phát từ sự đa dạng của quan điểm người dùng và cách họ biểu đạt chúng.

♦ **Dựa trên từng yếu tố - từ khóa phản ánh cho mỗi loại cảm xúc:**

Bảng 9: Phân cụm khách hàng theo yếu tố đại diện từng SENTIMENT

| Cụm | Nhóm khách hàng |
|-----------------------|--|
| 1: positive sentiment | <ul style="list-style-type: none"> + Những người trong nhóm này thường sử dụng các từ như "ngon" và "không gian" để mô tả trải nghiệm của họ tại quán. + Họ có xu hướng tập trung vào chất lượng đồ uống và không gian của quán. + Đánh giá của họ thường tích cực và thể hiện sự hài lòng với những khía cạnh tích cực của quán. |
| 0: negative sentiment | <ul style="list-style-type: none"> + Những người trong nhóm này thường sử dụng các từ khóa như "thái độ" và "nhân viên" để bày tỏ sự không hài lòng. + Họ có xu hướng tập trung vào thái độ và hiệu suất của nhân viên tại quán. + Đánh giá của họ thường tiêu cực và thể hiện sự không hài lòng về mặt dịch vụ và tương tác với nhân viên. |
| 2: neutral sentiment | <ul style="list-style-type: none"> + Những người trong nhóm này có bình luận trung lập, không dùng nhiều từ khóa tích cực hoặc tiêu cực. |

| | |
|--|--|
| | <ul style="list-style-type: none"> + Họ có thể tập trung vào nhiều khía cạnh của trải nghiệm tại quán mà không ảnh hưởng quá mạnh bởi một yếu tố cụ thể nào. + Đánh giá của họ thể hiện sự cân bằng và không phụ thuộc quá nhiều vào một khía cạnh cụ thể. |
|--|--|

3.2. Đề xuất giải pháp:

Dựa vào bảng phân cụm khách hàng trên, ta có một số đề xuất giải pháp như sau để có thể cải thiện trải nghiệm của khách hàng tại quán:

+ Đối với Nhóm Khách Hàng Tích Cực:

- Tiếp tục duy trì và cải thiện chất lượng đồ uống và không gian quán để đáp ứng nhu cầu của khách hàng.
- Tạo ra các chương trình khuyến mãi hoặc ưu đãi dành riêng cho những khách hàng thường xuyên để tăng sự gắn kết và lòng trung thành của họ.

+ Đối với Nhóm Khách Hàng Tiêu Cực:

- Đào tạo lại nhân viên về thái độ phục vụ và kỹ năng giao tiếp để cải thiện trải nghiệm của khách hàng.
- Xây dựng một hệ thống phản hồi khách hàng hiệu quả, nơi khách hàng có thể gửi phản hồi và khiếu nại của họ. Điều này không chỉ giúp quán cải thiện dịch vụ, mà còn cho khách hàng thấy rằng quán quan tâm đến ý kiến của họ.

+ Đối với Nhóm Khách Hàng Trung Lập:

- Tìm hiểu thêm về những yếu tố mà nhóm này quan tâm thông qua các cuộc khảo sát hoặc phỏng vấn trực tiếp.
- Thử nghiệm và điều chỉnh các yếu tố khác nhau của trải nghiệm khách hàng (như menu, giờ mở cửa, sự kiện) để xem liệu có thể tăng cường Sentiment tích cực từ nhóm này không.

KẾT LUẬN

Trong bài nghiên cứu này, ta đã thành công nghiên cứu và triển khai mô hình phoBERT để phân tích cảm xúc khách hàng của chuỗi HighLand coffee quận 10. Nhờ đó, doanh nghiệp có cái nhìn rõ hơn về 3 nhóm khách hàng hiện nay của Highland: nhóm tích cực, nhóm tiêu cực và nhóm trung lập. Ta cũng đã đề xuất các giải pháp cụ thể để cải thiện trải nghiệm của từng nhóm khách hàng, bằng cách tập trung vào các yếu tố quan trọng như chất lượng đồ uống, không gian quán, thái độ phục vụ và phản hồi khách hàng. Đồng thời, ta cũng đã tăng sự hài lòng và gắn kết của khách hàng, đồng thời thu hút thêm khách hàng mới.

Tuy nhiên, điều quan trọng là không chỉ tập trung vào việc cải thiện những điểm yếu mà còn phải duy trì và nâng cao những điểm mạnh. Đồng thời, việc lắng nghe và tiếp thu phản hồi từ khách hàng cũng rất quan trọng để không ngừng hoàn thiện. Bài nghiên cứu này đã chứng minh được sự khác biệt hiệu quả cao và tiềm năng lớn của mô hình phoBERT trong việc phân tích cảm xúc khách hàng. Tuy nhiên, mô hình vẫn có thể được nâng cao hơn nữa bằng cách sử dụng dữ liệu huấn luyện lớn hơn và đa dạng hơn, hoặc kết hợp với các phương pháp khác để đánh giá Sentiment một cách vượt trội toàn diện và chính xác.

Hy vọng bài nghiên cứu này sẽ góp phần vào việc nâng cao chất lượng dịch vụ và kinh doanh của chuỗi HighLand coffee quận 10!

TÀI LIỆU THAM KHẢO

PhamDinhThanh, Bài 39 - Thực hành ứng dụng BERT, 04 Jun 2020, nguồn:
https://phamdinhkhanh.github.io/2020/06/04/PhoBERT_Fairseq.html#:~:text=RoBERTa%20là%20một%20project%20của,chuông%20bởi%20công%20đồng%20AI.

GitHub, VinAIResearch/PhoBERT: PhoBERT: Pre-trained language models for Vietnamese (EMNLP-2020 Findings), October 22, 2023, nguồn:
<https://github.com/VinAIResearch/PhoBERT>

Trần Hồng Việt, Nguyễn Thu Hiền; Mô Hình Transformers Và Ứng Dụng Trong Xử Lý Ngôn Ngữ Tự Nhiên , 20/07/2020, nguồn:
<https://sti.vista.gov.vn/tw/Lists/TaiLieuKHCN/Attachments/341919/CVv15S272022032.pdf>

College of Information and Communication Technology , PhoBERT: Application in Disease Classification based on Vietnamese Symptom Analysis , June 2023, nguồn:
<https://intapi.sciendo.com/pdf/10.2478/acss-2023-0004>

Dat Quoc Nguyen , Anh Tuan Nguyen, PhoBERT: Pre-trained language models for Vietnamese , 16 November 2020, nguồn: <https://aclanthology.org/2020.findings-emnlp.92.pdf>

GitHub, vncorenlp/VnCoreNLP: A Vietnamese natural language processing toolkit (NAACL 2018), October 22, 2023, nguồn: <https://github.com/vncorenlp/VnCoreNLP>

PhamDinhKhanh, Bài 3 - Mô hình Word2Vec, 29 Apr 2019, nguồn:
<https://phamdinhkhanh.github.io/2019/04/29/ModelWord2Vec.html>

Vinh phạm, Selenium là gì? Tổng quan những thông tin cần biết về Selenium, 28-03-2022, nguồn: <https://bizflycloud.vn/tin-tuc/selenium-la-gi-20220328105303215.htm>