

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT



NGUYỄN QUANG VINH

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

MÃ SỐ: 7480201

TÊN ĐỀ TÀI:

NGHIÊN CỨU ỨNG DỤNG HỌC SÂU VÀO BÀI TOÁN
PHÂN LOẠI BÌNH LUẬN

(A Deep Learning Approach to Comment Classification)

Hà Nội – 2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT

ĐỒ ÁN TỐT NGHIỆP

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH ỨNG DỤNG

CÁN BỘ HƯỚNG DẪN
GV. TS. NGUYỄN THẾ LỘC
BỘ MÔN CÔNG NGHỆ PHẦN MỀM

SINH VIÊN THỰC HIỆN
NGUYỄN QUANG VINH
LỚP KHMTUD B K64

HÀ NỘI – 2023

LỜI CẢM ƠN

Được sự đồng ý của bộ môn Khoa học máy tính khoa Công nghệ thông tin của trường Đại học Mỏ - Địa chất và giảng viên hướng dẫn khoa học: TS. Nguyễn Thế Lộc, tôi đã thực hiện đồ án: “Nghiên cứu ứng dụng học sâu vào bài toán phân loại bình luận”.

Để hoàn thành đồ án này, tôi xin chân thành cảm ơn các thầy cô giảng viên trong khoa Công nghệ thông tin, đặc biệt là bộ môn Khoa học máy tính đã tận tình giảng dạy, hướng dẫn tôi trong suốt quá trình học tập và nghiên cứu ở trường.

Xin chân thành cảm ơn thầy giáo Nguyễn Thế Lộc, người đã trực tiếp hướng dẫn nghiên cứu khoa học cho tôi. Trong quá trình thực hiện đồ án, thầy đã tận tình chỉ bảo và truyền đạt những kinh nghiệm và kiến thức khoa học quý báu, đồng thời đưa ra các nhận xét, góp ý giúp tôi hoàn thành đồ án này.

Xin chân thành cảm ơn các anh chị, bạn bè, các thành viên lớp Khoa học máy tính ứng dụng 64B đã ủng hộ, giúp đỡ tôi trong thời gian học tập ở trường và tạo điều kiện làm đồ án tốt nghiệp.

MỤC LỤC

LỜI CẢM ƠN	i
MỤC LỤC.....	1
DANH MỤC TỪ VIẾT TẮT	3
DANH MỤC BẢNG BIÊU.....	4
DANH MỤC HÌNH VẼ	5
CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI.....	9
1.1. Lý do chọn đề tài	9
1.2. Mục tiêu của đề tài	10
1.3. Nội dung nghiên cứu	10
1.4. Phạm vi nghiên cứu	10
1.5. Bố cục của đồ án	11
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ	12
2.1. Giới thiệu xử lý ngôn ngữ tự nhiên	12
2.1.1. Ngôn ngữ tự nhiên.....	12
2.1.2. Ngôn ngữ tiếng Việt	12
2.1.3. Xử lý ngôn ngữ tự nhiên	14
2.1.4. Các ứng dụng của xử lý ngôn ngữ tự nhiên	15
2.2. Bài toán phân loại văn bản	18
2.2.1. Giới thiệu bài toán	18
2.2.2. Ứng dụng của phân loại văn bản	19
2.2.3. Quy trình các bước xây dựng hệ thống phân loại văn bản.....	20
2.3. Mô hình học sâu BERT để phân loại văn bản.....	21
2.3.1. Tìm hiểu về BERT	21
2.3.2. Biểu diễn dữ liệu đầu vào.....	22
2.3.3. Mô hình BERT được đào tạo trước.....	25
2.3.4. Tinh chỉnh mô hình BERT cho bài toán phân loại văn bản	29
2.3.5. Thông số đánh giá mô hình phân loại	31
2.4. Thu thập dữ liệu	34
2.4.1. Giới thiệu về cào dữ liệu từ trang web	34
2.4.2. Tìm hiểu về công nghệ phát triển web	34

2.4.3. Kỹ thuật tìm dữ liệu cho web	39
2.4.4. Làm việc với bảo mật web	42
2.5. Môi trường phát triển	44
2.5.1. Ngôn ngữ lập trình python	44
2.5.2. Các thư viện hỗ trợ	44
2.5.3. Các phần mềm và công cụ.....	49
CHƯƠNG 3. PHÂN LOẠI BÌNH LUẬN VỚI HỌC SÂU.....	52
3.1. Phát biểu bài toán	52
3.2. Thu thập dữ liệu	53
3.2.1. Giới thiệu diễn đàn VOZ.....	53
3.2.2. Cào dữ liệu trang web VOZ	54
3.2.3. Gán nhãn dữ liệu	59
3.2.4. Mô tả tập dữ liệu	66
3.3. Chuẩn bị dữ liệu	68
3.3.1. Phân tích dữ liệu.....	68
3.3.2. Tiền xử lý dữ liệu	74
3.3.3. Phân tách tập đào tạo và tập kiểm thử	75
3.4. Xây dựng mô hình.....	76
3.4.1. Thông số cấu hình máy tính để đào tạo mô hình	76
3.4.2. Đào tạo mô hình	76
3.5. Đánh giá mô hình	82
3.6. Dự đoán với mô hình đã xây dựng được	83
3.7. So sánh với mô hình học máy truyền thống	84
KẾT LUẬN	89
TÀI LIỆU THAM KHẢO	90

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
BERT	Bidirectional Encoder Representation from Transformers, biểu diễn bộ mã hóa hai chiều từ Transformers
CAPTCHA	“Completely Automated Public Turing test to tell Computers and Humans Apart”, phép thử Turing công cộng hoàn toàn tự động để phân biệt máy tính với người
CPU	Central Processing Unit, bộ xử lý trung tâm
DDoS	Distributed Denial-of-Service, từ chối dịch vụ phân tán
DOM	Document Object Model, mô hình đối tượng tài liệu
GPU	Graphics Processing Unit, bộ xử lý đồ họa
HTML	HyperText Markup Language, ngôn ngữ đánh dấu siêu văn bản
HTTP	Hypertext Transfer Protocol, giao thức truyền tải siêu văn bản
IP	Internet Protocol, giao thức Internet
KPI	Key Performance Indicators, chỉ số hiệu suất chính
MLM	Masked Language Model, mô hình ngôn ngữ che giấu
MultinomialNB	Multinomial Naive Bayes, thuật toán phân loại theo xác suất dựa trên định lý Bayes
NER	Named Entity Recognition, nhận dạng thực thể được đặt tên
NFC	Near-Field Communications, kết nối trường gần
NLP	Natural Language Processing, xử lý ngôn ngữ tự nhiên
NSP	Next Sentence Prediction, dự đoán câu tiếp theo
RAM	Random Access Memory, bộ nhớ truy cập ngẫu nhiên
Regex	Regular expression, biểu thức chính quy
TF-IDF	Term Frequency - Inverse Document Frequency, tần suất thuật ngữ - nghịch đảo tần suất văn bản
URL	Uniform Resource Locator, định vị tài nguyên thống nhất
Wi-Fi	Wireless Fidelity, truyền tín hiệu bằng sóng vô tuyến thông qua kết nối không dây

DANH MỤC BẢNG BIỂU

Bảng 3.1. Kết quả mô hình học sâu và mô hình học máy trên tập kiểm thử.....88

DANH MỤC HÌNH VẼ

Hình 2.1. Nhận dạng thực thể được đặt tên.....	15
Hình 2.2. Trích xuất từ khóa.....	15
Hình 2.3. Mô hình hóa chủ đề.....	16
Hình 2.4. Ứng dụng lọc email rác trong phân loại văn bản.	16
Hình 2.5. Tóm tắt văn bản.....	17
Hình 2.6. Phát hiện ngôn ngữ.....	17
Hình 2.7. Phân tích cảm xúc.	17
Hình 2.8. Quy trình xây dựng hệ thống phân loại văn bản.	20
Hình 2.9. Kiến trúc cấp cao của BERT.	22
Hình 2.10. Mã hóa từ.....	23
Hình 2.11. Phân đoạn nhúng.	23
Hình 2.12. Vị trí nhúng.	24
Hình 2.13. Biểu diễn mã hóa cuối cùng.	24
Hình 2.14. Mô hình ngôn ngữ che giấu.....	28
Hình 2.15. Mô hình dự đoán mã che giấu.	28
Hình 2.16. Tinh chỉnh mô hình BERT cho bài toán phân loại văn bản.	30
Hình 2.17. Unnormalized confusion matrix và normalized confusion matrix....	31
Hình 2.18. True Positive, False Positive, True Negative, False Negative.	32
Hình 2.19. Cách tính precision và recall cho phân loại nhị phân.....	33
Hình 2.20. Công thức tính F1-score.	34
Hình 2.21. Cách giao tiếp giữa máy chủ và máy khách.....	35
Hình 2.22. Công cụ nhà phát triển trên trình duyệt Chrome.	35
Hình 2.23. Tiêu đề yêu cầu.	36
Hình 2.24. Tiêu đề phản hồi.	36
Hình 2.25. Dữ liệu cookie.	37
Hình 2.26. Mã HTML.	38
Hình 2.27. Nhúng mã JavaScript vào HTML.	39
Hình 2.28. Nguồn trang YouTube.....	40
Hình 2.29. Kỹ thuật lấy phần tử.	41
Hình 2.30. Tệp robots.txt trang web Facebook.	42

Hình 2.31. Một số CAPTCHA.....	43
Hình 3.1. Giao diện một chủ đề của diễn đàn VOZ.....	52
Hình 3.2. Thống kê lượt truy cập của diễn đàn theo Similarweb.....	53
Hình 3.3. Thống kê nhân khẩu học của diễn đàn theo Similarweb.....	53
Hình 3.4. Xác thực CAPTCHA của Cloudflare.....	54
Hình 3.5. Khởi tạo selenium với hồ sơ trình duyệt.....	54
Hình 3.6. Hàm chọn phần tử và xóa phần tử.....	55
Hình 3.7. Bình luận bị thu gọn.....	55
Hình 3.8. Hàm hiển thị phần tử thu gọn.....	55
Hình 3.9. Nội dung trích dẫn trùng lặp với bình luận trước đó.....	56
Hình 3.10. Hàm xóa nội dung trùng lặp.....	56
Hình 3.11. Thông tin về thành viên của diễn đàn.....	56
Hình 3.12. Hàm lấy thông tin thành viên.....	57
Hình 3.13. Hàm lấy văn bản bình luận.....	57
Hình 3.14. Hàm tái trang tiếp theo.....	58
Hình 3.15. Hàm lưu dữ liệu vào một sheet vào tệp Excel.....	58
Hình 3.16. Hàm tổng hợp thu thập dữ liệu.....	58
Hình 3.17. Bộ dữ liệu sau khi thu thập.....	59
Hình 3.18. Gộp các sheet thành một sheet tổng hợp.....	62
Hình 3.19. Chuyển dữ liệu sang tệp văn bản.....	62
Hình 3.20. Giao diện cấu hình tùy chỉnh với Label Studio.....	63
Hình 3.21. Giao diện gán nhãn.....	63
Hình 3.22. Dữ liệu tệp JSON.....	64
Hình 3.23. Tính toán nhãn tổng thể.....	65
Hình 3.24. Mô tả tập dữ liệu thô.....	66
Hình 3.25. Dữ liệu được gán nhãn.....	67
Hình 3.26. Mô tả tập dữ liệu sau gán nhãn.....	67
Hình 3.27. Chuyển đổi dữ liệu ngày tháng sang dạng chuẩn.....	68
Hình 3.28. Phân phối số lượng bình luận theo ngày.....	68
Hình 3.29. Thống kê độ dài của bình luận.....	69
Hình 3.30. Thông tin về số lượng người bình luận.....	70

Hình 3.31. Thông tin về danh hiệu của các thành viên.	70
Hình 3.32. Thống kê ngày đăng ký làm thành viên.	71
Hình 3.33. Phân phối tổng số lượng bình luận mỗi thành viên.	71
Hình 3.34. Thống kê về số điểm phản ứng.	72
Hình 3.35. Xử lý dữ liệu điểm phản ứng ngoại lai.	72
Hình 3.36. Phân phối về số điểm phản ứng.	72
Hình 3.37. Thống kê số lượng từng khía cạnh.	73
Hình 3.38. Phân tích tỷ lệ tích cực và tiêu cực từng khía cạnh.	73
Hình 3.39. Kiểm tra cân bằng tập dữ liệu đã gán nhãn.	74
Hình 3.40. Tiền xử lý văn bản bình luận.	74
Hình 3.41. Tách tập dữ liệu ra thành tập đào tạo và tập kiểm thử.	75
Hình 3.42. Thông tin tập đào tạo và tập kiểm thử.	75
Hình 3.43. Kết nối Google Colab với Google Drive.	76
Hình 3.44. Sử dụng GPU để đào tạo mô hình.	76
Hình 3.45. Lấy dữ liệu từ tệp.	77
Hình 3.46. Sử dụng bộ mã hóa phoBERT.	77
Hình 3.47. Chuẩn bị dữ liệu đầu vào cho mô hình học sâu.	78
Hình 3.48. Xây dựng cấu trúc mô hình học sâu.	79
Hình 3.49. Hàm đào tạo mô hình học sâu.	79
Hình 3.50. Hàm đánh giá mô hình học sâu.	80
Hình 3.51. Đào tạo mô hình học sâu.	81
Hình 3.52. Kết quả đào tạo mô hình học sâu từ một K-fold.	82
Hình 3.53. Kết quả đánh giá mô hình học sâu.	82
Hình 3.54. Trực quan hóa kết quả đánh giá mô hình.	82
Hình 3.55. Hàm dự đoán dữ liệu mới.	83
Hình 3.56. Kết quả dự đoán mô hình học sâu trên dữ liệu mới.	84
Hình 3.57. Mã hóa bình luận với TF-IDF.	85
Hình 3.58. Hàm xây dựng mô hình MultinomialNB.	85
Hình 3.59. Tìm giá trị alpha cho mô hình MultinomialNB.	86
Hình 3.60. Kết quả precision, recall, F1-score của mô hình học máy.	86
Hình 3.61. Kết quả độ chính xác accuracy của mô hình học máy.	86

Hình 3.62. Trực quan hóa kết quả kiểm thử mô hình học máy.....	87
Hình 3.63. Kết quả dự đoán của mô hình học máy trên dữ liệu mới.	87

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Lý do chọn đề tài

Sự tiện lợi trên chiếc điện thoại thông minh nhỏ gọn, có kết nối mạng tạo ra sự phát triển mạnh mẽ của thương mại điện tử, mạng xã hội, trang web chia sẻ thông tin và các sản phẩm trực tuyến khác. Những yếu tố này đã tạo ra khối lượng thông tin khổng lồ, đặc biệt là dạng văn bản để ta có thể khai thác và tận dụng. Lượng dữ liệu về đánh giá, nhận xét về các sản phẩm trên các trang thương mại điện tử, mạng xã hội là rất lớn. Từ đây xuất hiện một yêu cầu tổng hợp những bình luận, đánh giá và nhận xét để phân tích và thống kê được mức độ hài lòng đối với sản phẩm và dịch vụ. Từ đó, các doanh nghiệp có thể cải thiện nâng cấp dịch vụ và thích ứng được với sự biến động không ngừng của thị trường.

Trước đây, để giải quyết yêu cầu trên, mọi người thường phải sử dụng những cách thủ công, tốn nhiều thời gian và nguồn lực vì lượng dữ liệu phải xử lý là rất lớn. Do đó, họ không thể nhanh chóng đưa ra được những quyết định phù hợp với nhu cầu thị trường.

Học sâu đã ra đời để xử lý lượng dữ liệu lớn, cho ra kết quả tốt nhanh chóng được ứng dụng trong xử lý ngôn ngữ tự nhiên. Các nghiên cứu về học sâu đã tạo ra các mô hình ngôn ngữ, giúp thực hiện việc xử lý dữ liệu dễ dàng hơn.

Phân loại văn bản là một bài toán nổi bật trong xử lý ngôn ngữ tự nhiên, được ứng dụng rộng rãi. Ví dụ, phân loại bình luận, lọc tin tức, phân loại sản phẩm, phát hiện tin rác hoặc lừa đảo... Trong đó, phân loại bình luận tập trung vào các nhận xét, bình luận, đánh giá trên các sàn thương mại điện tử như Lazada, Shopee hoặc các mạng xã hội như Facebook, LinkedIn. Mục đích của việc phân loại bình luận là sắp xếp hoặc gán những bình luận vào các nhóm dựa trên tiêu chí nhất định. Ví dụ, bình luận có thể được phân loại thành các loại cảm xúc như tích cực hoặc tiêu cực, để đánh giá sự hài lòng hoặc không hài lòng của người dùng đối với một sản phẩm hoặc dịch vụ cụ thể.

Phân loại bình luận có nhiều thách thức. Tiêu biểu là tính đa nghĩa của ngôn ngữ, từ viết tắt, sự không rõ ràng làm cho việc phân loại trở nên phức tạp và khó

khăn. Để giải quyết cần kết hợp các kỹ thuật tiền xử lý dữ liệu để cải thiện mô hình phân loại.

Sự hấp dẫn của bài toán, những thách thức đã chỉ ra ở trên, với niềm đam mê công nghệ hiện đại và những ứng dụng rộng rãi của nó, với khát khao khám phá và chinh phục tri thức... tôi đã chọn “Nghiên cứu ứng dụng học sâu vào bài toán phân loại bình luận” làm đề tài nghiên cứu đồ án tốt nghiệp.

1.2. Mục tiêu của đề tài

Đề tài “Nghiên cứu ứng dụng học sâu vào bài toán phân loại bình luận”, với mục tiêu là lấy dữ liệu bình luận từ một trang web, thực hiện các bước xử lý dữ liệu để đánh giá sơ bộ chất lượng nội dung bình luận của web, phân tích thống kê và trích rút thông tin hữu ích từ tập dữ liệu đã lấy được. Dựa vào tập dữ liệu để xây dựng mô hình học sâu và dự đoán các bình luận khác một cách tự động.

1.3. Nội dung nghiên cứu

Để đạt được các mục tiêu đã nêu trên, đồ án có những nội dung sau:

Về mặt lý thuyết:

- Nghiên cứu và tổng hợp lý thuyết về xử lý ngôn ngữ tự nhiên.
- Bài toán phân loại văn bản và phân loại bình luận.
- Các ứng dụng của bài toán phân loại văn bản.
- Mô tả một số phương pháp và mô tả từng bước triển khai thu thập dữ liệu, bước xử lý và xây dựng mô hình học sâu hoàn chỉnh.

Về mặt thực hành:

- Tạo môi trường thực hành để ứng dụng các lý thuyết vào bài toán cụ thể: Cài đặt ngôn ngữ lập trình python, các công cụ, phần mềm và thư viện hỗ trợ để giải quyết bài toán.
- Xây dựng mô hình có khả năng dự đoán các bình luận tương tự.
- Chạy thử nghiệm mô hình đã xây dựng được.

1.4. Phạm vi nghiên cứu

Phân loại bình luận là bài toán có nhiều ứng dụng trong thực tế. Trong đồ án này, tôi chỉ tập trung nghiên cứu phân loại bình luận tiếng Việt. Các bình luận

là các đánh giá sản phẩm trên trang web, được gán nhãn thủ công theo hai nhãn lớp là tích cực hoặc tiêu cực.

1.5. Bố cục của đồ án

Bố cục của đồ án được trình bày với các nội dung chính như sau:

Chương 1. Tổng quan về đề tài

Chương này giới thiệu về mục tiêu, ý nghĩa, phạm vi và tóm lược những nội dung của đồ án.

Chương 2. Cơ sở lý thuyết và công nghệ

Chương này giới thiệu về xử lý ngôn ngữ tự nhiên, bài toán phân loại bình luận, xây dựng mô hình, các công cụ, thư viện lập trình để triển khai trong bài toán cụ thể.

Chương 3. Phân loại bình luận với học sâu

Chương này trình bày từng bước cụ thể để giải quyết bài toán phân loại bình luận. Bao gồm thu thập dữ liệu, gán nhãn bình luận, tiền xử lý dữ liệu, khám phá dữ liệu, đào tạo mô hình và sử dụng mô hình để dự đoán các bình luận mới.

Kết luận

Phần này tôi tổng kết lại các kết quả và những đóng góp mà việc thực hiện đề tài này đem lại. Ngoài ra, tổng kết những việc chưa làm được cần khắc phục, đề xuất các phương hướng nghiên cứu tiếp theo, làm cho đề tài trở lên hoàn thiện và hữu ích hơn.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

2.1. Giới thiệu xử lý ngôn ngữ tự nhiên

2.1.1. Ngôn ngữ tự nhiên

Ngôn ngữ là hệ thống những âm, những từ và những quy tắc kết hợp chung mà những người trong một cộng đồng dùng làm phương tiện để giao tiếp với nhau [1]. Trong ngôn ngữ học, ngôn ngữ tự nhiên là ngôn ngữ nào phát sinh, không suy nghĩ trước trong não bộ của con người. Một số ngôn ngữ điển hình mà con người được sử dụng để giao tiếp với nhau, có thể ngôn ngữ âm thanh, ngôn ngữ ký hiệu, các ký hiệu xúc giác hay chữ viết [2].

Các phương pháp cơ bản thường được áp dụng trong so sánh ngôn ngữ là:

- Phương pháp so sánh lịch sử. Mục đích của nó là phát hiện những nét phản ảnh quan hệ thân thuộc, gần gũi về nguồn gốc giữa các ngôn ngữ để quy chúng vào những phô hệ ngôn ngữ cụ thể khác nhau.
- Phương pháp so sánh loại hình. Mục đích chính của nó là nghiên cứu những đặc trưng của các loại hình ngôn ngữ và nghiên cứu những đặc trưng về mặt loại hình của các ngôn ngữ, để quy các ngôn ngữ cụ thể vào những loại hình khác nhau.
- Phương pháp so sánh đối chiếu. Mục đích phát hiện những tương đồng và khác biệt chủ yếu trên diện đồng đại ở một hay nhiều bình diện, bộ phận của các ngôn ngữ đó [3].

2.1.2. Ngôn ngữ tiếng Việt

Tiếng Việt thuộc ngôn ngữ đơn lập, tức là mỗi một tiếng (âm tiết) được phát âm tách rời nhau và được thể hiện bằng một chữ viết. Đặc điểm này thể hiện rõ rệt ở tất cả các mặt ngữ âm, từ vựng, ngữ pháp.

Đặc điểm ngữ âm

Trong tiếng Việt có một loại đơn vị đặc biệt gọi là tiếng. Về mặt ngữ âm, mỗi tiếng là một âm tiết. Hệ thống âm vị tiếng Việt phong phú và có tính cân đối, tạo ra tiềm năng của ngữ âm tiếng Việt trong việc thể hiện các đơn vị có nghĩa.

Nhiều từ tượng hình, tượng thanh có giá trị gợi tả đặc sắc. Khi tạo câu, tạo lời, người Việt rất chú ý đến sự hài hoà về ngữ âm, đến nhạc điệu của câu văn.

Đặc điểm từ vựng

Mỗi tiếng, nói chung, là một yếu tố có nghĩa. Tiếng là đơn vị cơ sở của hệ thống các đơn vị có nghĩa của tiếng Việt. Từ tiếng, người ta tạo ra các đơn vị từ vựng khác để định danh sự vật, hiện tượng... chủ yếu nhờ phương thức ghép và phương thức láy.

Việc tạo ra các đơn vị từ vựng ở phương thức ghép luôn chịu sự chi phối của quy luật kết hợp ngữ nghĩa, ví dụ: đất nước, máy bay, nhà lầu xe hơi, nhà tan cửa nát... Hiện nay, đây là phương thức chủ yếu để sản sinh ra các đơn vị từ vựng. Theo phương thức này, tiếng Việt triệt để sử dụng các yếu tố cấu tạo từ thuần Việt hay vay mượn từ các ngôn ngữ khác để tạo ra các từ, ngữ mới, ví dụ: tiếp thị, karaoke, thư điện tử (e-mail), thư thoại (voice mail), phiên bản (version), xa lộ thông tin, siêu liên kết văn bản, truy cập ngẫu nhiên...

Việc tạo ra các đơn vị từ vựng ở phương thức láy thì quy luật phối hợp ngữ âm chi phối chủ yếu việc tạo ra các đơn vị từ vựng, chẳng hạn: chòm chìa, chòng chờ, đồng đa đồng đánh, thơ thẩn, lúng lá lúng liêng...

Vốn từ vựng tối thiểu của tiếng Việt phần lớn là các từ đơn tiết (một âm tiết, một tiếng). Sự linh hoạt trong sử dụng, việc tạo ra các từ ngữ mới một cách dễ dàng đã tạo điều kiện thuận lợi cho sự phát triển vốn từ, vừa phong phú về số lượng, vừa đa dạng trong hoạt động. Cùng một sự vật, hiện tượng, một hoạt động hay một đặc trưng, có thể có nhiều từ ngữ khác nhau biểu thị. Tiềm năng của vốn từ ngữ tiếng Việt được phát huy cao độ trong các phong cách chức năng ngôn ngữ, đặc biệt là trong phong cách ngôn ngữ nghệ thuật. Hiện nay, do sự phát triển vượt bậc của khoa học-kỹ thuật, đặc biệt là công nghệ thông tin, thì tiềm năng đó còn được phát huy mạnh mẽ hơn.

Đặc điểm ngữ pháp

Từ của tiếng Việt không biến đổi hình thái. Đặc điểm này sẽ chi phối các đặc điểm ngữ pháp khác. Khi từ kết hợp từ thành các kết cấu như ngữ, câu, tiếng Việt rất coi trọng phương thức trật tự từ và hư từ.

Việc sắp xếp các từ theo một trật tự nhất định là cách chủ yếu để biểu thị các quan hệ cú pháp. Trong tiếng Việt khi nói “Anh ta lại đến” là khác với “Lại đến anh ta”. Khi các từ cùng loại kết hợp với nhau theo quan hệ chính phụ thì từ đứng trước giữ vai trò chính, từ đứng sau giữ vai trò phụ. Nhờ trật tự kết hợp của từ mà “củ cải” khác với “cải củ”, “tình cảm” khác với “cảm tình”. Trật tự chủ ngữ đứng trước, vị ngữ đứng sau là trật tự phổ biến của kết cấu câu tiếng Việt.

Phương thức hư từ cũng là phương thức ngữ pháp chủ yếu của tiếng Việt. Nhờ hư từ mà tổ hợp “anh của em” khác với tổ hợp “anh và em”, “anh vì em”. Hư từ cùng với trật tự từ cho phép tiếng Việt tạo ra nhiều câu cùng có nội dung thông báo cơ bản như nhau nhưng khác nhau về sắc thái biểu cảm. Ví dụ, so sánh các câu sau đây:

- Ông ấy không hút thuốc.
- Thuốc, ông ấy không hút.
- Thuốc, ông ấy cũng không hút.

Ngoài trật tự từ và hư từ, tiếng Việt còn sử dụng phương thức ngữ điệu. Ngữ điệu giữ vai trò trong việc biểu hiện quan hệ cú pháp của các yếu tố trong câu, nhờ đó nhằm đưa ra nội dung muốn thông báo. Trên văn bản, ngữ điệu thường được biểu hiện bằng dấu câu. Ta thử so sánh 2 câu sau để thấy sự khác nhau trong nội dung thông báo:

- Đêm hôm qua, cầu gãy.
- Đêm hôm, qua cầu gãy.

Qua một số đặc điểm nổi bật vừa nêu trên đây, ta có thể hình dung được phần nào bản sắc và tiềm năng của tiếng Việt [4].

2.1.3. Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (tiếng Anh: Natural Language Processing, viết tắt: NLP) đề cập đến lĩnh vực lập trình máy tính để cho phép xử lý và phân tích ngôn ngữ tự nhiên. Từ một thứ cơ bản như một chương trình máy tính để đếm số lượng từ trong một đoạn văn bản, đến thứ phức tạp hơn như một chương trình phục vụ trả lời các câu hỏi của con người hoặc dịch giữa các ngôn ngữ, tất cả đều đủ điều

kiện là NLP. Về cơ bản, bất kể mức độ khó, bất kỳ nhiệm vụ nào liên quan đến máy tính xử lý ngôn ngữ thông qua chương trình đủ điều kiện là NLP [5].

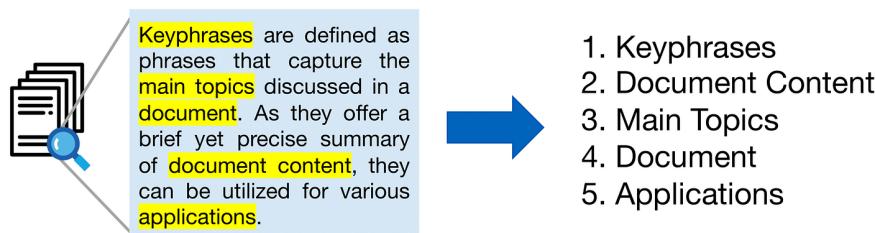
2.1.4. Các ứng dụng của xử lý ngôn ngữ tự nhiên

Nhận dạng thực thể được đặt tên (tiếng Anh: Named Entity Recognition, viết tắt: NER): Một hình thức xử lý ngôn ngữ tự nhiên và còn được gọi là trích xuất thực thể, nhận dạng thực thể hoặc theo đuôi thực thể. Kỹ thuật này xác định các phân đoạn thông tin chính trong một phần văn bản và phân loại các phân đoạn thành các danh mục được xác định trước như tên người, vị trí, ngày, dấu thời gian, tên tổ chức, tỷ lệ phần trăm, mã, số...



Hình 2.1. Nhận dạng thực thể được đặt tên.

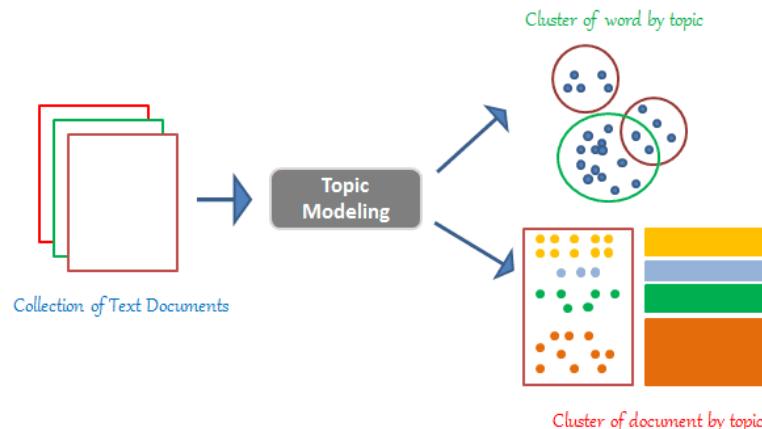
Trích xuất cụm từ khóa: Một tác vụ xử lý thông tin văn bản, liên quan đến việc trích xuất tự động các cụm từ đại diện và đặc trưng từ một tài liệu, thể hiện tất cả các khía cạnh chính của nội dung của nó. Nó nhằm mục đích đại diện cho một bản tóm tắt khái niệm ngắn gọn của một tài liệu văn bản. Các ứng dụng là hệ thống quản lý thông tin kỹ thuật số để lập chỉ mục ngữ nghĩa, tìm kiếm, phân cụm tài liệu và phân loại.



Hình 2.2. Trích xuất từ khóa.

Mô hình hóa chủ đề: Quá trình xác định các chủ đề khác nhau từ một tập hợp các tài liệu bằng cách phát hiện các mẫu từ và cụm từ. Nó được ứng dụng

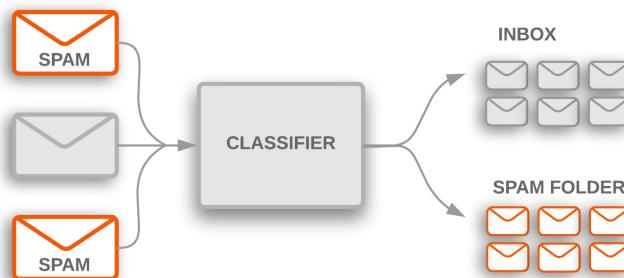
trong phân cụm tài liệu, tổ chức văn bản, truy xuất thông tin từ văn bản không cấu trúc và lựa chọn tính năng.



Hình 2.3. Mô hình hóa chủ đề.

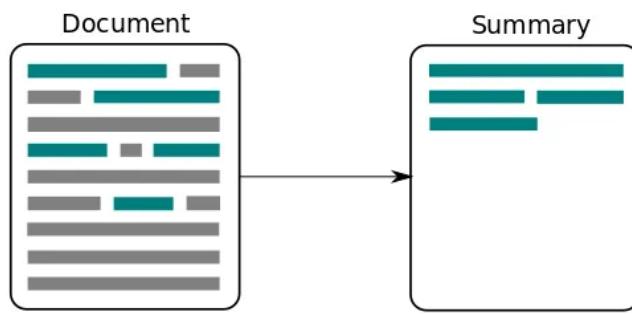
Sự tương đồng về văn bản: Một ứng dụng NLP phổ biến được sử dụng trong các hệ thống phụ thuộc vào việc tìm kiếm các tài liệu có mối quan hệ gần gũi. Một ví dụ phổ biến là các đề xuất nội dung trên các nền tảng truyền thông xã hội.

Phân loại văn bản: Phân loại thành các danh mục do người dùng xác định. Đơn giản như nhãn nhị phân đến hàng trăm và hàng ngàn danh mục. Ví dụ bao gồm phân loại nội dung truyền thông xã hội thành các chủ đề và phân loại khiếu nại của người tiêu dùng trong dịch vụ khách hàng.



Hình 2.4. Ứng dụng lọc email rác trong phân loại văn bản.

Tóm tắt văn bản: Những văn bản dài như bài báo, bài báo hoặc tài liệu được cô đọng thành một bản tóm tắt nhằm giữ lại thông tin quan trọng bằng các kỹ thuật tóm tắt văn bản. Google News và nhiều ứng dụng tổng hợp tin tức khác tận dụng các thuật toán tóm tắt văn bản.



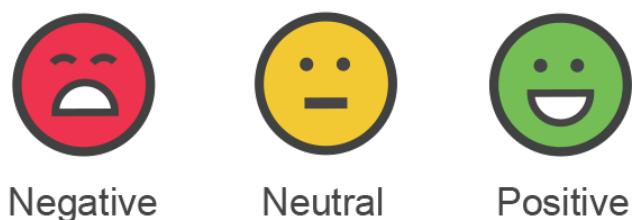
Hình 2.5. Tóm tắt văn bản.

Phát hiện ngôn ngữ và dịch thuật: Có nhiều mô hình được đào tạo trước cho nhiều nhiệm vụ ngôn ngữ có thể được sử dụng ngay. Hầu hết các mô hình được đào tạo trên văn bản của một ngôn ngữ cụ thể. Những mô hình như vậy cũng không thực hiện tốt nếu được sử dụng trên văn bản của một ngôn ngữ khác. Trong những trường hợp đó, ta thường dùng đến các kỹ thuật phát hiện ngôn ngữ và dịch thuật. Các kỹ thuật như vậy cũng được sử dụng trong các công cụ dịch ngôn ngữ để giúp mọi người giao tiếp bằng các ngôn ngữ không bản địa.

That is life	English
C'est la vie	Portuguese
Yahee jeevan hai	Hindi
Bubomi obo	isiXhosa
それが人生です	Japanese

Hình 2.6. Phát hiện ngôn ngữ.

Phân tích cảm xúc: Đánh giá tình cảm (hoặc cảm xúc) của con người trong một câu. Ứng dụng là hiểu tình cảm của người tiêu dùng đối với một sản phẩm hoặc nội dung [5].



Hình 2.7. Phân tích cảm xúc.

2.2. Bài toán phân loại văn bản

2.2.1. Giới thiệu bài toán

Tất cả chúng ta kiểm tra email mỗi ngày, có thể nhiều lần. Một tính năng hữu ích của hầu hết các nhà cung cấp dịch vụ email là khả năng tự động tách các email rác ra khỏi email thông thường. Đây là trường hợp sử dụng của một tác vụ NLP phổ biến được gọi là phân loại văn bản.

Phân loại văn bản là nhiệm vụ gán một hoặc nhiều danh mục cho một đoạn văn bản nhất định từ một tập hợp các danh mục. Trong ví dụ nhận dạng email rác, ta có hai danh mục rác và không phải rác và mỗi email đến được gán cho một trong các danh mục này. Phân loại văn bản được ứng dụng trên các lĩnh vực khác nhau, như phương tiện truyền thông xã hội, thương mại điện tử, chăm sóc sức khỏe, luật pháp và tiếp thị. Mặc dù mục đích và ứng dụng phân loại văn bản có thể khác nhau, vấn đề trừu tượng cơ bản vẫn giữ nguyên. Điều này liên quan đến vấn đề cốt lõi và các ứng dụng của nó trong vô số miền, làm cho phân loại văn bản cho đến nay là nhiệm vụ NLP được sử dụng rộng rãi và được nghiên cứu nhiều.

Trong học máy, phân loại là chia một thể hiện dữ liệu thành một hoặc nhiều lớp đã biết. Điểm dữ liệu ban đầu có thể thuộc các định dạng khác nhau, chẳng hạn như văn bản, lời nói, hình ảnh hoặc số. Phân loại văn bản là một ví dụ đặc biệt của vấn đề phân loại, trong đó các điểm dữ liệu đầu vào là văn bản và mục tiêu là chia đoạn văn bản thành một hoặc nhiều lớp từ một tập hợp các lớp được xác định trước. Văn bản có thể có độ dài tùy ý: một ký tự, một từ, một câu, một đoạn văn hoặc một tài liệu. Chẳng hạn, phân loại tất cả các đánh giá của khách hàng cho một sản phẩm thành ba loại: tích cực, tiêu cực, và trung tính. Thách thức của phân loại văn bản là học cách phân loại từ một tập hợp các ví dụ cho từng loại và dự đoán các danh mục cho các sản phẩm mới, chưa từng thấy và đánh giá khách hàng mới.

Bất kỳ phương pháp phân loại có giám sát nào, bao gồm phân loại văn bản, có thể được phân biệt thành ba loại dựa trên số lượng các danh mục liên quan: phân loại nhị phân, đa lớp và đa nhãn. Nếu số lượng lớp là hai, thì nó được gọi là phân loại nhị phân. Nếu số lượng lớp nhiều hơn hai lớp, thì nó được gọi là phân loại đa lớp. Như vậy, việc phân loại một email là rác hoặc không phải rác là một ví dụ về phân loại nhị phân. Phân loại tình cảm của một đánh giá của khách hàng

là tích cực, tiêu cực hoặc trung tính là một ví dụ về phân loại đa lớp. Trong cả phân loại nhị phân và đa lớp, mỗi tài liệu thuộc chính xác một lớp từ C, trong đó C là tập hợp của tất cả các lớp có thể. Trong phân loại đa nhãn, một tài liệu có thể có một hoặc nhiều nhãn lớp đồng thời. Ví dụ, một bài báo tin tức về một trận đấu bóng đá có thể thuộc nhiều hơn một hạng mục, chẳng hạn như “thể thao” và “bóng đá”. Một bài báo về cuộc bầu cử có thể có các nhãn “chính trị”, “bầu cử”. Theo đó, mỗi tài liệu có các nhãn là một tập hợp con của C. Mỗi bài viết không có trong lớp, chỉ một lớp, nhiều lớp hoặc tất cả các lớp. Đôi khi, số lượng nhãn trong tập C có thể rất lớn. Trong một số ngữ cảnh, ta có thể có một hệ thống phân loại phân cấp, có thể dẫn đến mỗi văn bản nhận được các nhãn khác nhau ở các cấp độ khác nhau trong hệ thống phân cấp [6].

2.2.2. *Ứng dụng của phân loại văn bản*

Phân loại và tổ chức nội dung: Điều này đề cập đến nhiệm vụ phân loại, gắn thẻ một lượng lớn dữ liệu văn bản. Nó được sử dụng để cung cấp sức mạnh cho các trường hợp như tổ chức nội dung, công cụ tìm kiếm và hệ thống đề xuất. Ví dụ về dữ liệu như vậy bao gồm các trang web tin tức, giá sách trực tuyến, đánh giá sản phẩm; gắn thẻ mô tả sản phẩm trong một trang web thương mại điện tử; tổ chức các email vào các chương trình khuyến mãi cá nhân, xã hội.

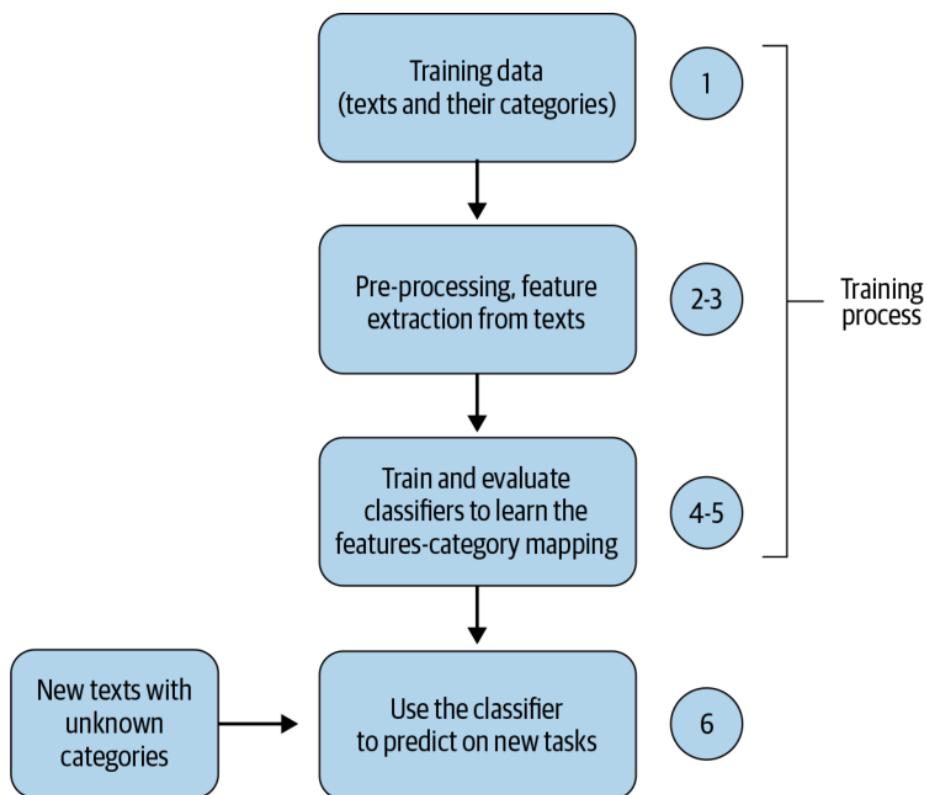
Thương mại điện tử: Khách hàng để lại đánh giá cho các sản phẩm trên các trang thương mại điện tử như Amazon, eBay... Một ví dụ sử dụng phân loại văn bản trong ngữ cảnh này là để hiểu và phân tích khách hàng về một sản phẩm hoặc dịch vụ dựa trên bình luận. Điều này thường được gọi là phân tích tình cảm. Nó được sử dụng rộng rãi bởi các thương hiệu trên toàn cầu, để hiểu rõ hơn về việc họ có thể đến gần hơn hay xa hơn với khách hàng của họ. Thay vì phân loại phản hồi đơn giản là tích cực, tiêu cực hoặc trung tính, trong một khoảng thời gian, phân tích tình cảm đã phát triển thành một mô hình phức tạp hơn: phân tích tình cảm dựa trên khía cạnh. Người làm việc với phân tích tình cảm đã nhận ra rằng nhiều sản phẩm hoặc dịch vụ có nhiều khía cạnh. Để hiểu được tình cảm tổng thể, việc hiểu mỗi khía cạnh đều quan trọng.

Các ứng dụng khác: Ngoài các lĩnh vực được đề cập ở trên, phân loại văn bản cũng được sử dụng trong một số lĩnh vực khác như nhận dạng ngôn ngữ, phân tách tin tức giả mạo khỏi tin tức thực sự.

2.2.3. Quy trình các bước xây dựng hệ thống phân loại văn bản

Người ta thường tuân theo các bước dưới đây khi xây dựng hệ thống phân loại văn bản:

1. Thu thập hoặc tạo một bộ dữ liệu được gán nhãn phù hợp cho nhiệm vụ.
 2. Chia bộ dữ liệu thành hai (đào tạo và kiểm thử) hoặc ba phần (đào tạo, xác thực và kiểm thử) sau đó quyết định về các số liệu đánh giá.
 3. Chuyển đổi văn bản thô thành các vectơ đặc trưng.
 4. Đào tạo một bộ phân loại bằng cách sử dụng các vectơ đặc trưng và các nhãn tương ứng từ bộ đào tạo.
 5. Sử dụng các số liệu đánh giá từ Bước 2, điểm chuẩn hiệu suất mô hình trên bộ thử nghiệm.
 6. Triển khai mô hình để sử dụng trong thế giới thực và giám sát hiệu suất của nó.



Hình 2.8. Quy trình xây dựng hệ thống phân loại văn bản.

Các bước 3 đến 5 được lắp lại để khám phá các biến thể khác nhau của các đặc trưng, thuật toán phân loại, các tham số của chúng và để tinh chỉnh các siêu tham số trước khi tăng lên Bước 6, triển khai mô hình tối ưu trong sản phẩm.

Khi các hệ thống phân loại được triển khai trong các ứng dụng trong thế giới thực, các chỉ số hiệu suất chính (tiếng Anh: key performance indicators, viết tắt: KPI) dành riêng cho trường hợp sử dụng kinh doanh nhất định cũng được sử dụng để đánh giá tác động của chúng và lợi tức đầu tư.

2.3. Mô hình học sâu BERT để phân loại văn bản

2.3.1. Tìm hiểu về BERT

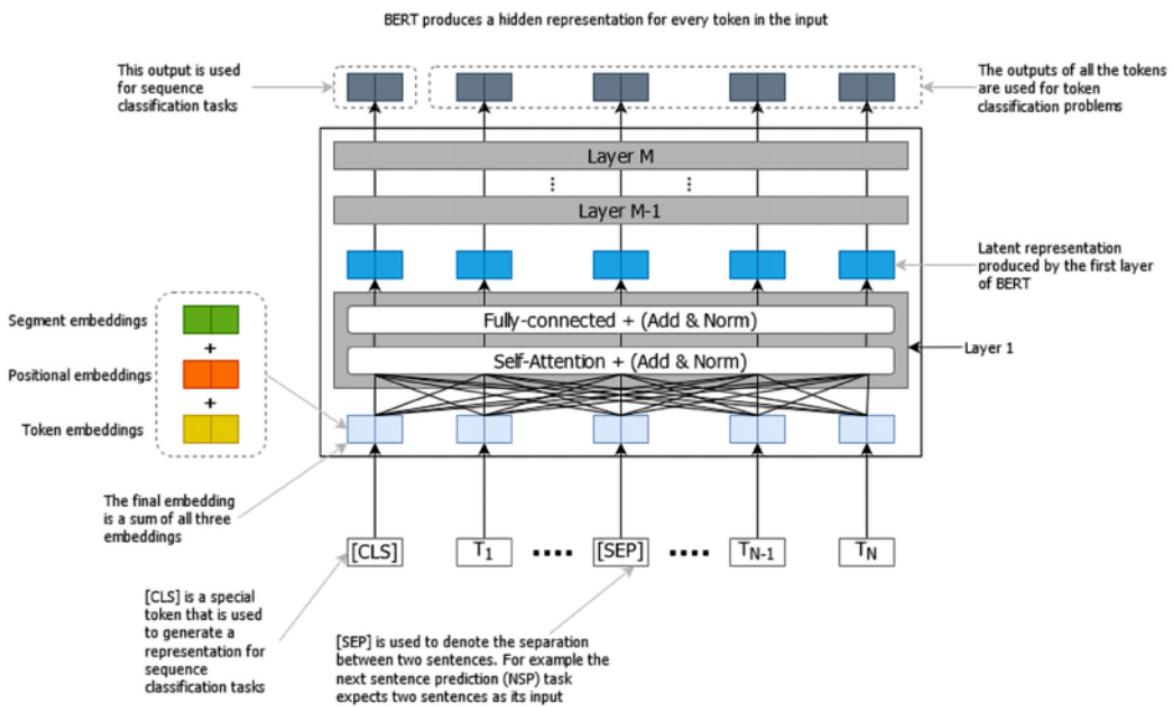
BERT (viết tắt của từ tiếng Anh: Bidirectional Encoder Representation from Transformers, biểu diễn bộ mã hóa hai chiều từ transformers) là một mô hình transformer trong số rất nhiều mô hình transformer trong vài năm qua.

BERT đã được giới thiệu trong bài báo “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” bởi Delvin et al. Các mô hình transformer được chia thành hai phe chính:

- Các mô hình dựa trên bộ mã hóa.
- Các mô hình dựa trên bộ giải mã.

Nói cách khác, bộ mã hóa hoặc bộ giải mã của transformer cung cấp cơ sở cho các mô hình này, so với sử dụng cả bộ mã hóa và bộ giải mã. Sự khác biệt chính giữa hai phe là cách sử dụng sự chú ý. Các mô hình dựa trên bộ mã hóa sử dụng sự chú ý hai chiều, trong khi các mô hình dựa trên bộ giải mã sử dụng sự chú ý Auto-regressive (nghĩa là từ trái sang phải).

BERT là một mô hình transformer dựa trên bộ mã hóa. Nó có một chuỗi đầu vào (một tập hợp các mã) và tạo ra một chuỗi đầu ra được mã hóa. Hình sau mô tả kiến trúc cấp cao của BERT [7]:



Hình 2.9. Kiến trúc cấp cao của BERT.

2.3.2. Biểu diễn dữ liệu đầu vào

Trước khi cung cấp đầu vào cho BERT, ta chuyển đổi đầu vào thành nhúng bằng ba lớp nhúng được chỉ ra ở đây:

- Mã nhúng (Token embedding).
- Phân đoạn nhúng (Segment embedding).
- Vị trí nhúng (Position embedding).

Mã nhúng

Đầu tiên, ta có một tầng mã nhúng. Giả sử ta có hai câu sau:

- Câu A: Paris is a beautiful city.
- Câu B: I love Paris.

Đầu tiên, ta mã hóa cả hai câu và có được mã:

tokens = [Paris, is, a, beautiful, city, I, love, Paris]

Tiếp theo, ta thêm một mã mới, gọi là mã [CLS], chỉ ở đầu câu đầu tiên:

tokens = [[CLS], Paris, is, a, beautiful, city, I, love, Paris]

Sau đó, ta thêm một mã mới có tên [SEP] ở cuối mỗi câu:

tokens = [[CLS], Paris, is, a, beautiful, city, [SEP], I, love, Paris, [SEP]]

Lưu ý rằng mã [CLS] chỉ được thêm vào đầu câu đầu tiên, trong khi mã [SEP] được thêm vào cuối mỗi câu. Mã [CLS] được sử dụng cho các tác vụ phân loại và mã [SEP] được sử dụng để chỉ ra kết thúc của mỗi câu.

Bây giờ, trước khi cấp tất cả các mã cho BERT, ta chuyển đổi các mã thành nhúng bằng cách sử dụng một tầng nhúng được gọi là mã nhúng (token embedding). Các giá trị của mã nhúng sẽ được học trong quá trình đào tạo. Trong sơ đồ sau, ta đã nhúng cho tất cả các mã, nghĩa là, $E_{[CLS]}$ chỉ ra việc nhúng mã [CLS], E_{Paris} chỉ ra việc nhúng mã Paris...

Input	[CLS]	Paris	is	a	beautiful	city	[SEP]	I	love	Paris	[SEP]
Token embeddings	E_{CLS}	E_{Paris}	E_{is}	E_a	$E_{\text{beautiful}}$	E_{city}	$E_{[\text{SEP}]}$	E_I	E_{love}	E_{Paris}	$E_{[\text{SEP}]}$

Hình 2.10. Mã hóa từ.

Phân đoạn nhúng

Tiếp theo, ta có một tầng phân đoạn nhúng. Phân đoạn nhúng được sử dụng để phân biệt giữa hai câu đã cho. Ta lấy hai câu trong phần trước:

- Câu A: Paris is a beautiful city.
- Câu B: I love Paris.

Sau khi mã hóa hai câu trước đó, ta sẽ có những câu sau:

$\text{tokens} = [[\text{CLS}], \text{Paris}, \text{is}, \text{a}, \text{beautiful}, \text{city}, [\text{SEP}], \text{I}, \text{love}, \text{Paris}, [\text{SEP}]]$

Bây giờ, ta phải đưa ra một số loại chỉ số cho mô hình để phân biệt giữa hai câu. Để làm điều này, ta cấp các mã đầu vào cho tầng phân đoạn nhúng.

Tầng phân đoạn nhúng chỉ trả về một trong hai loại nhúng, E_A hoặc E_B , dưới dạng đầu ra. Nghĩa là, nếu mã đầu vào thuộc câu A, thì mã sẽ được ánh xạ vào việc nhúng E_A và nếu mã thuộc câu B, thì nó sẽ được ánh xạ vào việc nhúng E_B .

Input	[CLS]	Paris	is	a	beautiful	city	[SEP]	I	love	Paris	[SEP]
Segment embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B

Hình 2.11. Phân đoạn nhúng.

Vị trí nhúng

Tiếp theo, ta có một tầng vị trí nhúng. Ta biết rằng BERT về cơ bản là bộ mã hóa của transformer, nên ta cần cung cấp thông tin về vị trí của các từ (mã)

trong câu trước khi cấp trực tiếp đến BERT. Vì vậy, ta sử dụng một tầng gọi là tầng vị trí nhúng.

Trong sơ đồ sau, E_0 cho biết vị trí nhúng của mã [CLS], E_1 cho biết vị trí nhúng của mã Paris...

Input	[CLS]	Paris	is	a	beautiful	city	[SEP]	I	love	Paris	[SEP]
Position embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

Hình 2.12. Vị trí nhúng.

Biểu diễn cuối cùng

Trong sơ đồ sau, trước tiên ta chuyển đổi các câu đầu vào đã cho thành mã, cấp các mã này cho các tầng mã nhúng, phân đoạn nhúng và vị trí nhúng để thu được các nhúng. Tiếp theo, ta tổng hợp tất cả các nhúng lại với nhau và cung cấp chúng làm đầu vào cho BERT:

Input	[CLS]	Paris	is	a	beautiful	city	[SEP]	I	love	Paris	[SEP]
Token embeddings	$E_{[CLS]}$	E_{Paris}	E_{is}	E_a	$E_{\text{beautiful}}$	E_{city}	$E_{[\text{SEP}]}$	E_I	E_{love}	E_{Paris}	$E_{[\text{SEP}]}$
Segment embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B
Position embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

Hình 2.13. Biểu diễn mã hóa cuối cùng.

WordPiece tokenizer

BERT sử dụng một loại bộ mã đặc biệt được gọi là WordPiece tokenizer. WordPiece tokenizer tuân theo sơ đồ mã từ con. Xem xét câu sau:

- “Let us start pretraining the model.”

Bây giờ, nếu ta mã hóa câu bằng cách sử dụng bộ mã WordPiece, từ pretraining được chia thành các từ con - pre, ##train, ##ing. ta sẽ thu được các mã như được hiển thị ở đây:

tokens = [let, us, start, pre, ##train, ##ing, the, model]

Khi ta sử dụng bộ mã WordPiece, trước tiên ta kiểm tra xem từ này có trong từ vựng hay không. Nếu có trong từ vựng, thì ta sử dụng nó như một mã. Nếu

không có trong từ vựng, thì ta chia từ thành các từ con và ta kiểm tra xem từ con có trong từ vựng hay không. Nếu từ con có trong từ vựng, thì ta sử dụng nó như một mã. Nhưng nếu từ con không có trong từ vựng, thì một lần nữa ta chia từ con và kiểm tra xem nó có trong từ vựng hay không. Nếu nó có trong từ vựng, thì ta sử dụng nó như một mã, nếu không ta lại chia nó lần nữa. Theo cách này, ta tiếp tục chia tách và kiểm tra từ với từ vựng cho đến khi ta tiếp cận các ký tự riêng lẻ.

Trong ví dụ, từ pretraining không có trong từ vựng của BERT. Do đó, ta chia từ pretraining thành các từ con pre, ##train, và ##ing. Các ký hiệu trước mã ##train và ##ing cho biết đó là một từ con và nó được đi trước bởi các từ khác. Nay giờ ta kiểm tra xem các từ con ##train và ##ing có trong từ vựng hay không. Vì chúng có mặt trong từ vựng, ta không chia tách nữa và sử dụng chúng làm mã.

Do đó, bằng cách sử dụng bộ mã WordPiece, ta có được các mã sau:

tokens = [let, us, start, pre, ##train, ##ing, the, model]

Tiếp theo, ta thêm một mã [CLS] ở đầu câu và mã [SEP] ở cuối câu:

tokens = [[CLS], let, us, start, pre, ##train, ##ing, the model, [SEP]]

Nay giờ, tương tự phần trước, ta cung cấp mã cho tầng mã, phân đoạn và vị trí nhúng, thu được các nhúng, tổng hợp các nhúng, sau đó cung cấp chúng dưới dạng đầu vào cho BERT.

2.3.3. Mô hình BERT được đào tạo trước

Giá trị thực của BERT xuất phát từ thực tế là nó đã được đào tạo trước trên một kho dữ liệu lớn theo kiểu tự giám sát. Trong giai đoạn trước khi đào tạo, BERT được đào tạo trên hai nhiệm vụ khác nhau:

- Mô hình ngôn ngữ che giấu.
- Dự đoán câu tiếp theo.

Mô hình ngôn ngữ

Trong nhiệm vụ mô hình ngôn ngữ, ta đào tạo mô hình để dự đoán từ tiếp theo được cho bởi một chuỗi các từ. Ta có thể phân loại mô hình ngôn ngữ thành hai khía cạnh:

- Mô hình ngôn ngữ Auto-regressive.
- Mô hình ngôn ngữ Auto-encoding.

Mô hình ngôn ngữ Auto-regressive

Ta có thể phân loại mô hình ngôn ngữ Auto-regressive như sau:

- Dự đoán tiến (trái sang phải).
- Dự đoán ngược (từ phải sang trái).

Xem xét văn bản: “Paris is a beautiful city. I love Paris”. Ví dụ, loại bỏ từ “city” và thêm trống, như được hiển thị ở đây:

Paris is a beautiful ___. I love Paris

Bây giờ, mô hình phải dự đoán chỗ trống. Nếu ta sử dụng dự đoán tiến, thì mô hình sẽ đọc tất cả các từ, từ trái sang phải đến chỗ trống để đưa ra dự đoán:

Paris is a beautiful ___.

Nếu ta sử dụng dự đoán ngược, thì mô hình sẽ đọc tất cả các từ, từ phải sang trái đến chỗ trống để đưa ra dự đoán:

___. I love Paris

Do đó, các mô hình Auto-regressive là đơn hướng trong tự nhiên, có nghĩa là chúng đọc câu chỉ theo một hướng.

Mô hình ngôn ngữ Auto-encoding

Mô hình ngôn ngữ Auto-encoding tận dụng dự đoán của cả hai dự đoán từ tiến (từ trái sang phải) và ngược (từ phải sang trái). Nghĩa là, nó đọc câu theo cả hai hướng trong khi đưa ra dự đoán. Vì vậy, ta có thể nói rằng mô hình ngôn ngữ Auto-encoding là hai chiều (bidirectional) trong tự nhiên. Ta có thể quan sát, để dự đoán chỗ trống, mô hình ngôn ngữ Auto-encoding đọc câu theo cả hai hướng, nghĩa là, từ trái sang phải và từ phải sang trái:

Paris is a beautiful ___. I love Paris

Mô hình hai chiều cho kết quả tốt hơn bởi vì nếu ta đọc câu từ cả hai hướng, nó sẽ cho ta sự rõ ràng hơn về mặt hiểu được câu.

Mô hình ngôn ngữ che giấu (MLM)

BERT là một mô hình ngôn ngữ Auto-encoding, có nghĩa là nó đọc câu theo cả hai hướng để đưa ra dự đoán. Trong một tác vụ mô hình ngôn ngữ che giấu, trong một câu đầu vào nhất định, ta ngẫu nhiên che giấu 15% từ và đào tạo mạng để dự đoán các từ bị che giấu. Mô hình sẽ đọc câu theo cả hai hướng và cố gắng dự đoán các từ được che giấu.

Xem xét câu trước đó: “Paris is a beautiful city”, và “I love Paris”. Đầu tiên, ta mã hóa các câu và nhận mã:

tokens = [Paris, is, a beautiful, city, I, love, Paris]

Bây giờ, ta thêm mã [CLS] ở đầu câu đầu tiên và mã [SEP] ở cuối mỗi câu:

tokens = [[CLS], Paris, is, a beautiful, city, [SEP], I, love, Paris, [SEP]]

Tiếp theo, ta ngẫu nhiên che giấu 15% mã (từ) trong danh sách mã trên. Giả sử, ta che giấu từ “city” và thay thế từ “city” bằng mã [MASK]:

tokens = [[CLS], Paris, is, a beautiful, [MASK], [SEP], I, love, Paris, [SEP]]

Bây giờ, ta đào tạo mô hình BERT để dự đoán mã [MASK]. Mã che giấu theo cách này sẽ tạo ra sự khác biệt giữa đào tạo trước và tinh chỉnh. Nghĩa là, ta đào tạo BERT bằng cách dự đoán mã [MASK]. Sau khi đào tạo, ta có thể tinh chỉnh mô hình BERT được đào tạo trước cho các nhiệm vụ phía sau, chẳng hạn như phân tích tình cảm. Nhưng trong quá trình tinh chỉnh, ta sẽ không có bất kỳ mã [MASK] nào trong đầu vào. Vì vậy, nó sẽ gây ra sự không phù hợp giữa cách mà BERT được đào tạo trước và cách sử dụng để tinh chỉnh.

Để khắc phục vấn đề này, ta áp dụng quy tắc 80-10-10%. Ta đã biết ngẫu nhiên che giấu 15% các mã trong câu. Bây giờ, đối với những mã 15% này, ta làm như sau:

- Trong 80% thời gian, ta thay thế mã (từ thực) bằng [MASK]. Vì vậy, trong 80% thời gian, đầu vào cho mô hình sẽ như sau:

tokens = [[CLS], Paris, is, a beautiful, [MASK], [SEP], I, love, Paris, [SEP]]

- Trong 10% thời gian, ta thay thế mã (từ thực) với mã ngẫu nhiên. Vì vậy, trong 10% thời gian, đầu vào cho mô hình sẽ như sau:

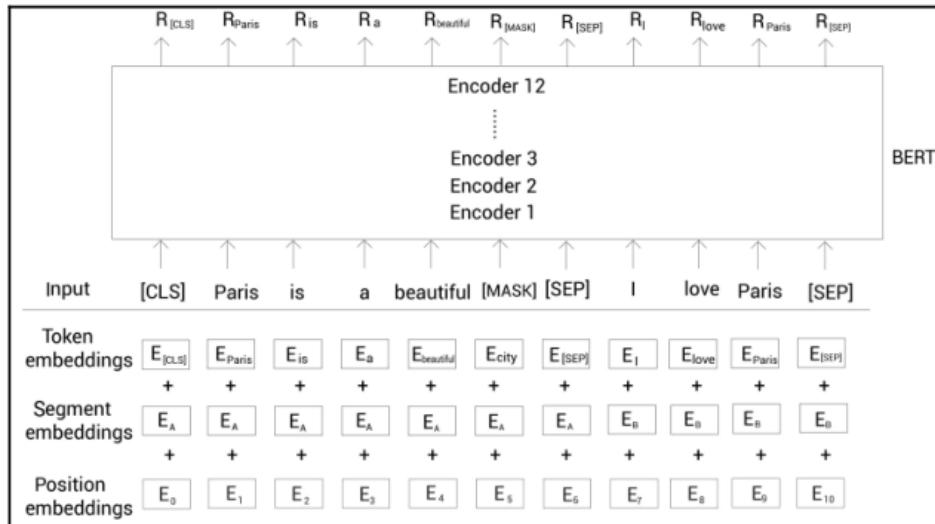
tokens = [[CLS], Paris, is, a beautiful, love, [SEP], I, love, Paris, [SEP]]

- Trong 10% thời gian, ta không thực hiện bất kỳ thay đổi nào. Vì vậy, trong 10% thời gian, đầu vào cho mô hình sẽ như sau:

tokens = [[CLS], Paris, is, a beautiful, city, [SEP], I, love, Paris, [SEP]]

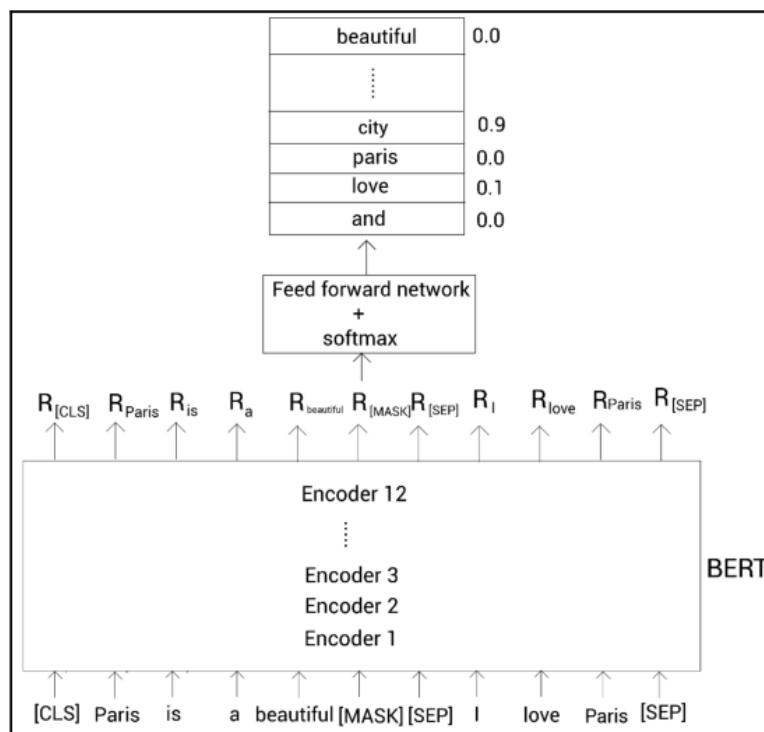
Sau đó, ta cấp các mã đầu vào cho các tầng mã, phân đoạn và vị trí và nhận các nhúng đầu vào.

Tiếp theo, ta cấp các nhung đđu vào cho BERT. Trong sơ đồ sau, BERT lấy đđu vào và trả về một biđu diđn của mđi mđ làm đđu ra. R_{CLS} biđu thị biđu diđn của [CLS], R_{Paris} biđu thị mđ Paris...



Hình 2.14. Mô hình ngôn ngữ che giấu.

Từ sơ đồ trên, ta đã thu được biđu diđn R của từng mđ. Để dự đoán [MASK], ta cấp biđu diđn của mđ R_{MASK} được trả về bởi BERT, cấp cho tầng mạng feedforward với hàm kích hoạt softmax và trả về xác suất của tất cả các từ trong từ vựng, để đoán từ bị che giấu:



Hình 2.15. Mô hình dự đoán mđ che giấu.

Trong sơ đồ trước, có xác suất cao từ “city” là từ bị che giấu. Vì vậy, trong trường hợp này, từ che giấu sẽ được dự đoán là “city”.

Trong các lần lặp lại ban đầu, mô hình sẽ không trả về xác suất chính xác vì trọng số của mạng feedforward và các tầng mã hóa của BERT sẽ không tối ưu. Tuy nhiên, trong một loạt các lần lặp, với phép lan truyền ngược, ta cập nhật các trọng số của mạng và các lớp mã hóa của BERT và học các trọng số tối ưu.

Dự đoán câu tiếp theo (NSP)

Dự đoán câu tiếp theo (tiếng Anh: Next sentence prediction, viết tắt: NSP) là một chiến lược khác được sử dụng để đào tạo mô hình BERT. NSP là một nhiệm vụ phân loại nhị phân. Trong nhiệm vụ NSP, ta cấp hai câu cho BERT và nó phải dự đoán liệu câu thứ hai có phải là phần tiếp theo (câu tiếp theo) của câu đầu tiên hay không.

Trong nhiệm vụ NSP, mục tiêu của mô hình là dự đoán liệu cặp câu thuộc danh mục isNext hoặc notNext. Ta cấp cặp câu (câu A và B) cho BERT và đào tạo nó để dự đoán liệu câu B có theo sau câu A không. Mô hình trả về isNext nếu câu B theo sau câu A, nếu không, nó sẽ trả về notNext dưới dạng đầu ra. Do đó, NSP về cơ bản là một nhiệm vụ phân loại nhị phân.

Bằng cách thực hiện nhiệm vụ NSP, mô hình có thể hiểu mối quan hệ giữa hai câu. Điều này hữu ích trong nhiều nhiệm vụ phía sau, chẳng hạn như trả lời câu hỏi và tạo văn bản.

Ta có thể tạo dữ liệu từ bất kỳ kho văn bản đơn ngữ nào. Giả sử, ta có một vài tài liệu. Đối với lớp isNext, ta lấy bất kỳ hai câu liên tiếp nào từ một tài liệu và gán nhãn cho chúng là isNext, và đối với lớp notNext, ta lấy một câu từ một tài liệu và một câu khác từ một tài liệu ngẫu nhiên và gán nhãn cho chúng là notNext.

2.3.4. Tinh chỉnh mô hình BERT cho bài toán phân loại văn bản

Giả sử ta đang thực hiện phân tích tình cảm. Trong nhiệm vụ phân tích tình cảm, mục tiêu là phân loại một câu là tích cực hay tiêu cực. Ví dụ, ta có một bộ dữ liệu chứa các câu cùng với nhãn của chúng.

Xem xét một câu: “I love Paris”. Đầu tiên, ta mã hóa câu, thêm mã [CLS] ở đầu và thêm mã [SEP] ở cuối câu. Sau đó, ta cung cấp các mã như một đầu vào cho mô hình BERT được đào tạo trước và nhận được các nhúng của tất cả các mã.

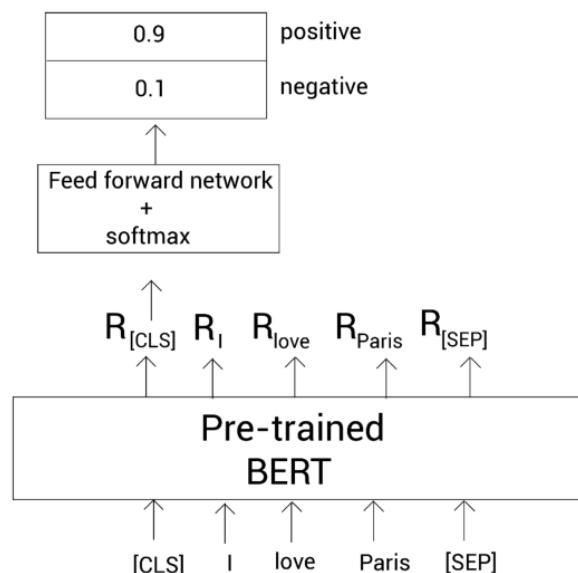
Tiếp theo, ta bỏ qua việc nhúng tất cả các mã khác và chỉ lấy mã thô nhúng $R_{[CLS]}$. Việc nhúng mã [CLS] sẽ giữ biểu diễn tổng hợp của câu. Ta cung cấp $R_{[CLS]}$ cho một bộ phân loại (mạng feedforward với hàm softmax) và đào tạo bộ phân loại để thực hiện phân tích tình cảm.

Khi tinh chỉnh mô hình BERT được đào tạo trước, ta cập nhật các trọng số của mô hình cùng với bộ phân loại. Nhưng khi ta sử dụng mô hình BERT được đào tạo trước làm trình trích xuất tính năng, ta chỉ cập nhật các trọng số của bộ phân loại chứ không phải mô hình BERT được đào tạo sẵn.

Trong quá trình tinh chỉnh, ta có thể tinh chỉnh các trọng số của mô hình theo hai cách sau:

- Cập nhật các trọng số của mô hình BERT được đào tạo trước cùng với tầng phân loại.
- Chỉ cập nhật các trọng số của tầng phân loại chứ không phải mô hình BERT được đào tạo trước. Khi ta làm điều này, nó giống như sử dụng mô hình BERT được đào tạo trước làm bộ trích xuất tính năng.

Hình sau đây cho thấy cách tinh chỉnh mô hình BERT được đào tạo trước cho một nhiệm vụ phân loại văn bản [8]:



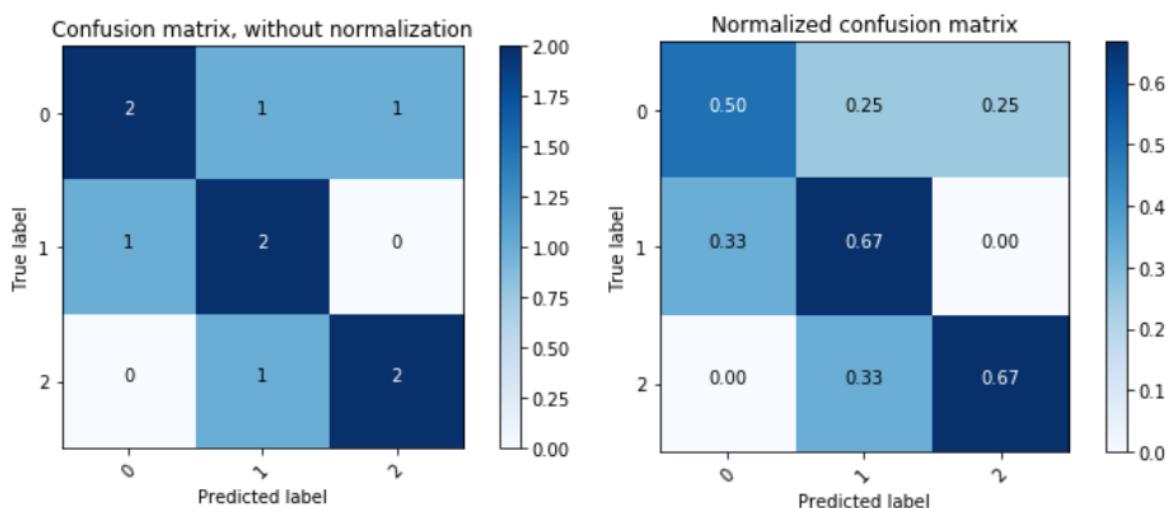
Hình 2.16. Tinh chỉnh mô hình BERT cho bài toán phân loại văn bản.

2.3.5. Thông số đánh giá mô hình phân loại

Khi xây dựng một mô hình, chúng ta cần một phép đánh giá để xem mô hình sử dụng có hiệu quả không và để so sánh khả năng của các mô hình. Các phương pháp thường được sử dụng là confusion matrix, accuracy, precision, recall và F1-score [9].

Confusion matrix

Về cơ bản, confusion matrix thể hiện có bao nhiêu mẫu dữ liệu thực sự thuộc vào một lớp, và bao nhiêu mẫu được dự đoán vào một lớp. Nó là một ma trận vuông với kích thước mỗi chiều bằng số lượng lớp dữ liệu. Giá trị tại hàng thứ i, cột thứ j là số lượng mẫu lẻ ra thuộc vào lớp i nhưng lại được dự đoán là thuộc vào lớp j. Tổng các phần tử trong ma trận này chính là số mẫu trong tập kiểm thử. Các phần tử trên đường chéo chính của ma trận là số mẫu được phân loại đúng của mỗi lớp dữ liệu. Cách biểu diễn ở trên còn được gọi là unnormalized confusion matrix, nghĩa là confusion matrix chưa chuẩn hoá. Để có cái nhìn rõ hơn, ta có thể dùng normalized confusion matrix, nghĩa là confusion matrix được chuẩn hoá. Để có normalized confusion matrix, ta lấy mỗi hàng của unnormalized confusion matrix sẽ được chia cho tổng các phần tử trên hàng đó. Như vậy, ta có nhận xét rằng, tổng các phần tử trên một hàng của normalized confusion matrix luôn bằng 1. Điều này thường không đúng trên mỗi cột.



Hình 2.17. Unnormalized confusion matrix và normalized confusion matrix.

Với các bài toán với nhiều lớp dữ liệu, cách biểu diễn bằng màu này rất hữu ích. Các ô màu đậm thể hiện các giá trị cao. Một mô hình tốt sẽ cho một confusion matrix có các phần tử trên đường chéo chính có giá trị lớn, các phần tử còn lại có giá trị nhỏ. Nghĩa là, khi biểu diễn bằng màu sắc, đường chéo có màu càng đậm so với phần còn lại sẽ càng tốt. Từ hai hình trên ta thấy rằng confusion matrix đã chuẩn hóa mang nhiều thông tin hơn. Sự khác nhau được thấy ở ô trên cùng bên trái. Lớp dữ liệu 0 được phân loại không thực sự tốt nhưng trong unnormalized confusion matrix, nó vẫn có màu đậm như hai ô còn lại trên đường chéo chính.

True/False Positive/Negative

Cách đánh giá này thường được áp dụng cho các bài toán phân lớp có hai lớp dữ liệu. Cụ thể hơn, trong hai lớp dữ liệu này có một lớp nghiêm trọng hơn lớp kia và cần được dự đoán chính xác. Ví dụ, trong bài toán xác định có bệnh ung thư hay không thì việc không bị sót quan trọng hơn là việc chẩn đoán nhầm âm tính thành dương tính. Trong bài toán xác định có mìn dưới lòng đất hay không thì việc bỏ sót nghiêm trọng hơn việc báo động nhầm. Trong bài toán lọc email rác thì việc cho nhầm email quan trọng vào thư rác nghiêm trọng hơn việc xác định một email rác là email thường. Trong những bài toán này, người ta thường định nghĩa lớp dữ liệu quan trọng hơn cần được xác định đúng là lớp Positive (P, dương tính), lớp còn lại được gọi là Negative (N, âm tính). Ta định nghĩa True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) dựa trên confusion matrix chưa chuẩn hóa trong hình sau:

		Predicted Class	
		True	False
True	True	True Positive (TP)	False Negative (FN)
	False	False Positive (FP)	True Negative (TN)

Hình 2.18. True Positive, False Positive, True Negative, False Negative.

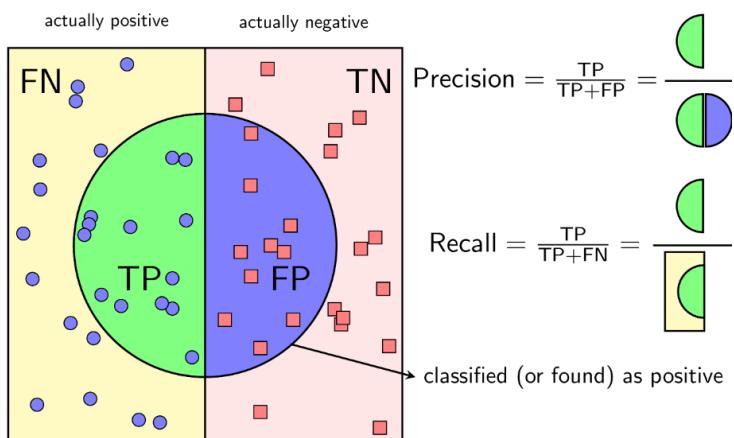
Accuracy

Cách đơn giản và hay được sử dụng nhất là accuracy (độ chính xác). Cách đánh giá này đơn giản tính tỉ lệ giữa số mẫu được dự đoán đúng và tổng số mẫu trong tập dữ liệu kiểm thử. Accuracy chỉ cho ta biết được bao nhiêu phần trăm lượng dữ liệu được phân loại đúng mà không chỉ ra được cụ thể mỗi loại được phân loại như thế nào, lớp nào được phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào thường bị phân loại nhầm vào lớp khác.

Precision và recall

Với bài toán phân loại mà tập dữ liệu của các lớp là chênh lệch nhau rất nhiều, có một phép đo hiệu quả thường được sử dụng là precision-recall. Giả sử ta phân loại nhị phân, ta cũng coi một trong hai lớp là positive, lớp còn lại là negative.

Với một cách xác định một lớp là positive, precision được định nghĩa là tỉ lệ số điểm true positive (TP) trong số những điểm được dự đoán là positive (TP + FP). Recall được định nghĩa là tỉ lệ số điểm true positive (TP) trong số những điểm thực sự là positive (TP + FN), như được thể hiện trong hình sau:



Hình 2.19. Cách tính precision và recall cho phân loại nhị phân.

Cả precision và recall đều là các số không âm nhỏ hơn hoặc bằng một. Một mô hình phân lớp tốt là mô hình có cả precision và recall đều cao, tức càng gần 1 càng tốt. Precision cao đồng nghĩa với việc độ chính xác của các mẫu tìm được là cao. Recall cao đồng nghĩa với việc tỉ lệ bỏ sót các mẫu thực sự là thấp.

$\text{Precision} = 1$ không đảm bảo mô hình là tốt. Nếu một mô hình chỉ tìm được đúng một mẫu mà nó chắc chắn nhất thì ta không thể gọi nó là một mô hình tốt. Nếu mô hình phân loại mọi mẫu là positive thì $\text{Recall} = 1$, tuy nhiên dễ nhận ra đây là một mô hình cực tồi.

F1-score

F1-score (hay F1 score) là giá trị trung bình hài hòa của precision và recall (giả sử rằng hai đại lượng này khác không). Do đó, nó thể hiện một cách đối xứng cả precision và recall trong một số liệu. Giá trị cao nhất có thể có của F1-score là 1, còn giá trị thấp nhất có thể là 0, nếu precision hoặc recall bằng 0. Công thức tính F1-score được hiển thị trong hình sau:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Hình 2.20. Công thức tính F1-score.

2.4. Thu thập dữ liệu

2.4.1. Giới thiệu về cào dữ liệu từ trang web

Cào (hoặc cạo) là quá trình trích xuất, sao chép, sàng lọc hoặc thu thập dữ liệu. Cào web cung cấp các công cụ và kỹ thuật được sử dụng để thu thập dữ liệu từ các trang web phù hợp với các nhu cầu liên quan đến cá nhân hoặc kinh doanh, nhưng với một số cân nhắc pháp lý.

Có một số yếu tố pháp lý cần xem xét trước khi thực hiện các nhiệm vụ cào. Hầu hết các trang web đều chứa các trang như chính sách bảo mật, về chúng tôi, và các điều khoản và điều kiện, trong đó các điều khoản pháp lý, chính sách nội dung bị cấm và thông tin chung có sẵn. Nhiệm vụ đạo đức của nhà phát triển là tuân theo các chính sách đó trước khi lên kế hoạch cho bất kỳ hoạt động thu thập dữ liệu nào từ các trang web [10].

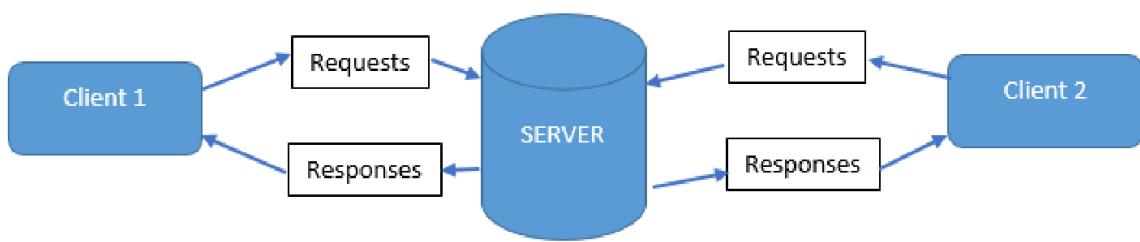
2.4.2. Tìm hiểu về công nghệ phát triển web

Ta (người dùng) sử dụng trình duyệt web (như Google Chrome, Firefox Mozilla, và Safari) để truy cập thông tin từ web. Các trình duyệt web cung cấp các chức năng dựa trên tài liệu khác nhau cho người dùng và chứa các tính năng cấp ứng dụng thường hữu ích cho các nhà phát triển web.

Một trang web là một tài liệu chứa các khối thẻ HTML (viết tắt của từ tiếng Anh: HyperText Markup Language, ngôn ngữ đánh dấu siêu văn bản). Hầu hết thời gian, nó được xây dựng với các khối phụ khác nhau được liên kết dưới dạng các thành phần phụ thuộc hoặc độc lập từ các công nghệ liên kết khác nhau, bao gồm JavaScript và CSS.

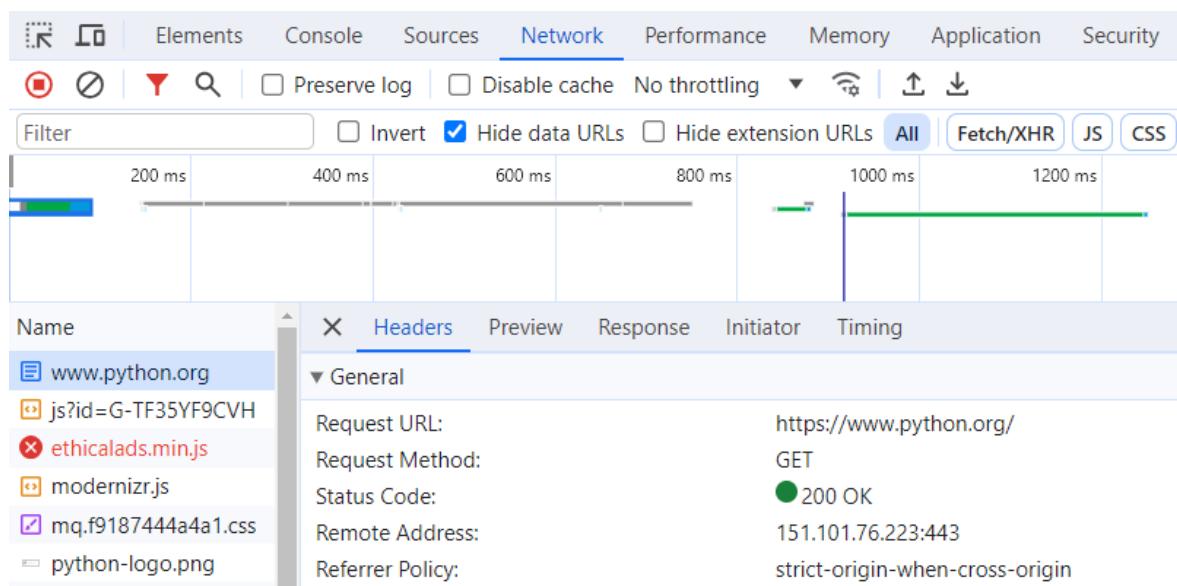
Giao thức HTTP

Giao thức truyền tải siêu văn bản (tiếng Anh: Hypertext Transfer Protocol, viết tắt: HTTP) là một giao thức ứng dụng truyền tải nguyên như tài liệu HTML giữa máy khách và máy chủ web. HTTP là một giao thức không trạng thái tuân theo mô hình máy khách-máy chủ. Máy khách (trình duyệt web) và máy chủ web truyền đạt hoặc trao đổi thông tin bằng cách sử dụng các yêu cầu HTTP và phản hồi HTTP:



Hình 2.21. Cách giao tiếp giữa máy chủ và máy khách.

Sử dụng các công cụ nhà phát triển trên trình duyệt để xem phương thức yêu cầu, cùng với thông tin khác liên quan đến HTTP:



Hình 2.22. Công cụ nhà phát triển trên trình duyệt Chrome.

Các tiêu đề HTTP truyền thông tin bổ sung cho máy khách hoặc máy chủ trong khi thực hiện yêu cầu hoặc phản hồi. Các tiêu đề thường là các cặp thông tin tên – giá trị được truyền giữa máy khách và máy chủ trong quá trình giao tiếp và thường được nhóm thành các tiêu đề yêu cầu và phản hồi:

- Tiêu đề yêu cầu: Các thông tin ngôn ngữ và yêu cầu mã hóa, cookie, thông tin liên quan đến trình duyệt... được cung cấp cho máy chủ trong khi thực hiện yêu cầu. Hình dưới đây minh họa tiêu đề yêu cầu tới <https://www.python.org>:

▼ Request Headers	
:authority:	www.python.org
:method:	GET
:path:	/
:scheme:	https
Accept:	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=1.0
Accept-Encoding:	gzip, deflate, br
Accept-Language:	en-US,en;q=0.9,vi;q=0.8
Cache-Control:	max-age=0
Dnt:	1
Sec-Ch-Ua:	"Google Chrome";v="117", "Not;A=Brand";v="8", "Chromium";v="117"
Sec-Ch-Ua-Mobile:	?0
Sec-Ch-Ua-Platform:	"Windows"
Sec-Fetch-Dest:	document
Sec-Fetch-Mode:	navigate
Sec-Fetch-Site:	none
Sec-Fetch-User:	?1
Upgrade-Insecure-Requests:	1
User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36

Hình 2.23. Tiêu đề yêu cầu.

- Tiêu đề phản hồi: Các thông tin liên quan đến phản hồi (bao gồm kích thước, loại và ngày) và trạng thái máy chủ. Hình dưới đây minh họa các tiêu đề phản hồi, sau khi đưa ra yêu cầu tới <https://www.python.org>:

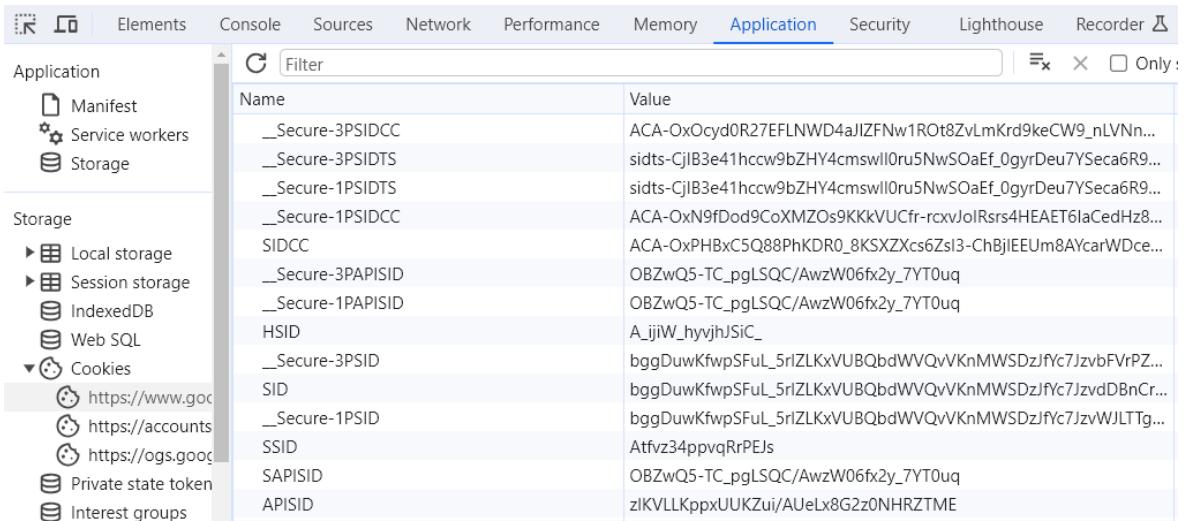
▼ Response Headers	
Accept-Ranges:	bytes
Age:	571
Content-Length:	50418
Content-Type:	text/html; charset=utf-8
Date:	Sat, 07 Oct 2023 14:13:20 GMT
Nel:	{"report_to": "heroku-nel", "max_age": 3600, "success_fraction": 0.005, "failure_fraction": 0.05, "response_headers": ["Via"]}
Report-To:	{"group": "heroku-nel", "max_age": 3600, "endpoints": [{"url": "https://nel.herokuapp.com/reports?ts=1696687381&sid=67ff5de4-ad2b-4112-9289-cf96be89efed&s=fIT0Zc"}]}
Reporting-Endpoints:	heroku-nel=https://nel.herokuapp.com/reports?ts=1696687381&sid=67ff5de4-ad2b-4112-9289-cf96be89efed&s=fIT0Zc
Server:	nginx
Strict-Transport-Security:	max-age=63072000; includeSubDomains; preload
Vary:	Cookie
Via:	1.1 vegur, 1.1 varnish, 1.1 varnish
X-Cache:	HIT, HIT
X-Cache-Hits:	8, 58
X-Frame-Options:	SAMEORIGIN
X-Served-By:	cache-iad-kiad7000025-IAD, cache-hkg17929-HKG
X-Timer:	S1696688000.213740,V\$0,VE1

Hình 2.24. Tiêu đề phản hồi.

Cookie HTTP là dữ liệu được gửi bởi máy chủ đến trình duyệt. Cookie là dữ liệu được tạo và lưu trữ bởi các trang web trên hệ thống hoặc máy tính của bạn. Dữ liệu trong cookie giúp xác định các yêu cầu HTTP từ người dùng đến trang

web, chứa thông tin liên quan đến quản lý phiên, sở thích của người dùng và hành vi của người dùng.

Máy chủ xác định và giao tiếp với trình duyệt dựa trên thông tin được lưu trữ trong cookie. Dữ liệu được lưu trữ trong cookie giúp một trang web truy cập và chuyển một số giá trị đã lưu nhất định như ID phiên, ngày và giờ hết hạn... cung cấp tương tác nhanh yêu cầu và phản hồi web:



The screenshot shows the Chrome DevTools interface with the 'Application' tab selected. On the left, the 'Storage' section is expanded, showing 'Cookies'. Under 'Cookies', there are entries for three domains: 'https://www.google.com', 'https://accounts.google.com', and 'https://ogs.google.com'. Each domain has several cookie items listed with their names and values. For example, under 'https://www.google.com', there are cookies like '_Secure-3PSIDCC' with value 'ACA-OxOcyd0R27EFLNWD4ajIZFNw1ROt8ZvLmKrd9keCW9_nLVNn...', '_Secure-3PSIDTS' with value 'sidts-CjIB3e41hccw9bZHY4cmswill0ru5NwSOaEf_0gyrDeu7YSeca6R9...', '_Secure-1PSIDTS' with value 'sidts-CjIB3e41hccw9bZHY4cmswill0ru5NwSOaEf_0gyrDeu7YSeca6R9...', '_Secure-1PSIDCC' with value 'ACA-OxN9fDod9CoXMZO9KKKVUCfr-rcxvJolRsrs4HEAET6laCedHz8...', 'SIDCC' with value 'ACA-OxPHBxC5Q88PhKDR0_2KSXZXcs6ZsI3-ChBjIEUm8AYcarWDce...', '_Secure-3PAPISID' with value 'OBZwQ5-TC_pgLSQC/AwzW06fx2y_7YT0uq', '_Secure-1PAPISID' with value 'OBZwQ5-TC_pgLSQC/AwzW06fx2y_7YT0uq', 'HSID' with value 'A_jjiW_hyyjhJSiC...', '_Secure-3PSID' with value 'bggDuvKfwpSFul_5rlZLKxVUBQbdWVQvVKnMWSDzJfYc7JzvbFVrPZ...', 'SID' with value 'bggDuvKfwpSFul_5rlZLKxVUBQbdWVQvVKnMWSDzJfYc7JzvdDbnCr...', '_Secure-1PSID' with value 'bggDuvKfwpSFul_5rlZLKxVUBQbdWVQvVKnMWSDzJfYc7JzvWJLTt...', 'SSID' with value 'Atfvz34ppvqRrPEjS...', 'SAPISID' with value 'OBZwQ5-TC_pgLSQC/AwzW06fx2y_7YT0uq', and 'APISID' with value 'zIKVLLKppxUUKZui/AUeLx8G2z0NHRZTME'. The table has columns for 'Name' and 'Value'.

Name	Value
_Secure-3PSIDCC	ACA-OxOcyd0R27EFLNWD4ajIZFNw1ROt8ZvLmKrd9keCW9_nLVNn...
_Secure-3PSIDTS	sidts-CjIB3e41hccw9bZHY4cmswill0ru5NwSOaEf_0gyrDeu7YSeca6R9...
_Secure-1PSIDTS	sidts-CjIB3e41hccw9bZHY4cmswill0ru5NwSOaEf_0gyrDeu7YSeca6R9...
_Secure-1PSIDCC	ACA-OxN9fDod9CoXMZO9KKKVUCfr-rcxvJolRsrs4HEAET6laCedHz8...
SIDCC	ACA-OxPHBxC5Q88PhKDR0_2KSXZXcs6ZsI3-ChBjIEUm8AYcarWDce...
_Secure-3PAPISID	OBZwQ5-TC_pgLSQC/AwzW06fx2y_7YT0uq
_Secure-1PAPISID	OBZwQ5-TC_pgLSQC/AwzW06fx2y_7YT0uq
HSID	A_jjiW_hyyjhJSiC...
_Secure-3PSID	bggDuvKfwpSFul_5rlZLKxVUBQbdWVQvVKnMWSDzJfYc7JzvbFVrPZ...
SID	bggDuvKfwpSFul_5rlZLKxVUBQbdWVQvVKnMWSDzJfYc7JzvdDbnCr...
_Secure-1PSID	bggDuvKfwpSFul_5rlZLKxVUBQbdWVQvVKnMWSDzJfYc7JzvWJLTt...
SSID	Atfvz34ppvqRrPEjS...
SAPISID	OBZwQ5-TC_pgLSQC/AwzW06fx2y_7YT0uq
APISID	zIKVLLKppxUUKZui/AUeLx8G2z0NHRZTME

Hình 2.25. Dữ liệu cookie.

Với các proxy HTTP, máy chủ proxy hoạt động như một máy chủ trung gian giữa máy khách và máy chủ web chính. Trình duyệt web gửi các yêu cầu đến máy chủ thực sự được chuyển qua proxy và proxy trả về phản hồi từ máy chủ cho máy khách.

Proxy thường được sử dụng để giám sát/lọc, cải tiến hiệu suất, dịch thuật và bảo mật cho các tài nguyên liên quan đến Internet. Proxy cũng có thể được mua dưới dạng dịch vụ. Ngoài ra còn có nhiều hình thức thực hiện proxy khác nhau, chẳng hạn như proxy web (có thể được sử dụng để bỏ qua chặn IP).

HTML

Các trang web được tạo thành từ các trang hoặc tài liệu chứa văn bản, hình ảnh, phong cách và kịch bản... Chúng thường được xây dựng với các ngôn ngữ đánh dấu như HTML và ngôn ngữ đánh dấu siêu văn bản mở rộng (tiếng Anh: Extensible Hypertext Markup Language, XHTML).

HTML định nghĩa và chứa nội dung của một trang web. Dữ liệu có thể được trích xuất từ trang HTML, trong các phần tử đánh dấu được gọi là thẻ. Thẻ HTML thường là một trình giữ chỗ được đặt tên mang các thuộc tính được xác định trước.

```
<div id="header" class="header">
    <h1>Welcome to My Website</h1>
</div>
<div id="menu" class="content">
    <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Contact</a></li>
    </ul>
</div>
```

Hình 2.26. Mã HTML.

Phần tử HTML và các thuộc tính

Các phần tử HTML (hoặc các nút tài liệu) là khái niệm xây dựng các tài liệu web. Các phần tử HTML được xây dựng với thẻ bắt đầu `<...>` và thẻ cuối `</...>`, với một số nội dung nhất định bên trong. Phần tử HTML cũng có thể chứa các thuộc tính, thường được xác định là tên thuộc tính = giá trị thuộc tính, cung cấp thông tin bổ sung cho phần tử. Các phần tử HTML cũng có thể được lồng trong cấu trúc dạng cây với hệ thống phân cấp cha con.

Các phần tử HTML có thể chứa một số thông tin bổ sung, chẳng hạn như các cặp khóa/giá trị. Chúng còn được gọi là thuộc tính phần tử HTML. Các thuộc tính giữ các giá trị và cung cấp nhận dạng, hoặc chứa thông tin bổ sung có thể hữu ích trong các hoạt động cào, như xác định các phần tử web chính xác và trích xuất các giá trị hoặc văn bản từ chúng, duyệt các phần tử...

Có một số thuộc tính nhất định phổ biến cho các phần tử HTML hoặc có thể được áp dụng cho tất cả các phần tử HTML như sau:

- id
- class
- style

Các thuộc tính của các phần tử HTML như id và class chủ yếu được sử dụng để xác định hoặc định dạng các phần tử riêng lẻ hoặc các nhóm phần tử. Các thuộc tính này cũng có thể được quản lý bởi CSS và các ngôn ngữ kịch bản khác.

Các giá trị thuộc tính id phải là duy nhất cho phần tử mà chúng được áp dụng. Các giá trị thuộc tính class chủ yếu được sử dụng với CSS, cung cấp các tùy chọn định dạng trạng thái như nhau và có thể được sử dụng với nhiều phần tử.

Các thuộc tính như ID và lớp được xác định bằng cách đặt “#” và “.” tương ứng trước tên thuộc tính khi được sử dụng với CSS, và phân tích cú pháp. Thủ HTML và các thuộc tính là một nguồn dữ liệu chính khi trích xuất.

JavaScript

JavaScript là ngôn ngữ lập trình được sử dụng lập trình các ứng dụng web chạy trong trình duyệt. JavaScript được ưa thích để thêm các tính năng động và cung cấp tương tác dựa trên người dùng bên trong các trang web. Tính khả dụng phía máy khách của JavaScript thường được sử dụng trong thử nghiệm và gỡ lỗi ứng dụng.

Mã JavaScript có thể được thêm vào HTML bằng cách sử dụng thẻ `<script>` hoặc được nhúng dưới dạng tệp. `<script>` chứa logic lập trình với các biến JavaScript, toán tử, hàm, mảng, vòng lặp, điều kiện và sự kiện, nhằm mục tiêu mô hình đối tượng tài liệu (tiếng Anh: Document Object Model, viết tắt: DOM):

```
<body>
  <h1>Hello, HTML and JavaScript!</h1>
  <button id="myButton">Click me!</button>

  <script>
    // JavaScript code
    document.getElementById("myButton").addEventListener("click", function () {
      alert("Button clicked!");
    });
  </script>

</body>
```

Hình 2.27. Nhúng mã JavaScript vào HTML.

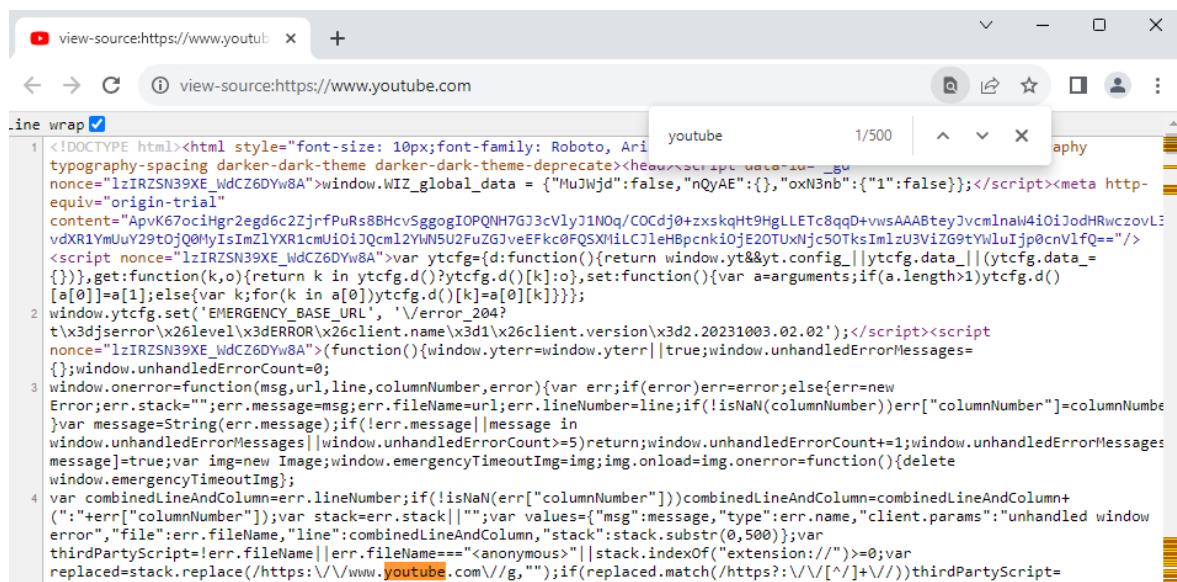
2.4.3. Kỹ thuật tìm dữ liệu cho web

Có nhiều công nghệ khác nhau có thể được sử dụng để phát triển một trang web. Nội dung được trình bày cho người dùng cuối bằng trình duyệt web cũng ở nhiều định dạng và mẫu khác nhau.

Nội dung trang cũng có thể bao gồm nội dung tĩnh được hiển thị với HTML và các công nghệ liên quan, hoặc được trình bày và tạo ra một cách nhanh chóng. Nội dung cũng có thể được truy xuất bằng cách sử dụng các nguồn của bên thứ ba và được trình bày cho người dùng cuối.

Nguồn trang HTML

Trình duyệt web được sử dụng để tương tác dựa trên máy khách – máy chủ lưu trữ nội dung web. Thanh địa chỉ trình duyệt được cung cấp với địa chỉ web hoặc URL (tiếng Anh: Uniform Resource Locator, định vị tài nguyên thống nhất) và URL được truyền đến máy chủ và nhận được phản hồi, nghĩa là được tải bởi trình duyệt. Phản hồi hoặc nguồn trang có thể được tìm kiếm ở định dạng thô.



```
<!DOCTYPE html><html style="font-size: 10px;font-family: Roboto, Arial, sans-serif"><head><script data-uid="1zIRZSN39XE_wdCZ6DYw8A">window.WIZ_global_data = {"MuJWjd":false,"nQyAE":{},"oxN3nb":{"1":false}};</script><meta http-equiv="origin-trial" content="ApvK67ociHgn2egd6c2ZjrfPuRs8BHcvSggogIOPQNH7GJ3cVlyJ1NOq/cOCdj0+zxsqHt9HgLLETc8qqD+vwsAAABteyJvcmlnaW4i0iJodHRwczovL3vdXR1YmUu29t0j00lyIsImZ1YXR1cmUi0i0j0cm12YMSU2fuZGJveFkccFQ5XHiLCJleHBpcnkioje20TUXNjc50TksImlzU3ViZG9tVnLuIjp0cnVlF0=="/><script nonce="1zIRZSN39XE_wdCZ6DYw8A">var ytcfg=(d:function(){return window.yt&&yt.config||ytcfg.data_||ytcfg.data_= {}}),get:function(k,o){return k in ytcfg.d()?ytcfg.d()[k]:o},set:function(){var a=arguments;if(a.length>1)ytcfg.d()[a[0]]=a[1];else{var k;for(k in a[0])ytcfg.d()[k]=a[0][k]}},getLineWrap:1</script><script nonce="1zIRZSN39XE_wdCZ6DYw8A">(function(){window.yterr=window.yterr||true;window.unhandledErrorMessages=[];window.onerror=function(e){var err=e.message||e.name||e.stack||e.lineNumber||e.columnNumber||e.fileName||e.url||e.lineNumber||e.columnNumber;var message=String(err.message);if(!err.message||message in window.unhandledErrorMessages||window.unhandledErrorCount>5) return;window.unhandledErrorCount+=1;window.unhandledErrorMessages[message]=true;var img=new Image;window.emergencyTimeoutImg=img;img.onload=img.onerror=function(){delete window.emergencyTimeoutImg};},combinedLineAndColumn=err.lineNumber;if(!isNaN(err["columnNumber"]))combinedLineAndColumn=combinedLineAndColumn+(":"+err["columnNumber"]);var stack=err.stack||"";var values={"msg":message,"type":err.name,"client.params":"unhandled window error","file":err.fileName,"line":combinedLineAndColumn,"stack":stack.substr(0,500)};var thirdPartyScript=!err.fileName||err.fileName==='<anonymous>'||stack.indexOf('extension://')>0;var replaced=stack.replace(/https?:\/\/www.youtube.com\/\//g,'');if(replaced.match(/https?:\/\/[^/]+\/\//))thirdPartyScript=
```

Hình 2.28. Nguồn trang YouTube.

Công cụ dành cho nhà phát triển

Các công cụ phát triển (hoặc DevTools) được nhúng trong hầu hết các trình duyệt phổ biến trên thị trường hiện nay. Các nhà phát triển và người dùng cuối đều có thể xác định vị trí tài nguyên và tìm kiếm nội dung web trong quá trình giao tiếp máy khách-máy chủ hoặc trong một yêu cầu và phản hồi HTTP.

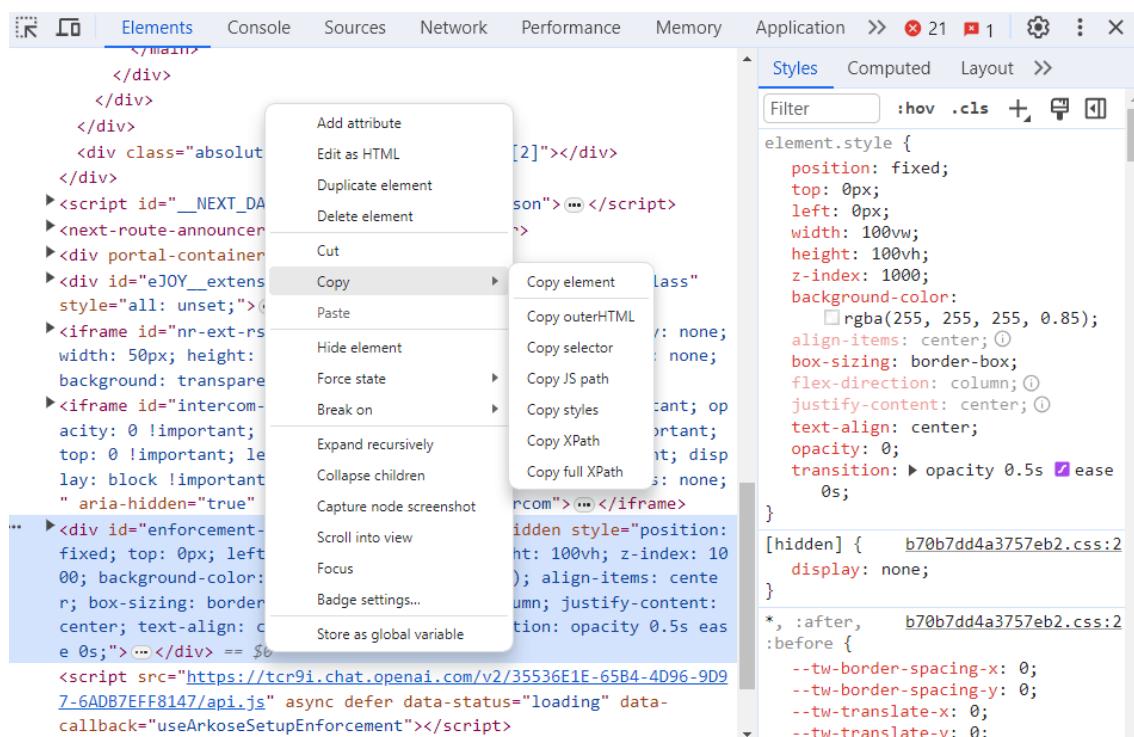
DevTools cho phép người dùng kiểm tra, tạo, chỉnh sửa và gỡ lỗi HTML, CSS và JavaScript. Chúng cũng cho phép ta xử lý các vấn đề về hiệu suất, tạo điều kiện cho việc trích xuất dữ liệu được trình bày bởi trình duyệt.

Trong Google Chrome, ta có thể mở DevTools bằng cách làm theo bất kỳ hướng dẫn nào sau đây:

- Nhấn phím F12 hoặc tổ hợp Ctrl + Shift + I.
- Nhấp chuột phải vào trang và nhấn tùy chọn “Kiểm tra” (Inspect).
- Thông qua menu Chrome, điều hướng đến “Công cụ khác” (More tools) và chọn “Công cụ dành cho Nhà phát triển” (Developer tools).

Có nhiều bảng khác nhau trong DevTools để phân tích, bao gồm các nguồn, bộ nhớ, hiệu suất và mạng. Một số bảng thường được dùng trong Chrome DevTools:

- Element: Hiển thị nội dung HTML của trang được xem. Điều này được sử dụng để xem và chỉnh sửa DOM và CSS, hoặc tìm cả CSS selector và XPath.
- Console: Được sử dụng để chạy và tương tác với mã JavaScript và xem thông báo nhật ký.

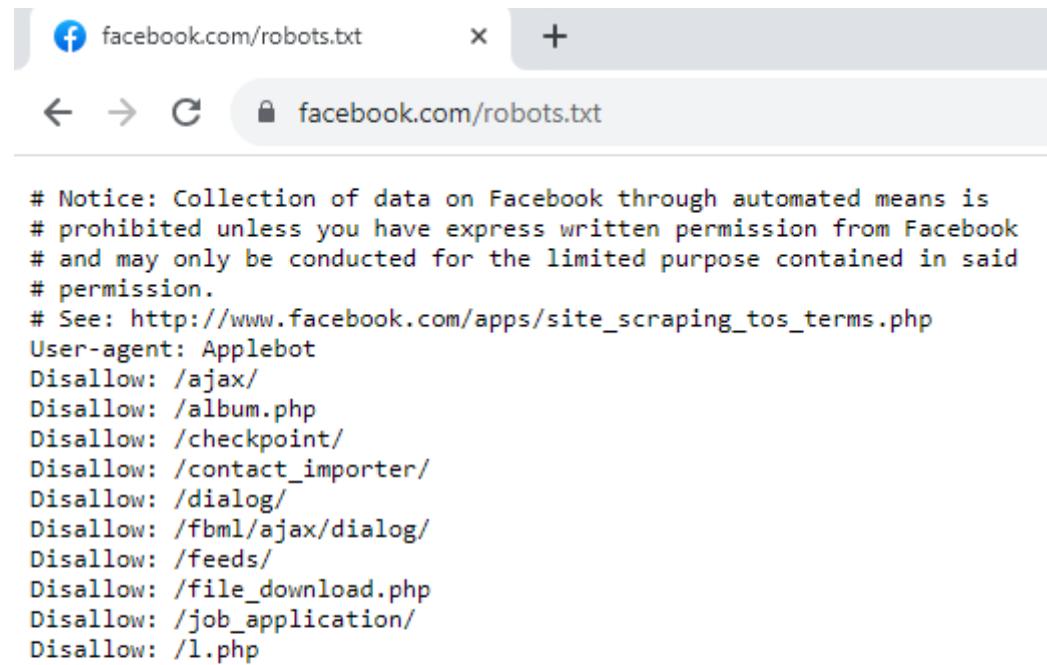


Hình 2.29. Kỹ thuật lấy phần tử.

Tệp robots.txt

Robots.txt, còn được gọi là giao thức loại trừ robot, là một tiêu chuẩn dựa trên web được sử dụng bởi các trang web để trao đổi thông tin với các tập lệnh tự động. Nói chung, robots.txt mang các hướng dẫn liên quan đến URL, trang và thư

mục trên trang web của họ đến các robot web (còn được gọi là wanderers, crawlers hoặc spiders) để điều khiển hành vi:



Hình 2.30. Tệp robots.txt trang web Facebook.

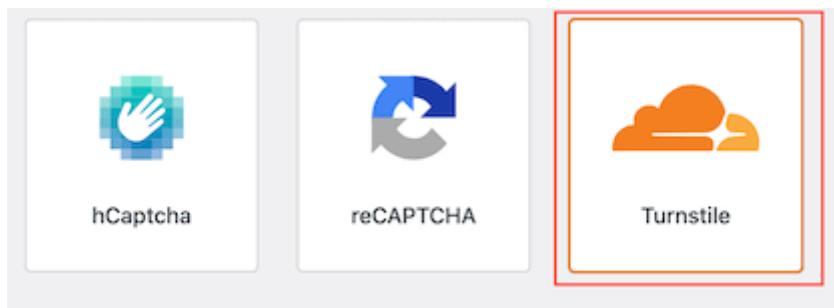
Đối với bất kỳ URL, tệp robots.txt có thể được truy cập bằng cách thêm robots.txt vào cuối URL. Ví dụ, <https://facebook.com/robots.txt>.

Trình thu thập dữ liệu web nên tuân theo các chỉ thị được đề cập trong tệp, nhưng với mục đích trích xuất dữ liệu thông thường, không có hạn chế nào được áp dụng cho đến khi và trừ khi tập lệnh thu thập thông tin cần trả lưu lượng truy cập trang web. Tệp robots.txt là không bắt buộc trên mỗi trang web.

2.4.4. Làm việc với bảo mật web

Bảo mật web

Việc triển khai các tính năng bảo mật dựa trên web (hoặc các tính năng duy trì trạng thái truy cập an toàn) để truy cập thông tin đang phát triển nhanh chóng, từng ngày. Bảo vệ nội dung web thường là thách thức từ góc độ thu thập thông tin và cào. Các trình thu thập thông tin web thông thường không thể tìm được một trang web bằng cách bị chặn qua CAPTCHA (viết tắt của từ tiếng Anh: "Completely Automated Public Turing test to tell Computers and Humans Apart", Phép thử Turing công cộng hoàn toàn tự động để phân biệt máy tính với người) hoặc một số phương thức khác.



Hình 2.31. Một số CAPTCHA.

Xoay proxy IP và tác nhân người dùng

Một trong những cách rõ ràng nhất mà một trang web có thể phát hiện người dùng là bằng cách theo dõi địa chỉ IP (viết tắt của từ tiếng Anh: Internet Protocol, giao thức Internet) được sử dụng trong khi thực hiện các yêu cầu đến trang. Nếu phát hiện hàng trăm yêu cầu từ một địa chỉ IP duy nhất trong một khoảng thời gian ngắn, thì có thể nói một cách đáng tin cậy rằng, nó là một máy cào tự động thực hiện yêu cầu thay vì người dùng thực sự. Trong trường hợp đó, một ý tưởng tốt để chặn hoặc giảm bớt các yêu cầu bằng cách dùng một lỗi máy chủ 501/502 để có thể hạn chế tài nguyên máy chủ.

Nếu thực tế bạn đang chạy một bộ cào, thì bạn có thể gửi các yêu cầu bằng cách sử dụng một nhóm địa chỉ IP proxy với thời gian độ trễ theo cách mà bạn chỉ vào một miền đích một vài lần một phút từ một địa chỉ IP. Bằng chiến lược này, bạn làm cho các hoạt động cào của bạn rất khó nhận ra và nó sẽ ngăn bạn khỏi bị chặn. Nếu bạn tiếp tục sử dụng cùng một bộ địa chỉ IP để thực hiện các yêu cầu, thì cuối cùng bạn sẽ bị chặn và bạn sẽ phải xoay địa chỉ IP sang các địa chỉ mới.

Cloudflare

Cloudflare là một mạng phân phối nội dung phổ biến được sử dụng bởi hơn 20% trong số trang web. Dự án miễn phí của họ bao gồm bảo vệ chống lại các cuộc tấn công từ chối dịch vụ phân tán (tiếng Anh: distributed denial-of-service attacks, viết tắt: DDoS). Đây là lý do tại sao ta nên có thời gian chờ hợp lý giữa các yêu cầu, nhưng đôi khi, điều đó không đủ để ngăn chặn bị gắn cờ như một cuộc tấn công DDoS tiềm năng của Cloudflare.

Một trang web bị tấn công của Cloudflare là biện pháp chống bot phía máy khách và nó kiểm tra xem JavaScript có được bật hay không và đưa ra một thử

thách. Điều này dễ dàng để vượt qua nếu bạn đang sử dụng trình duyệt thực; nhưng thông thường hơn, ta đang yêu cầu HTML trực tiếp và điều đó sẽ bị bắt bởi trang này, dẫn đến CAPTCHA, hoặc tệ hơn nữa, đặt địa chỉ IP vào danh sách đen.

2.5. Môi trường phát triển

2.5.1. Ngôn ngữ lập trình python

Python là một ngôn ngữ lập trình bậc cao, đa năng, được thiết kế để dễ đọc, dễ học và dễ sử dụng. Nó được phát triển bởi Guido van Rossum vào năm 1991 và đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

Dưới đây là một số đặc điểm cơ bản về Python:

- Mã nguồn mở: Python là ngôn ngữ mã nguồn mở, nghĩa là nó miễn phí và có thể được sử dụng bởi bất kỳ ai.
- Dễ hiểu: Cú pháp của Python sử dụng khoảng trắng (thụt lề) để định nghĩa các khối mã, không sử dụng dấu ngoặc như nhiều ngôn ngữ lập trình khác.
- Đa năng: Python có thể được sử dụng cho nhiều mục đích khác nhau, bao gồm phát triển ứng dụng web, phân tích dữ liệu, trí tuệ nhân tạo, học máy, tự động hóa công việc, và nhiều ứng dụng khác.
- Thư viện và khung lập trình: Python có một hệ sinh thái mạnh mẽ của thư viện và khung lập trình. Django và Flask cho phát triển web, NumPy và Pandas cho xử lý dữ liệu, TensorFlow và PyTorch cho học máy và trí tuệ nhân tạo, và nhiều thư viện khác.
- Cộng đồng lớn: Python có một cộng đồng rất lớn và đa dạng, với hàng triệu lập trình viên trên khắp thế giới. Nghĩa là có rất nhiều tài liệu, và cộng đồng để hỗ trợ người mới học và những người đã có kinh nghiệm.
- Sự phát triển liên tục: Python liên tục phát triển, với việc ra mắt các phiên bản mới với nâng cấp và cải tiến định kỳ. Python 3.x là phiên bản được khuyến nghị cho mọi dự án mới, trong khi Python 2.x đã bị ngừng hỗ trợ.

2.5.2. Các thư viện hỗ trợ

Thư viện selenium

Selenium là một thư viện mã nguồn mở được sử dụng rộng rãi để tự động hóa các thao tác trình duyệt web. Nó cho phép tạo các kịch bản tự động để tương tác với trang web giống như một người dùng thực sự thao tác trình duyệt.

Dưới đây là một số đặc điểm về selenium:

- Selenium WebDriver: Một phần quan trọng của Selenium. Nó cho phép tương tác với các trình duyệt web và thực hiện các hoạt động như điều hướng, điền dữ liệu, và kiểm tra nội dung trên trang web.
- Đa nền tảng: Selenium có thể chạy trên nhiều hệ điều hành khác nhau như Windows, macOS và Linux.
- Kiểm thử tự động chức năng và giao diện: Ta có thể kiểm tra xem các chức năng có hoạt động như mong đợi và giao diện có đúng với thiết kế hay không.
- Hỗ trợ nhiều trình duyệt: Selenium hỗ trợ một loạt trình duyệt phổ biến và cho phép chạy cùng một kịch bản kiểm thử trên nhiều trình duyệt khác nhau.
- Sử dụng trong kiểm thử liên tục: Selenium có thể tích hợp vào các quy trình kiểm thử liên tục để đảm bảo rằng các thay đổi trong mã nguồn không làm hỏng các chức năng của ứng dụng web.

Thư viện re

Thư viện re cung cấp các hoạt động khớp biểu thức chính quy (tiếng Anh: Regular Expression, viết tắt là regex hoặc regexp). Cả mẫu và chuỗi cần tìm kiếm đều có thể là chuỗi Unicode (str) cũng như chuỗi 8 bit (byte). Tuy nhiên, không thể trộn lẫn chuỗi Unicode và chuỗi 8 bit. Nghĩa là ta không thể khớp chuỗi Unicode với mẫu byte hoặc ngược lại; tương tự, khi yêu cầu thay thế, chuỗi thay thế phải cùng loại với cả mẫu và chuỗi tìm kiếm.

Dưới đây là một số ứng dụng của biểu thức chính quy:

- Tìm kiếm và thay thế: cho phép ta tìm kiếm các chuỗi ký tự phù hợp với một mẫu và thay thế chúng bằng một chuỗi khác. Điều này rất hữu ích trong việc chỉnh sửa văn bản hoặc dự án lớn.

- Phân tích dữ liệu: phân tích và trích xuất thông tin từ dữ liệu không cấu trúc. Ví dụ, ta có thể tìm kiếm và trích xuất địa chỉ email, số điện thoại, liên kết web, hoặc các định dạng dữ liệu khác từ một văn bản.
- Xử lý dữ liệu đầu vào: kiểm tra dữ liệu đầu vào của người dùng trong các ứng dụng web hoặc ứng dụng định dạng dữ liệu, đảm bảo rằng dữ liệu được nhập theo đúng định dạng mong muốn.
- Tối ưu hóa dữ liệu: tìm kiếm và thay thế các mẫu phức tạp, giúp ta tối ưu hóa dữ liệu hoặc sửa lỗi dữ liệu.

Thư viện numpy

NumPy là gói cơ bản dành cho tính toán khoa học bằng Python. Nó là một thư viện Python cung cấp một đối tượng mảng đa chiều và một loạt các thao tác nhanh trên mảng, bao gồm toán học, logic, thao tác hình dạng, sắp xếp, chọn, đại số tuyến tính, các phép toán thống kê.

Dưới đây là một số tính năng của numpy:

- Tốc độ xử lý: NumPy xử lý mảng đa chiều có hiệu suất cao, giúp thực hiện các phép toán số học nhanh chóng và dễ dàng trên dữ liệu đa chiều, như vectơ, ma trận và tensor.
- Phân tích dữ liệu: NumPy thường được sử dụng trong phân tích dữ liệu và khoa học dữ liệu để thực hiện các phép toán thống kê, tính toán trung bình, độ lệch chuẩn, và phương sai trên dữ liệu.
- Thực hiện nhân tích chập: NumPy là một công cụ mạnh mẽ để thực hiện tích chập trong mạng nơ-ron.
- Tính toán đại số tuyến tính: NumPy cung cấp các hàm để giải các vấn đề đại số tuyến tính như giải phương trình tuyến tính và tính toán giá trị riêng, ma trận nghịch đảo và nhiều phép toán khác.
- Tính toán đại số đạo hàm: NumPy có thể được sử dụng để tính đạo hàm và tích phân, đặc biệt trong các vấn đề tối ưu hóa và học máy.

Thư viện pandas

Pandas là một thư viện mã nguồn mở theo giấy phép BSD cung cấp các công cụ phân tích dữ liệu và cấu trúc dữ liệu hiệu suất cao, dễ sử dụng cho ngôn ngữ lập trình Python.

Dưới đây là một số ứng dụng của pandas:

- Đọc và ghi dữ liệu: Pandas cho phép đọc dữ liệu từ nhiều nguồn khác nhau như tệp CSV, Excel, SQL và nhiều định dạng dữ liệu khác. Nó cũng cho phép ghi dữ liệu vào các định dạng này.
- Khám phá và tiền xử lý dữ liệu: xem dữ liệu, kiểm tra giá trị trống, loại bỏ trùng lặp, và các thao tác khác để chuẩn bị dữ liệu cho việc phân tích hoặc học máy.
- Lọc và truy vấn dữ liệu: lọc dữ liệu dựa trên các điều kiện, thực hiện truy vấn và tổng hợp dữ liệu theo nhiều cách khác nhau.
- Thao tác với dữ liệu đa chiều: hỗ trợ các phép biến đổi dữ liệu đa chiều để cấu trúc lại dữ liệu cho mục đích phân tích cụ thể.
- Thống kê và tính toán: tính toán các thống kê cơ bản như trung bình, trung vị, mode và các phép toán thống kê phức tạp hơn trên dữ liệu.
- Kết hợp dữ liệu: kết hợp dữ liệu từ nhiều nguồn thông qua các phép toán như merge, join và concat.

Thư viện matplotlib

Matplotlib là một thư viện tạo ra các biểu đồ và đồ thị đa dạng. Nó được sử dụng rộng rãi trong lĩnh vực khoa học dữ liệu, phân tích số liệu, và trực quan hóa dữ liệu.

Dưới đây là một số đặc điểm và ứng dụng của matplotlib:

- Tạo biểu đồ: biểu diễn dữ liệu số học bằng cách tạo các biểu đồ dựa trên dữ liệu, ví dụ như biểu đồ đường, biểu đồ cột, biểu đồ phân phối, biểu đồ điểm, và nhiều loại biểu đồ khác.
- Trực quan hóa khoa học dữ liệu: biểu đồ dùng để biểu thị dữ liệu đo lường từ các thí nghiệm khoa học, dữ liệu về tài chính, dữ liệu địa lý, và nhiều loại dữ liệu khác.

- Hỗ trợ nhiều định dạng đầu ra: lưu biểu đồ được tạo ra bằng matplotlib dưới nhiều định dạng khác nhau.
- Biểu đồ tùy chỉnh: cho phép tùy chỉnh hầu hết các khía cạnh của biểu đồ, bao gồm tiêu đề, nhãn, màu sắc, kích thước, chú thích, vùng đánh dấu, và nhiều thuộc tính khác.

Thư viện seaborn

Seaborn là thư viện trực quan hóa dữ liệu dựa trên matplotlib. Nó cung cấp một giao diện cấp cao để vẽ đồ họa thống kê hấp dẫn và giàu thông tin.

Dưới đây là một số đặc điểm của seaborn:

- Dễ sử dụng: tạo ra các biểu đồ phổ biến như biểu đồ đường, biểu đồ điểm, biểu đồ hộp, và heatmap một cách dễ dàng và nhanh chóng.
- Tùy chỉnh dễ dàng: có thể tùy chỉnh nhiều khía cạnh của biểu đồ, bao gồm màu sắc, kiểu dáng, tiêu đề và chú thích để làm cho biểu đồ phù hợp với nhu cầu.
- Hỗ trợ đa biểu đồ: hỗ trợ việc tạo ra các biểu đồ kết hợp như Pair Plots, Joint Plots và Facet Grids để hiển thị mối tương quan giữa các biến trong dữ liệu.

Thư viện scikit-learn

Scikit learn (hoặc sklearn) bao gồm các công cụ đơn giản và hiệu quả để phân tích dự đoán dữ liệu. Mọi người đều có thể truy cập và có thể sử dụng lại trong nhiều bối cảnh khác nhau. Nó được xây dựng trên NumPy, SciPy và matplotlib, và có thể sử dụng về mặt thương mại - giấy phép BSD.

Dưới đây là một số đặc điểm của scikit-learn:

- Hỗ trợ cho nhiều loại mô hình: cung cấp một loạt các thuật toán học máy và khám phá dữ liệu, bao gồm hồi quy tuyến tính, phân loại, gom cụm, giảm chiều dữ liệu và nhiều thuật toán khác.
- Hỗ trợ cho việc tiền xử lý dữ liệu: cung cấp nhiều công cụ để tiền xử lý dữ liệu như chuẩn hóa, mã hóa biến phân loại, xử lý dữ liệu thiếu, và trích xuất đặc trưng.

- Hiệu suất tốt: Scikit-learn được xây dựng trên thư viện NumPy và SciPy, giúp tối ưu hóa hiệu suất và khả năng mở rộng.
- Cộng đồng lớn và tài liệu phong phú: Scikit-learn có một cộng đồng sử dụng rộng lớn và nhiều tài liệu giúp người dùng học và làm việc với nó.

Thư viện pytorch

PyTorch một thư viện mã nguồn mở và mạnh mẽ được phát triển bởi Facebook, để hỗ trợ việc xây dựng và đào tạo các mạng nơ-ron.

Dưới đây là một số tính năng của pytorch:

- Hỗ trợ bộ xử lý đồ họa (GPU): được tối ưu hóa để hoạt động trên GPU, giúp tăng tốc quá trình đào tạo mô hình. Nó cung cấp các phép toán được tối ưu hóa để sử dụng GPU hiệu quả và hỗ trợ nhiều kiến trúc GPU khác nhau.
- Cộng đồng lớn và phong phú: có một cộng đồng người dùng và phát triển đông đảo, với nhiều tài liệu và thư viện bổ sung cho việc xây dựng các ứng dụng học máy và trí tuệ nhân tạo.
- Mạnh mẽ trong việc xây dựng mô hình mạng và học sâu: PyTorch cung cấp các công cụ mạnh mẽ để xây dựng và tinh chỉnh các mạng phức tạp. Ta có thể dễ dàng tạo ra các mô hình mạng tùy chỉnh và kiểm soát chúng một cách linh hoạt.

2.5.3. Các phần mềm và công cụ

Phần mềm visual studio code

Visual Studio Code là trình soạn thảo mã phổ biến được phát triển bởi Microsoft. Nó nhẹ nhưng mạnh mẽ cho cả Windows, macOS và Linux. Nó đi kèm với hỗ trợ tích hợp cho JavaScript, TypeScript và Node.js, đồng thời có kho phần mở rộng phong phú cho các ngôn ngữ khác, chẳng hạn như C++, C#, Java, Python.

Dưới đây là các tính năng nổi bật về Visual Studio Code:

- Hoạt động như một môi trường phát triển tích hợp: Cung cấp hoàn thành mã thông minh, đề xuất và định dạng mã tự động dựa trên ngữ cảnh cụ thể của ngôn ngữ.

- **Khả năng tùy chỉnh:** Tùy biến cao để phù hợp với sở thích và nhu cầu của từng cá nhân. Nó cho phép người dùng thay đổi chủ đề, tổ hợp phím và cài đặt. Ngoài ra, các nhà phát triển có thể tạo các tiện ích mở rộng hoặc cài đặt các tiện ích mở rộng được phát triển để thêm các chức năng.
- **Tích hợp ứng dụng đầu cuối:** Tính năng này cho phép chạy các lệnh, xây dựng dự án và thực hiện các tác vụ khác nhau mà không cần chuyển sang một ứng dụng đầu cuối riêng biệt.
- **Gỡ lỗi:** Cung cấp trải nghiệm gỡ lỗi mạnh mẽ với sự hỗ trợ cho các ngôn ngữ khác nhau. Nó cho phép thiết lập các điểm ngắt, duyệt qua mã, kiểm tra các biến và xử lý các ngoại lệ, giúp xác định và khắc phục sự cố trong mã dễ dàng hơn.
- **Kiểm soát phiên bản:** Tích hợp git cho phép quản lý mã nguồn trực tiếp. Nó cung cấp các tính năng như quản lý nhánh và giải quyết xung đột, giúp làm việc thuận tiện với các hệ thống kiểm soát phiên bản.
- **Chia sẻ trực tiếp:** Chia sẻ phiên viết mã với những người khác, cho phép cộng tác theo thời gian thực, đánh giá mã.
- **Cộng đồng và tài liệu:** Có một cộng đồng lớn và tích cực các nhà phát triển. Tài liệu chính thức cung cấp thông tin chi tiết về các tính năng khác nhau, tùy chọn cấu hình và phát triển tiện ích mở rộng, cùng với các hướng dẫn để giúp người dùng tận dụng tối đa trình chỉnh sửa.

Phần mềm gán nhãn Label Studio

Label Studio là một công cụ ghi nhãn và ghi chú dữ liệu mã nguồn mở, được sử dụng để tạo dữ liệu được gán nhãn cho học máy. Nó cung cấp một giao diện thân thiện với người dùng để gán nhãn nhiều loại dữ liệu khác nhau, bao gồm văn bản, hình ảnh, âm thanh.

Dưới đây là một số đặc điểm về Label Studio:

- **Hỗ trợ đa phương thức:** linh hoạt cho các tác vụ như phát hiện đối tượng, phân loại hình ảnh, phân loại văn bản, phân tích tình cảm...
- **Giao diện tùy chỉnh:** thiết kế giao diện chủ thích riêng, điều chỉnh nó theo nhu cầu cụ thể của nhiệm vụ ghi nhãn.

- Hợp tác: Nhiều chủ thích có thể hợp tác trong các tác vụ gán nhãn và theo dõi tiến trình của họ và đảm bảo tính nhất quán trong ghi nhãn.
- Cộng đồng tích cực: Label Studio có một cộng đồng người dùng và người đóng góp tích cực.

Công cụ Google Colab

Google Colaboratory (hoặc Colab), là một sản phẩm của Google Research. Colab cho phép mọi người viết và thực thi mã python thông qua trình duyệt và đặc biệt phù hợp với học máy, phân tích dữ liệu và giáo dục. Về kỹ thuật, Colab là một dịch vụ lưu trữ Jupyter Notebook mà không yêu cầu thiết lập để sử dụng, đồng thời cung cấp quyền truy cập miễn phí vào các tài nguyên máy tính.

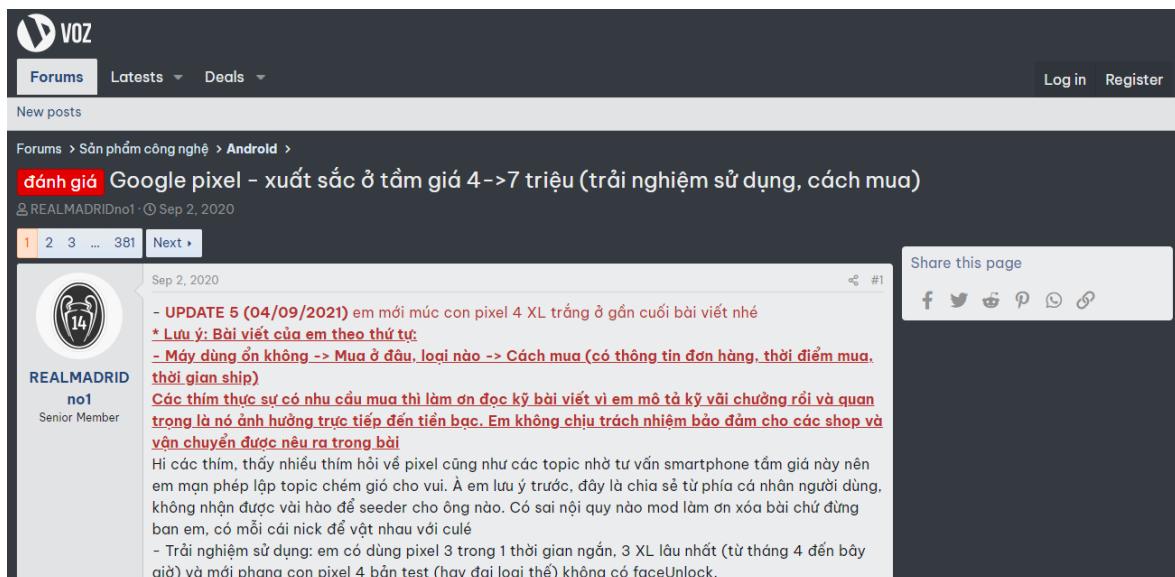
Sau đây là các tính năng chính về Google Colab:

- Môi trường dựa trên đám mây: Google Colab chạy hoàn toàn trên đám mây, nghĩa là bạn không cần phải lo lắng về việc cài đặt hoặc định cấu hình. Các thư viện cần thiết đều được cài đặt sẵn, giúp thiết lập dễ dàng.
- Tích hợp Jupyter Notebook: Colab được xây dựng trên Jupyter Notebook, một dự án mã nguồn mở cho phép người dùng tạo mã trực tiếp, hình ảnh trực quan. Giao diện này giúp ta dễ dàng viết và thực thi mã python từng bước.
- Tài nguyên phần cứng: Colab cung cấp quyền truy cập vào các tài nguyên phần cứng mạnh mẽ, giúp tăng tốc đáng kể các tác vụ tính toán chuyên sâu, chẳng hạn như đào tạo các mô hình học sâu, bằng cách tính toán song song.
- Quản lý tệp: Colab cho phép tải lên và tải xuống các tệp trực tiếp từ máy cục bộ, thuận tiện cho việc làm việc với bộ dữ liệu hoặc lưu kết quả đầu ra của mô hình. Nó cũng cung cấp quyền truy cập vào Google Drive, cho phép tích hợp liền mạch với bộ nhớ đám mây của Google.
- Đoạn mã và ví dụ: Colab cung cấp kho lưu trữ các đoạn mã và ví dụ về nhiều chủ đề khác nhau. Những đoạn mã này có thể hữu ích cho việc học và thử nghiệm các kỹ thuật khác nhau hoặc hiểu các chức năng cụ thể của thư viện.

CHƯƠNG 3. PHÂN LOẠI BÌNH LUẬN VỚI HỌC SÂU

3.1. Phát biểu bài toán

Trong nội dung của đồ án này, tôi tập trung nghiên cứu, tìm hiểu và ứng dụng các cơ sở lý thuyết vào bài toán phân loại các bình luận tiếng Việt. Tập dữ liệu thực tế là các đánh giá, nhận xét và thảo luận liên quan đến điện thoại thông minh Google Pixel trên diễn đàn công nghệ Việt Nam VOZ. Tên chủ đề là “Google pixel - xuất sắc ở tầm giá 4->7 triệu (trải nghiệm sử dụng, cách mua)” [11].



Hình 3.1. Giao diện một chủ đề của diễn đàn VOZ.

Mục tiêu của bài toán bao gồm đưa ra các nhận xét, phân tích về diễn đàn VOZ, cũng như các nhận xét, phân tích về điện thoại Google Pixel. Trên phương diện người dùng, giúp đưa ra quyết định có nên mua điện thoại hay không, các yếu tố để chọn điện thoại phù hợp với nhu cầu. Trên phương diện nhà sản xuất thiết bị, giúp cải tiến nâng cao chất lượng sản phẩm phù hợp hơn với nhu cầu thị trường.

Để đạt được các mục tiêu đã liệt kê ở trên, bài toán được chia ra thành các bài toán nhỏ như sau:

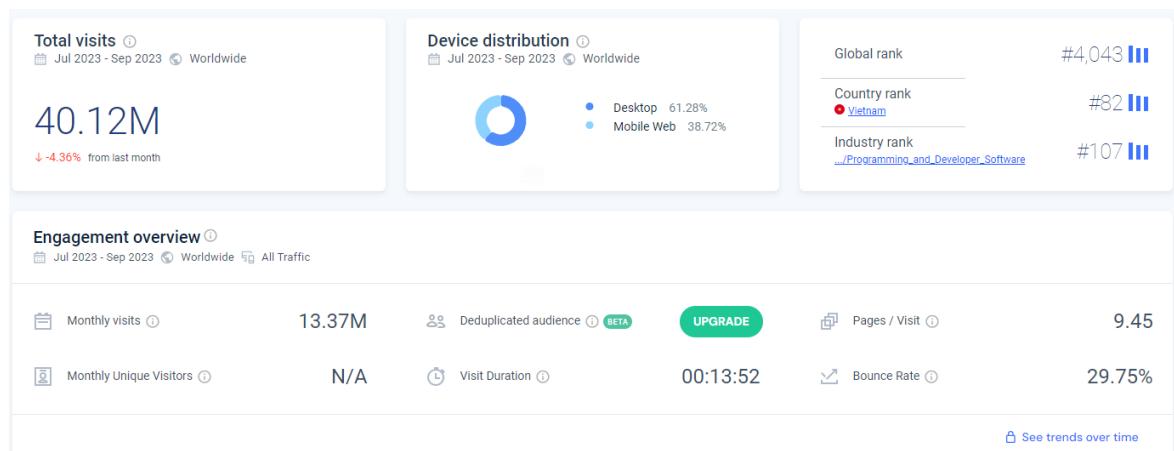
1. Thu thập dữ liệu các bình luận trong chủ đề của diễn đàn.
2. Gán nhãn dữ liệu thủ công các bình luận.
3. Chuẩn bị, tiền xử lý, phân tích dữ liệu.
4. Xây dựng mô hình dựa trên tập dữ liệu đã gán nhãn.
5. Triển khai mô hình để dự đoán các dữ liệu mới.

3.2. Thu thập dữ liệu

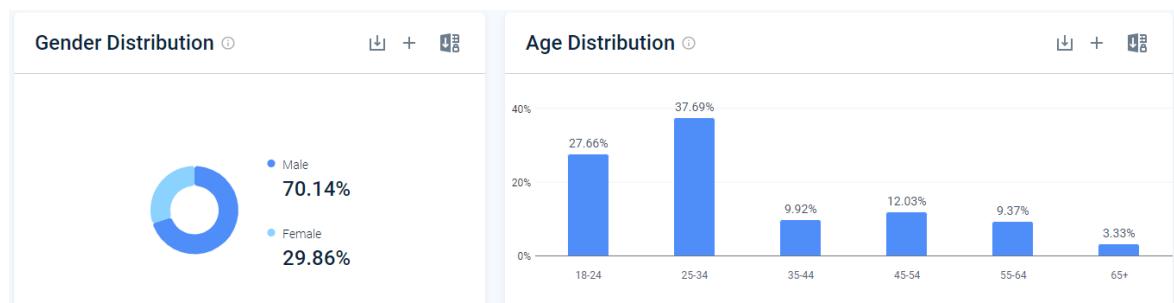
3.2.1. Giới thiệu diễn đàn VOZ

Kể từ khi thành lập, diễn đàn công nghệ Việt Nam VOZ đã trải qua nhiều lần nâng cấp. Tên gọi VOZ là viết tắt của “Vietnam Overclock Zone” (ép xung máy tính) và chỉ được sử dụng từ năm 2004. Cũng từ đây, VOZ trở thành nơi thảo luận đa dạng mọi chủ đề và sau này trở nên nổi tiếng với những tập truyện dài kỳ gắn mác VOZ cùng câu nói nổi tiếng mà các thành viên này hay sử dụng “from VOZ with love”. Tên miền hiện tại của diễn đàn là voz.vn.

Theo dữ liệu thống kê từ Similarweb [12], diễn đàn có khoảng hơn 13 triệu lượt truy cập hàng tháng. Người dùng truy cập hầu hết là nam chiếm khoảng 70%, phân bố độ tuổi khá đa dạng, khoảng độ tuổi từ 25-34 có lượt truy cập lớn nhất.



Hình 3.2. Thống kê lượt truy cập của diễn đàn theo Similarweb.



Hình 3.3. Thống kê nhân khẩu học của diễn đàn theo Similarweb.

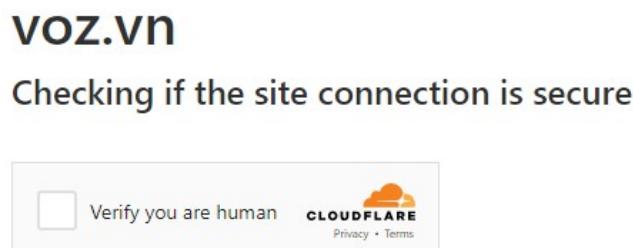
Từ các phân tích ở trên, ta có thể suy luận mức độ tổng quát của tập dữ liệu dự định thu thập. Diễn đàn tập trung vào công nghệ và lượng nữ giới thấp, nên các đánh giá và thảo luận sẽ có xu hướng tập trung vào khía cạnh phản ứng, cầu hình

mà ít quan tâm đến yếu tố ngoại hình. Độ tuổi từ 25-34 là độ tuổi có cân nhắc chi tiêu nên yếu tố về giá sẽ được quan tâm hơn.

3.2.2. Cào dữ liệu trang web VOZ

Bỏ qua sự phát hiện selenium từ Cloudflare

Diễn đàn VOZ được bảo vệ bằng Cloudflare, nên trước khi thực hiện cào dữ liệu, cần truy cập trang web cần cào là người dùng thực (truy cập thủ công) và khởi chạy selenium với hồ sơ trình duyệt tương ứng với trình duyệt đã truy cập trước đó. Khi truy cập trang web cần cào, trình duyệt sẽ lưu cookie và Cloudflare sẽ bỏ qua xác thực khi sử dụng selenium. Hơn nữa, để tránh bị chặn do lượng truy cập lớn từ một IP, cần thêm độ trễ ngẫu nhiên để đợi trang web được tải xong và có hành vi giống người dùng thật nhất có thể.



Hình 3.4. Xác thực CAPTCHA của Cloudflare.

Tiến hành thu thập dữ liệu

Khởi tạo selenium với hồ sơ trình duyệt và tiến hành tải trang web:

```
20 # Create a ChromeOptions object
21 options = webdriver.ChromeOptions()
22
23 profile_path = "C:\\\\Users\\\\Vinh\\\\AppData\\\\Local\\\\Google\\\\Chrome\\\\User Data"
24 options.add_argument(f"--user-data-dir={profile_path}")
25
26 # provide the profile name with which we want to open browser
27 options.add_argument(r"--profile-directory=Profile 1")
28
29 # specify where your chrome driver present in your pc
30 driver = webdriver.Chrome(options=options)
31
32 # crawl page
33 url = "https://voz.vn/t/google-pixel-xuat-sac-o-tam-gia-4-7-trieu-trai-nghiem-su-dung-cach-mua.122469/"
34 print("Crawl page:", url)
35 driver.get(url)
36 sleep(random.randint(10, 12))
```

Hình 3.5. Khởi tạo selenium với hồ sơ trình duyệt.

Xây dựng hàm nhấn chọn vào phần tử và xóa phần tử:

```
1 def click_element(driver, element):
2     driver.execute_script(
3         "arguments[0].click();",
4         WebDriverWait(driver, 20).until(EC.element_to_be_clickable(element)),
5     )
6
7
8 def remove_element(driver, element):
9     driver.execute_script("arguments[0].remove();", element)
10
```

Hình 3.6. Hàm chọn phần tử và xóa phần tử.

Với bình luận dài, trang VOZ hiển thị trong một chế độ xem thu gọn nên cần mở rộng các phần tử ẩn để thu thập được toàn bộ nội dung:

-**UPDATE 1:** con pixel 4 dùng esim viettel ngon lành nhé các bác (esim vina ko nhận, dị vcd), 2 sim 2 sóng.

-**UPDATE 2:** có thím bảo không chọn mua được, em mới thử logout thì đúng là nó éo cho chọn thật, vì lâu rồi ko để ý đoạn này, thôi các thím chịu khó tạo cái acc, dùng sdt VN nó cũng gửi OTP về được.

Về vấn đề nạp tiền vào tài khoản: cá nhân em rút kinh nghiệm về vụ sml với thằng ***** trước đây thì

- Liên hệ với support, hotline với những hệ thống trung bình và lớn. Với những hệ thống nhỏ thì **add Zalo** trao đổi xem có nhận vận chuyển không? Cách thức chuyển tiền như thế nào
- Nhắn tin mà éo trả lời thì cút

- Những hệ thống như ông Tài (tại thời điểm lập thread), xxx, em đánh giá là còn nhỏ và chưa chuyên nghiệp. Chuyển tiền còn phải ping Zalo, chụp ảnh giao dịch để AI chạy cơm nạp cho. baogam chuyển tiền theo cú pháp là 1 phát ăn ngay. Thương gia nhỏ thì phí thường cao hơn, như các thím thấy phí trên kia là đắt hơn so với baogam (baogam mới cập nhật, tăng cách tính cước với hàng điện tử nên nó vượt về giá rồi nhé). Nhưng nó về được hàng. Ping Zalo thường trả lời khá nhanh, hỗ trợ nhiệt tình

- Một số lỗi gặp phải trên pixel 4 0044

- Không có face unlock (cái này hình như là tính năng)

[Click to expand...](#)

Hình 3.7. Bình luận bị thu gọn.

Hàm mở rộng phần tử thu gọn:

```
12 def expand_all_hidden_elements(driver):
13     hidden_elements = driver.find_elements(By.CSS_SELECTOR, ".bbCodeBlock-expandLink")
14     for element in hidden_elements:
15         try:
16             element.click()
17             sleep(random.randint(3, 5))
18         except ElementNotInteractableException:
19             continue
20         except ElementClickInterceptedException:
21             continue
```

Hình 3.8. Hàm hiển thị phần tử thu gọn.

VOZ sử dụng trích dẫn để phản hồi bình luận trước đó thay vì sử dụng hệ thống phân cấp, do đó cần xóa bỏ trích dẫn để tránh trùng lặp nội dung:

Sep 2, 2020 #13
rangeri said: ⏺
đợt trước thấy nghỉ thốn quá , order vẫn về vù vù hả thím , bao nhiêu ngày là tới tay
11 ngày từ lúc chuyển tiền, em có post thông tin đơn hàng pixel 4 trên rồi đó thím. Con 3xl trước là 31 ngày😊😊

Sep 2, 2020 #14
REALMADRIDno1 said: ⏺
11 ngày từ lúc chuyển tiền, em có post thông tin đơn hàng pixel 4 trên rồi đó thím. Con 3xl trước là 31 ngày😊
để sáng mai e ib làm luôn con 3XL 😊😊 đang ham hố vãi chưởng à mà 4gb ram đa nhiệm kém đúng ko thím

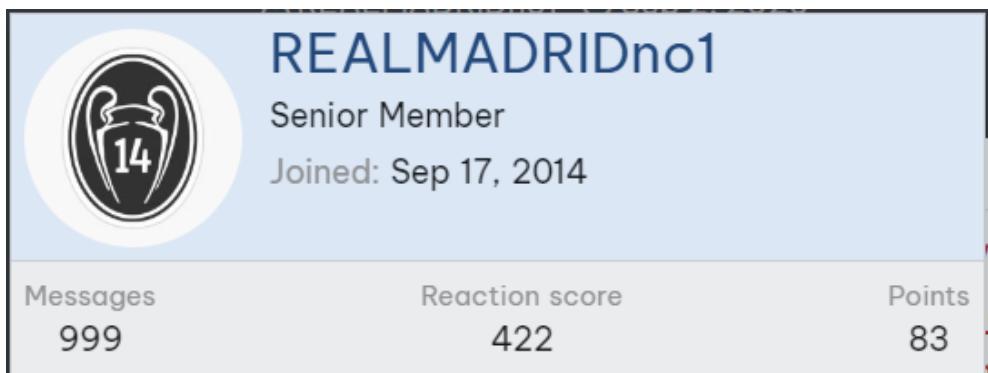
Hình 3.9. Nội dung trích dẫn trùng lặp với bình luận trước đó.

Hàm xóa bỏ phần tử trùng lặp:

```
24 def remove_duplicate_quote_elements(driver):
25     root_quotes = driver.find_elements(By.CSS_SELECTOR, "blockquote")
26     for root_quote in root_quotes:
27         try:
28             quotes = root_quote.find_elements(By.CSS_SELECTOR, "div.bbCodeBlock-title")
29             for quote in quotes:
30                 parent_quote = quote.find_element(By.XPATH, "..")
31                 remove_element(driver, parent_quote)
32         except NoSuchElementException:
33             continue
34         except StaleElementReferenceException:
35             continue
```

Hình 3.10. Hàm xóa nội dung trùng lặp.

Lấy các thông tin về người bình luận (thành viên) gồm tên, ngày tham gia, số lượng bình luận, số điểm phản ứng, và điểm thành tựu:



Hình 3.11. Thông tin về thành viên của diễn đàn.

Hàm lấy thông tin người bình luận (thành viên):

```
38 def get_user_infomation(driver):
39     user_elements = driver.find_elements(By.CSS_SELECTOR, "h4.message-name [href]")
40     username_list = []
41     usertitle_list = []
42     join_date_list = []
43     message_counts_list = []
44     reactions_counts_list = []
45     points_list = []
46     for idx, user_element in enumerate(user_elements):
47         click_element(driver, user_element)
48         sleep(0.8)
49         username_element = driver.find_elements(
50             By.CSS_SELECTOR, "span.memberTooltip-nameWrapper"
51         )[idx]
52         usertitle_element = driver.find_elements(By.CSS_SELECTOR, "span.userTitle")[idx]
53         join_date_element = driver.find_elements(
54             By.CSS_SELECTOR,
55             "div.memberTooltip-blurbContainer > div:nth-child(2) > dl > dd > time",
56         )[idx]
57         message_counts_element = driver.find_elements(
58             By.CSS_SELECTOR, "dl.pairs--rows--centered:nth-child(1) > dd > a"
59         )[idx]
60         reactions_counts_element = driver.find_elements(
61             By.CSS_SELECTOR, "dl.pairs--rows--centered:nth-child(2) > dd"
62         )[idx]
63         points_element = driver.find_elements(
64             By.CSS_SELECTOR, "dl.pairs--rows--centered:nth-child(3) > dd > a"
65         )[idx]
66
67         username_list.append(username_element.text)
68         usertitle_list.append(usertitle_element.text)
69         join_date_list.append(join_date_element.text)
70         message_counts_list.append(message_counts_element.text)
71         reactions_counts_list.append(reactions_counts_element.text)
72         points_list.append(points_element.text)
73
74     return dict(
75         username=username_list,
76         usertitle=usertitle_list,
77         join_date=join_date_list,
78         message_counts=message_counts_list,
79         reactions_counts=reactions_counts_list,
80         points=points_list,
81     )
82 
```

Hình 3.12. Hàm lấy thông tin thành viên.

Lấy văn bản bình luận, thay thế nhiều dòng trống liên tiếp thành một dòng trống, thay thế nhiều khoảng trắng liên tiếp thành một khoảng trắng:

```
84 def get_message_infomation(driver):
85     # get message elements
86     date_elements = driver.find_elements(By.CSS_SELECTOR, "ul.message-attribution-main")
87     comment_elements = driver.find_elements(By.CSS_SELECTOR, ".bbWrapper")
88
89     # get message data
90     date_list = []
91     username_list = []
92     comment_list = []
93     for date_element, comment_element in zip(date_elements, comment_elements):
94         comment = comment_element.text
95         # remove duplicate new lines
96         comment = re.sub("\n{2,}", "\n", comment)
97         # remove duplicate spaces
98         comment = re.sub(" {2,}", " ", comment)
99
100        date_list.append(date_element.text)
101        comment_list.append(comment_element.text)
102
103    return dict(comment_date=date_list, comment=comment_list)
```

Hình 3.13. Hàm lấy văn bản bình luận.

Lấy trang tiếp theo, nếu không có trang tiếp theo, nghĩa là đến trang cuối, trả về False, điều kiện kết thúc vòng lặp và kết thúc quá trình cào dữ liệu:

```
106 def get_next_page(driver):
107     try:
108         next_page_button = driver.find_element(By.CSS_SELECTOR, ".pageNav-jump--next")
109         driver.execute_script("arguments[0].click()", next_page_button)
110         sleep(random.randint(10, 12))
111         return True
112     except NoSuchElementException:
113         return False
```

Hình 3.14. Hàm tải trang tiếp theo.

Lưu mỗi trang vào một sheet trong Excel:

```
129 def save_to_sheet(data: dict, excel_file_path: str, sheet_name: str):
130     df = pd.DataFrame(data)
131
132     with pd.ExcelWriter(excel_file_path, mode="a") as writer:
133         df.to_excel(writer, sheet_name=sheet_name, index=False)
```

Hình 3.15. Hàm lưu dữ liệu vào một sheet vào tệp Excel.

Tổng hợp các hàm để thu thập dữ liệu tự động:

```
136 def crawl_page(driver):
137     # create new excel file
138     if not os.path.exists("data.xlsx"):
139         df = pd.DataFrame()
140         df.to_excel("data.xlsx")
141     # crawl page
142     while True:
143         print("crawl page:", driver.current_url)
144         expand_all_hidden_elements(driver)
145         remove_duplicate_quote_elements(driver)
146         message_data = get_message_infomation(driver)
147         user_data = get_user_infomation(driver)
148         merged_data = {**message_data, **user_data}
149         page_number = get_page_number(driver)
150         save_to_sheet(merged_data, "data.xlsx", str(page_number))
151         if not get_next_page(driver):
152             print("no next page")
153             break
154
155
156 crawl_page(driver)
```

Hình 3.16. Hàm tổng hợp thu thập dữ liệu.

Bộ dữ liệu sau khi cào được lưu trong tệp data.xlsx, tên sheet là số trang, mỗi sheet gồm 20 bình luận tương ứng với số lượng bình luận trong một trang của diễn đàn:

A	B	C	D	E	F	G	H	I
comment_date	comment	username	user title	join_date	message_count	actions_count	points	
2 Sep 2, 2020	- UPDATE 5 (04/REALMADRIDn)	Senior Member	Sep 17, 2014	998	422	83		
3 Sep 2, 2020	quên mất, khỏi	REALMADRIDn	Senior Member	Sep 17, 2014	998	422	83	
4 Sep 2, 2020	Xài pixel sướng	lon_ton_sai_va	Đã tốn tiền	Jul 24, 2006	11,155	6,161	113	
5 Sep 2, 2020	mình cũng đang rangeri	Senior Member	Aug 6, 2009	3,684	2,819	113		
6 Sep 2, 2020	cũng đang mê prangeri	Senior Member	Aug 6, 2009	3,684	2,819	113		
7 Sep 2, 2020	vẫn làm thím n	REALMADRIDn	Senior Member	Sep 17, 2014	998	422	83	
8 Sep 2, 2020	đợt trước thấy	rangeri	Senior Member	Aug 6, 2009	3,684	2,819	113	
9 Sep 2, 2020	thấy đổi stt bar rangeri	Senior Member	Aug 6, 2009	3,684	2,819	113		
10 Sep 2, 2020	Bảo hành bảo t	yeuthaoquyen	Đã tốn tiền	May 30, 2013	7,586	1,187	113	
11 Sep 2, 2020	bảo hành gi	rangeri	Senior Member	Aug 6, 2009	3,684	2,819	113	
12 Sep 2, 2020	được cái giá rẻ	rangeri	Senior Member	Aug 6, 2009	3,684	2,819	113	
13 Sep 2, 2020	Standard của n	vic1806	Member	Nov 1, 2009	1,424	747	113	
14 Sep 2, 2020	11 ngày từ lúc c	REALMADRIDn	Senior Member	Sep 17, 2014	998	422	83	
15 Sep 2, 2020	để sáng mai e i	rangeri	Senior Member	Aug 6, 2009	3,684	2,819	113	
16 Sep 2, 2020	à còn bắn m	rangeri	Senior Member	Aug 6, 2009	3,684	2,819	113	
17 Sep 2, 2020	em chuyên m	REALMADRIDn	Senior Member	Sep 17, 2014	998	422	83	
18 Sep 2, 2020	Hộp này có trù	vic1806	Member	Nov 1, 2009	1,424	747	113	
19 Sep 2, 2020	EU -> có thể un	REALMADRIDn	Senior Member	Sep 17, 2014	998	422	83	
20 Sep 2, 2020	đây là bản test	REALMADRIDn	Senior Member	Sep 17, 2014	998	422	83	
21 Sep 2, 2020	Ngon vãi. Minh	vic1806	Member	Nov 1, 2009	1,424	747	113	
22								

Hình 3.17. Bộ dữ liệu sau khi thu thập.

3.2.3. Gán nhãn dữ liệu

Quy ước gán nhãn

Nhãn đánh giá, nhận xét, thảo luận bao gồm:

- pos: tích cực, khen, ỗn, được, thông tin về sản phẩm.
- neg: tiêu cực, chê.

Nếu đánh giá, nhận xét bị xung đột, nghĩa là có cả tích cực và tiêu cực thì nhãn được gán là nhãn của nhận xét xuất hiện đầu tiên. Các bình luận là các câu hỏi như khắc phục lỗi, hỏi giá thành, nơi mua, các bình luận không rõ ràng hoặc không liên quan về Google Pixel bị bỏ qua và không được gán nhãn. Nếu bình luận không chỉ định rõ ràng là điện thoại nào thì được suy luận nói về Google Pixel.

Các bộ khía cạnh để đánh giá của điện thoại Google Pixel được tham khảo từ trang chuyên đánh giá các điện thoại thông minh [13]:

1. general: nói chung về điện thoại, tổng kết.
2. body: kích thước (dài, rộng, cao), cân nặng (gam), vỏ ngoài (nhựa, nhôm, thép), cảm giác cầm nắm.
3. display: loại hiển thị (OLED, LCD), kích thước màn hình (inch), chất liệu màn hình, độ phân giải (Full HD, 2K), tần số quét (60Hz, 90Hz, 120Hz).

4. platform: hệ điều hành (phiên bản Android, bảo mật hệ thống, cử chỉ, hiệu ứng chuyển động, các ứng dụng), bộ xử lý CPU, GPU.
5. memory: RAM, bộ nhớ trong, bộ nhớ ngoài (thẻ SD).
6. camera: camera trước, camera sau, khả năng quay chụp.
7. sound: số lượng loa, cổng cắm tai nghe 3.5mm, chất lượng âm thanh.
8. comms: các cổng giao tiếp, chuẩn Wi-Fi, mạng di động, sim, Bluetooth, định vị, NFC, radio, cổng USB.
9. features: các cảm biến (khuôn mặt, vân tay), bộ rung.
10. misc: các yếu tố khác, ví dụ: màu sắc vỏ, unlock bootloader, tải ảnh lên Google Photos.
11. price: các yếu tố về giá (đắt hoặc rẻ), giá điện thoại, phụ kiện đi kèm, chính sách bảo hành, dịch vụ vận chuyển, trạng thái máy (mua mới, đã sửa chữa).

Ví dụ về cách gán nhãn

Tổng quát, gán nhãn sẽ có cấu trúc như sau:

Text

<Khía cạnh 1> pos hoặc neg

<Khía cạnh 2> pos hoặc neg

1. Nói chung (general) về điện thoại Google Pixel:
 - (pos) Vẫn đang dùng XL 1 đây, thấy nó vẫn đập ứng được nhu cầu sử dụng hàng ngày.
 - (pos) ảnh lộ con px5 thím.
 - (pos) sắp về rồi đó thím, nhớ review nhẹ để em quất 1 con.
2. Ngoại hình (body):
 - (pos) Nhìn vỏ nguyên khối kia chắc là nhựa rồi. Nhưng dc cái mặt trước cầm mỏng, đẹp.
 - (neg) pixel 5 vỏ nhôm pha ke à.
 - (pos) tiêu chuẩn chính thức máy đẹp leng keng từ khay sim đến lỗ loa.
3. Màn hình (display):
 - (neg) Mình mới hỏng màn con Pixel XL.

- (pos) Xem review thì thấy bọn nào cũng bảo màn ổn, ko lởm đâu.

- (neg) Đang tính chốt em 3xl mà thấy cái vụ tai trâu

4. Nền tảng (platform):

- (pos) máy tác vụ cơ bản em thấy nó vẫn mượt như android 10 thôi.

- (neg) không cầu hình cũ quá rồi, pixel 2 còn đuối nữa là pixel.

- (pos) gần như không có gì phải chê trừ việc nó dùng emmc 5.1. Giờ lên 4a ufs 2.1 ngon choết.

5. Bộ nhớ (memory):

- (pos) chuẩn, ram 6Gb và 4Gb là sự khác biệt kha khá đây. 4Gb hay tràn ram lắm dù đã được tối ưu rất tốt.

- (neg) Điểm nữa là ko có thẻ nhớ mở rộng.

- (pos) 64GB là đủ xài cho nhu cầu bình thường của đa số rồi ấy.

6. Ống kính (camera):

- (pos) Cam gần như 4XL mà có thêm quả cam Wide (dù ko rộng bằng hằng khác nhưng cũng ok).

- (neg) thím nào có link Gcam bản mới nhất ko cho mình xin về cài chứ app mặc định nó cứ kêu tách tách mà bản jp ko tắt được.

- (pos) Con 5 cũng 2 cam mà.

7. Âm thanh (sound):

- (neg) Vè âm thanh. thu âm pixel 3 thua.

- (neg) Loa nghe dở hơn nè, thu âm tệ hơn nè.

- (neg) nghe nói các review chê loa dưới màn con pixel 5.

8. Giao tiếp (comms):

- (pos) chõ nào bán thì mình không biết nhưng mình dùng pixel 5 với esim viettel ngon nhé!

- (pos) Rung thì ngon như bên iPhone. Rung tê cả tay luôn.

- (neg) đã thê lại không 5G.

9. Tính năng (features):

- (neg) Nhận diện khuôn mặt ko dùng để log in được.

- (pos) Faceid mở khoá ngon ko kém iphone, đỉnh luôn.

- (neg) mỗi tội thời buổi covid nên cái 3D Face Unlock không tiện dụng mấy.

10. Yếu tố khác (misc):

- (pos) Nhớ hình như là 1 cái unlock bootloader được đó fen.
- (pos) kiêm con pixel lấy gg photo free cũng ổn :3.
- (pos) Pixel giờ phô biến rồi, đầy thợ sửa.

11. Giá thành (price):

- (neg) Google chưa có ý định phân phối chính hãng các máy Pixel.
- (pos) 4XL giá đang tốt, thời gian tới có lẽ vẫn giảm đều (và chậm).
- (pos) Bây giờ 2 con 4a với 5a xuống khoảng 10tr có khi cũng là lựa chọn đáng giá.

Cấu hình gán nhãn bình luận với Label Studio

Gộp các sheet thành một sheet:

```
1 xls = pd.ExcelFile('data.xlsx')
2 sheet_names = xls.sheet_names
3
4 # Initialize an empty DataFrame to store the combined data
5 combined_data = pd.DataFrame()
6
7 # Iterate through the sheet names, check if a sheet is empty, and concatenate
8 for sheet_name in sheet_names:
9     df = pd.read_excel(xls, sheet_name)
10    if not df.empty:
11        combined_data = pd.concat([combined_data, df], ignore_index=True)
12
13 # Save the combined data to a new Excel file
14 combined_data.to_excel('done.xlsx', index=False)
```

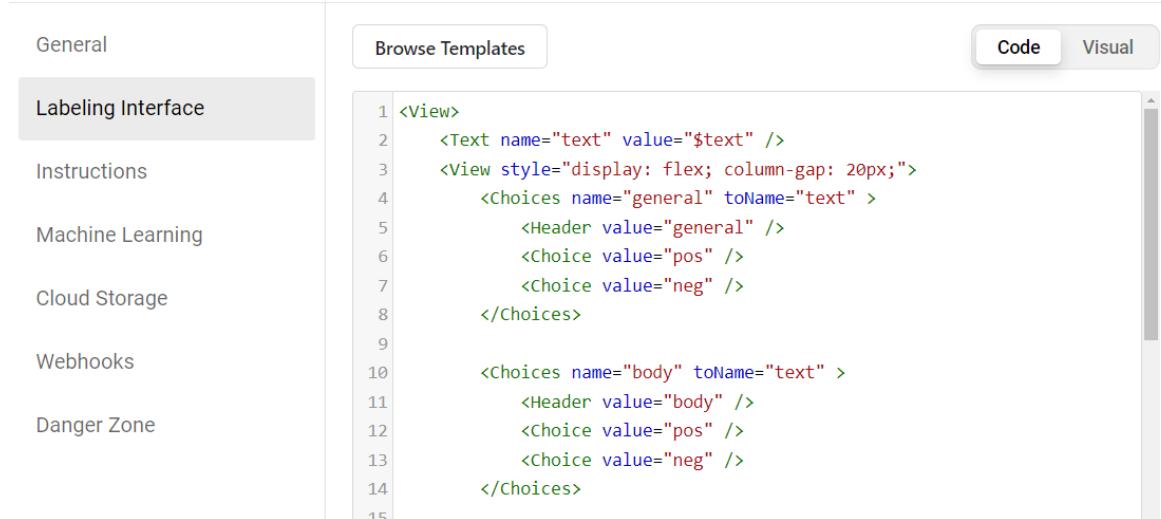
Hình 3.18. Gộp các sheet thành một sheet tổng hợp.

Label Studio không hỗ trợ định dạng Excel nên ta phải chuyển đổi các bình luận thành tệp văn bản có phần mở rộng “txt” để nhập dữ liệu.

```
1 df = pd.read_excel("done.xlsx")
2 with open("comments.txt", "w", encoding="utf-8") as f:
3     for comment in df['comment']:
4         # replace all newlines with spaces
5         f.write(re.sub("\n+", " ", str(comment)))
6         f.write("\n")
```

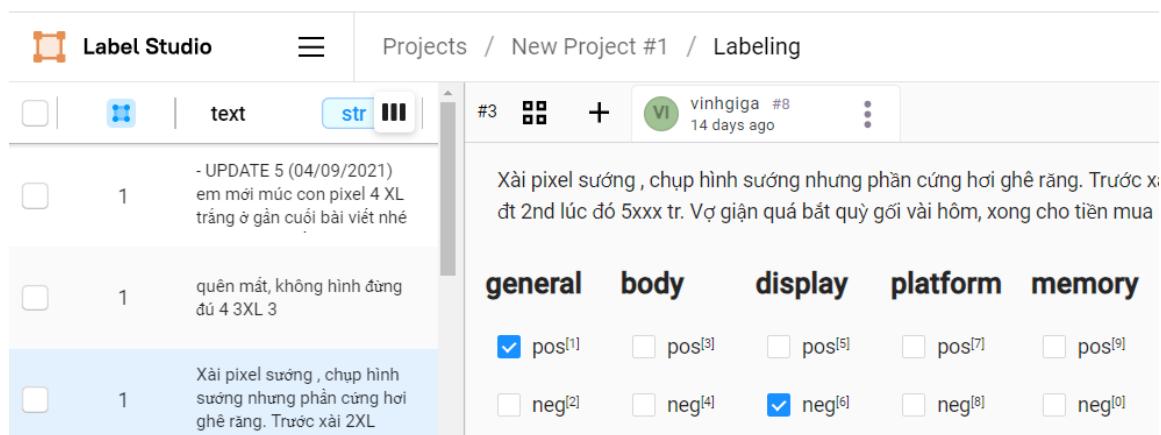
Hình 3.19. Chuyển dữ liệu sang tệp văn bản.

Label Studio chưa có mẫu nào được tích hợp sẵn trong hồ trợ quy ước gán nhãn ở trên, do đó, ta phải thiết lập cấu hình bổ sung để đáp ứng với yêu cầu:



Hình 3.20. Giao diện cấu hình tùy chỉnh với Label Studio.

Giao diện gán nhãn sau khi cấu hình như sau:



Hình 3.21. Giao diện gán nhãn.

Sau khi gán nhãn xong, ta xuất dữ liệu gán nhãn ra tệp JSON. Dữ liệu xuất được hiển thị trong hình sau:

```
1 [  
2 {  
3     "text": "- UPDATE 5 (04\\09\\2021) em mới mua con pixel 4 XL  
4     "id": 1,  
5     "platform": "pos",  
6     "features": "neg",  
7     "camera": "pos",  
8     "battery": "pos",  
9     "sound": "neg",  
10    "price": "neg",  
11    "display": "neg",  
12    "comms": "neg",  
13    "body": "pos",  
14    "annotator": 1,  
15    "annotation_id": 9,  
16    "created_at": "2023-09-24T05:16:30.896564Z",  
17    "updated_at": "2023-09-24T05:16:30.896564Z",  
18    "lead_time": 1871.085  
19 },  
20 {  
21     "text": "quên mất, không hình đứng đú 4 3XL 3",  
22     "id": 2,  
23     "annotator": 1,  
24     "annotation_id": 17,  
25     "created_at": "2023-09-24T05:36:30.400109Z",  
26     "updated_at": "2023-09-24T05:36:30.400109Z",  
27     "lead_time": 1.325  
28 }]
```

Hình 3.22. Dữ liệu tệp JSON.

Tính toán nhãn tổng thể

Từ dữ liệu gán nhãn theo khía cạnh, tính toán mức độ hài lòng tổng thể thành một nhãn duy nhất là tích cực hoặc tiêu cực. Nhãn tổng thể được tính bằng trung bình cộng các nhãn theo khía cạnh, nếu ngưỡng lớn hơn hoặc bằng trung bình (0,5) thì nhãn tổng thể là hài lòng và ngược lại.

```
10 categories = [
11     "general",
12     "body",
13     "display",
14     "platform",
15     "memory",
16     "camera",
17     "sound",
18     "comms",
19     "features",
20     "misc",
21     "price",
22 ]
23 # read data from json file
24 with open("labels.json", encoding="utf-8") as f:
25     data = json.load(f)
26 # calculate labels
27 comment_list = []
28 label_list = []
29 for comment in data:
30     sum = 0
31     count = 0
32     for key, value in comment.items():
33         if key in categories:
34             weight = 1 if value == "pos" else 0
35             sum += weight
36             count += 1
37     # check if the comment has category
38     if count > 0:
39         comment_list.append(comment["text"])
40         average = sum / count
41         # set threshold to 0.5
42         if average ≥ 0.5:
43             label_list.append("pos")
44         else:
45             label_list.append("neg")
46 # save to csv file
47 df = pd.DataFrame({"comment": comment_list, "label": label_list})
48 df.to_csv("labeled_data.csv", index=False)
```

Hình 3.23. Tính toán nhãn tổng thể.

Nhận xét về quá trình gán nhãn

Gán nhãn văn bản thường được xử lý thủ công, tốn nhiều thời gian và công sức. Người gán nhãn cần có những hiểu biết nhất định về miền dữ liệu gán nhãn. Ngoài ra, cần có các công cụ quản lý gán nhãn để đảm bảo thống nhất giữa những người gán nhãn, dự đoán gán nhãn để tăng tốc quá trình gán nhãn.

3.2.4. Mô tả tập dữ liệu

Mô tả tập dữ liệu thô sau khi càò

```
RangeIndex: 7479 entries, 0 to 7478
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype  
 --- 
  0   comment_date    7479 non-null   object  
  1   comment        7457 non-null   object  
  2   username       7461 non-null   object  
  3   usertitle      7461 non-null   object  
  4   join_date      7461 non-null   object  
  5   message_counts 7461 non-null   object  
  6   reactions_counts 7461 non-null   object  
  7   points         7461 non-null   float64 
dtypes: float64(1), object(7)
memory usage: 467.6+ KB
```

Hình 3.24. Mô tả tập dữ liệu thô.

Tập dữ liệu chứa 7479 bình luận từ các thành viên của diễn đàn VOZ. Ý nghĩa của các thuộc tính được trình bày dưới đây:

- comment_date: Ngày bình luận.
- comment: Nội dung của bình luận.
- username: Tên người bình luận.
- usertitle: Danh hiệu người dùng, VOZ có các danh hiệu chính là Junior Member, Member, Senior Member, Moderator. Dựa vào danh hiệu này, có thể đánh giá mức độ hoạt động của người dùng đó trên diễn đàn.
- join_date: Ngày đăng ký vào diễn đàn.
- message_counts: Tổng số lượt bình luận trên các chủ đề của diễn đàn.
- reactions_counts: được tính bằng tổng số lượt phản ứng tích cực (ưng) trừ tổng số lượt phản ứng tiêu cực (gạch).
- points: tổng điểm thành tựu khi đạt các mốc về số lượt bình luận, lượt phản ứng.

Mô tả tập dữ liệu gán nhãn

```
RangeIndex: 1581 entries, 0 to 1580
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype  
 --- 
  0   text        1581 non-null    object 
  1   general     78 non-null     object 
  2   body         28 non-null     object 
  3   display      58 non-null     object 
  4   platform    67 non-null     object 
  5   memory       7 non-null     object 
  6   camera       48 non-null     object 
  7   sound        11 non-null     object 
  8   comms        26 non-null     object 
  9   features     32 non-null     object 
  10  misc         34 non-null     object 
  11  price        237 non-null    object 
dtypes: object(12)
memory usage: 148.3+ KB
```

Hình 3.25. Dữ liệu được gán nhãn.

Vì thời gian gán nhãn hạn chế, tập dữ liệu được gán nhãn chỉ có 1581 mẫu. Trong đó, có rất nhiều bình luận không có thông tin nhãn về khía cạnh (không đáp ứng quy ước gán nhãn). Nghĩa là những bình luận mơ hồ, bình luận có nội dung không liên quan đến chủ đề, bình luận rỗng (người bình luận chỉ đăng hình ảnh) hoặc bình luận là những câu hỏi. Dữ liệu có 11 khía cạnh đã được mô tả trong phần gán nhãn.

Mô tả tập dữ liệu được gán nhãn tổng thể (dữ liệu để đào tạo mô hình):

```
RangeIndex: 507 entries, 0 to 506
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
 --- 
  0   comment     507 non-null    object 
  1   label       507 non-null    object 
dtypes: object(2)
memory usage: 8.0+ KB
```

Hình 3.26. Mô tả tập dữ liệu sau gán nhãn.

Trong số 1581 mẫu đã gán nhãn, chỉ có 507 mẫu đạt các yêu cầu về quy ước gán nhãn. Ta tính toán gán nhãn tổng thể được trình bày trong phần gán nhãn

dữ liệu, thu được 507 mẫu, gồm 2 thuộc tính là comment và label tương ứng với bình luận và nhãn để phân loại.

3.3. Chuẩn bị dữ liệu

3.3.1. Phân tích dữ liệu

Phân tích dữ liệu thô sau khi cào

Xử lý dữ liệu ngày tháng để phân tích dữ liệu:

```

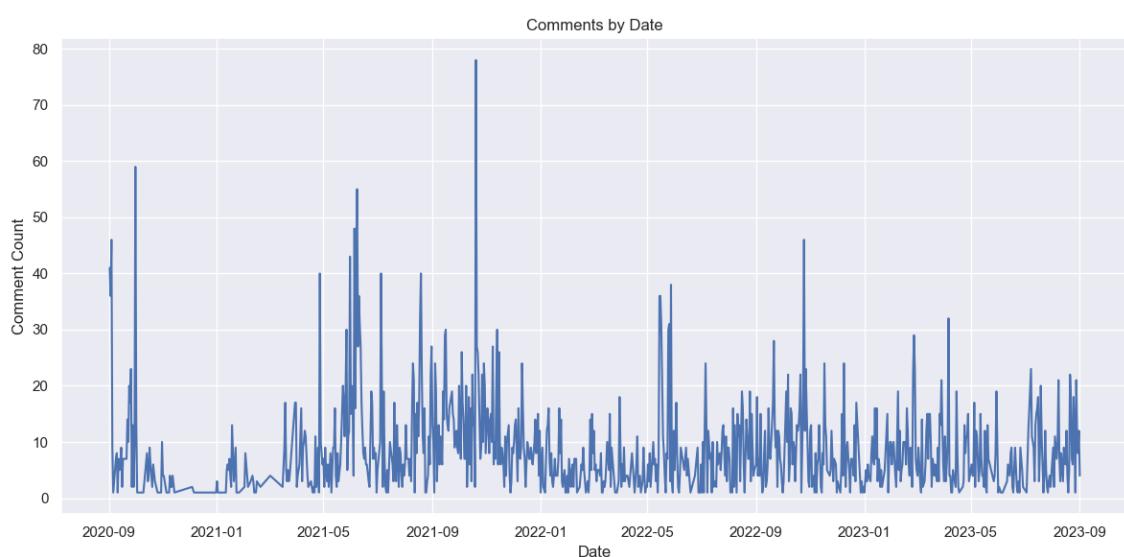
1 def convert_to_ymd(date_str):
2     if date_str is np.nan:
3         return np.nan
4     date_obj = datetime.strptime(str(date_str), '%b %d, %Y')
5     return date_obj.strftime('%Y-%m-%d')
6
7 def convert_to_int(text):
8     if text is np.nan:
9         return 0
10    return int(str(text).replace(',', ''))
```

✓ 0.5s

	comment_date	comment	username	usertitle	join_date	message_counts	reactions_counts	points
856	2021-05-07	px3 làm thế nào để bóp nó ra chức năng ...	KoDoThey	Senior Member	2012-12-15	4684	606	113.0
936	2021-05-23	nhớ review từ đâu đến đít luôn nhé\nmà h Pixel...	WENBIE	Member	2007-10-21	43829	9926	113.0
1678	2021-07-09	Minh coi lại rồi. Hình như chỉ free khi upload...	jnguyen87	Senior Member	2014-08-27	243	82	28.0
3971	2022-03-10	mua pixel 6 lock mỹ hoặc nhật về VN unlock dc ...	zoodkool	Junior Member	2014-09-20	160	13	18.0
920	2021-05-20	Pixel có cách nào chụp màn hình home được k cá...	Vợ bạn	Junior Member	2020-11-17	63	44	18.0

Hình 3.27. Chuyển đổi dữ liệu ngày tháng sang dạng chuẩn.

Phân phối bình luận theo ngày:

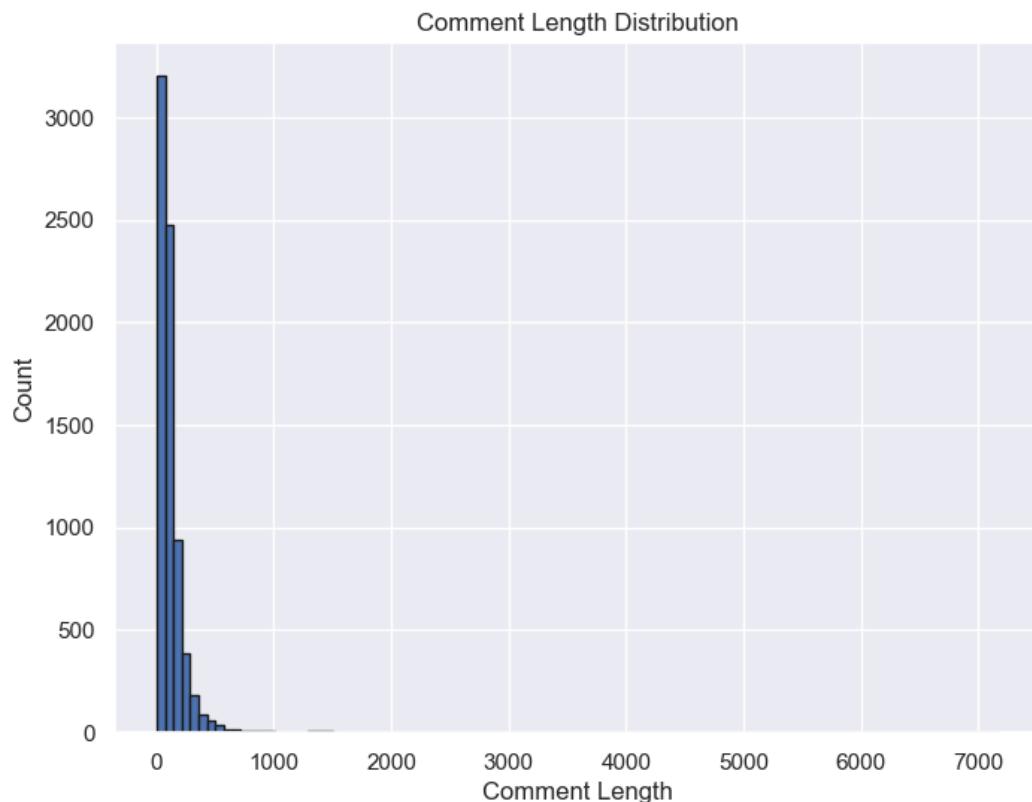


Hình 3.28. Phân phối số lượng bình luận theo ngày.

Nhận xét: Chủ đề này đã duy trì được lượng bình luận theo thời gian, từ lúc bắt đầu lập chủ đề đến nay. Khoảng tháng 6/2021, chủ đề được hoạt động tích cực

hơn. Đây là khoảng thời gian Google Photos thay đổi chính sách, chỉ miễn phí tải lên ảnh từ điện thoại Pixel, cho thấy bước đi hợp lý của Google, làm cho Pixel trở nên hấp dẫn hơn và tạo sự chú ý đến người dùng mới. Ngoài ra, việc duy trì được bình luận theo thời gian tạo ra sự khác biệt của diễn đàn VOZ, so với các mạng xã hội như Facebook chỉ hoạt động tích cực trong thời gian ngắn.

Thống kê độ dài của bình luận:



Lấy thông tin số lượng người bình luận:

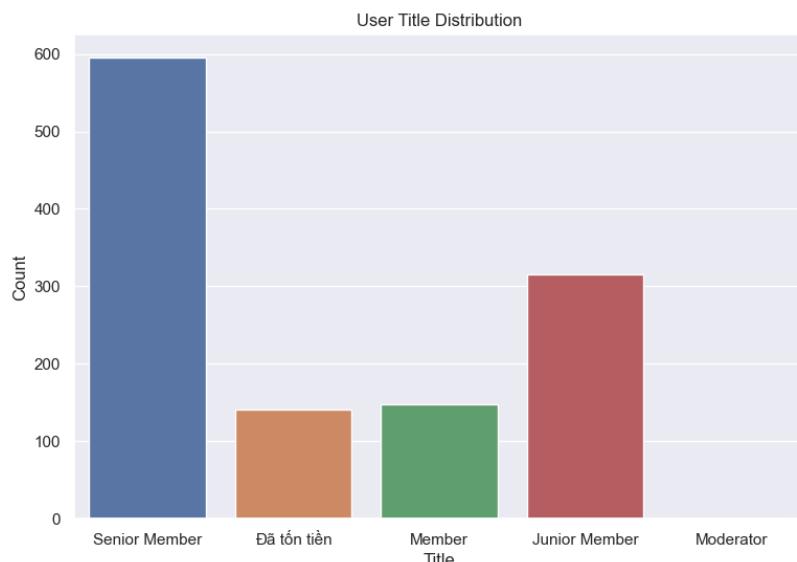
```
1 # remove duplicate usernames
2 df_user.drop_duplicates(subset="username", inplace=True)
3 # get number of unique usernames
4 num_unique_usernames = len(df_user["username"].unique())
5 print("Số lượng người dùng tham gia bình luận:", num_unique_usernames)
6 print(
7     "Tỷ lệ % số người bình luận trong tổng số bình luận:",
8     num_unique_usernames / len(df),
9 )
✓ 0.0s
```

Số lượng người dùng tham gia bình luận: 1208
Tỷ lệ % số người bình luận trong tổng số bình luận: 0.16151891964166332

Hình 3.30. Thông tin về số lượng người bình luận.

Nhận xét: tỷ lệ số người bình luận (thành viên) khác nhau trong tổng số bình luận rất thấp, cho thấy Google Pixel có một tập người dùng cụ thể.

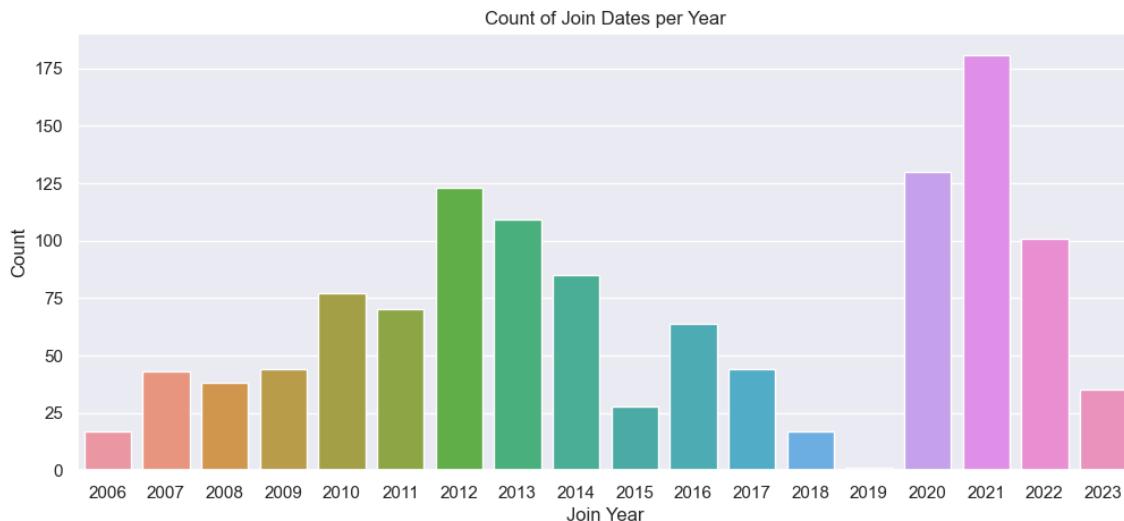
Thống kê danh hiệu của các thành viên:



Hình 3.31. Thông tin về danh hiệu của các thành viên.

Nhận xét: Danh hiệu của thành viên phần lớn là Senior Member. Đây là danh hiệu có được sau khi đạt đủ số lượng bình luận nhất định và một số yêu cầu về bảo mật khác.

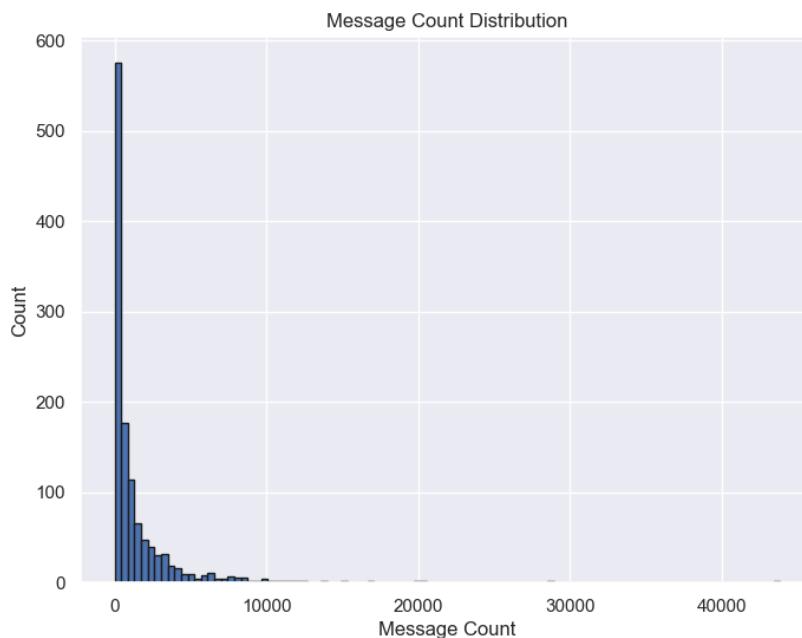
Thống kê ngày đăng ký làm thành viên của diễn đàn:



Hình 3.32. Thông kê ngày đăng ký làm thành viên.

Nhận xét: Mặc dù trải qua nhiều lần dừng hoạt động, nhưng diễn đàn vẫn thu hút được lượng thành viên lâu năm gắn bó với diễn đàn. Vào năm 2020, diễn đàn chuyển sang công nghệ mới và tên miền mới, đã nhanh chóng thu hút được lượng thành viên.

Thông kê tổng số lượng bình luận từng thành viên:



Hình 3.33. Phân phối tổng số lượng bình luận mỗi thành viên.

Nhận xét: Một số thành viên hoạt động sôi nổi, biểu thị bởi số lượng bình luận hơn 40000, hầu hết thành viên ít hoạt động hoặc mới đăng ký diễn đàn.

Thông kê về điểm phản ứng mà thành viên nhận được:

```
1 df_user['reactions_counts'].describe()
✓ 0.0s

count    1.208000e+03
mean     -3.953343e+03
std      1.681195e+05
min     -5.838025e+06
25%      1.800000e+01
50%      1.315000e+02
75%      6.232500e+02
max      2.069980e+05
Name: reactions_counts, dtype: float64
```

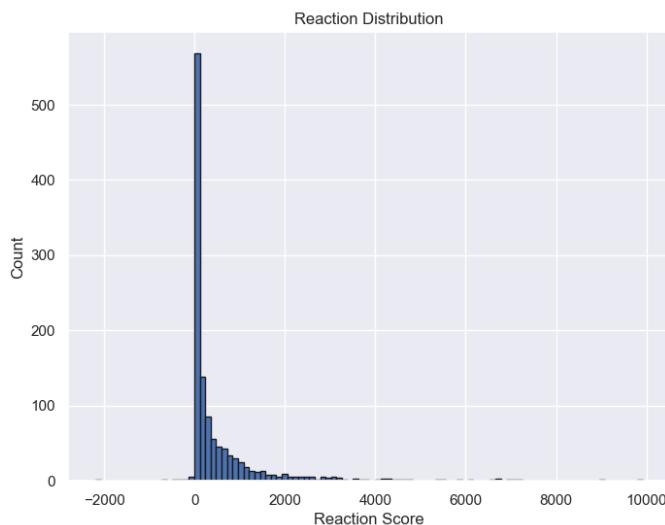
Hình 3.34. Thống kê về số điểm phản ứng.

Nhận xét: có sự bất thường về lượt phản ứng, có thành viên đạt hơn 200 nghìn lượt tích cực, và có thành viên nhận hơn 5 triệu lượt phản ứng tiêu cực. Trong khi đó, trung vị của dữ liệu chỉ khoảng 130. Do đó cần xử lý dữ liệu ngoại lai, trong trường hợp này thay thế giá trị ngoại lai về 0:

```
1 # edit the reactions_counts column in a DataFrame
2 # so that values outside the range [-10000, 10000] are set to 0
3 df_user["reactions_counts"] = df_user["reactions_counts"].apply(
4     lambda x: 0 if x < -10000 or x > 10000 else x
5 )
```

Hình 3.35. Xử lý dữ liệu điểm phản ứng ngoại lai.

Phân phối điểm phản ứng sau khi xử lý ngoại lai:

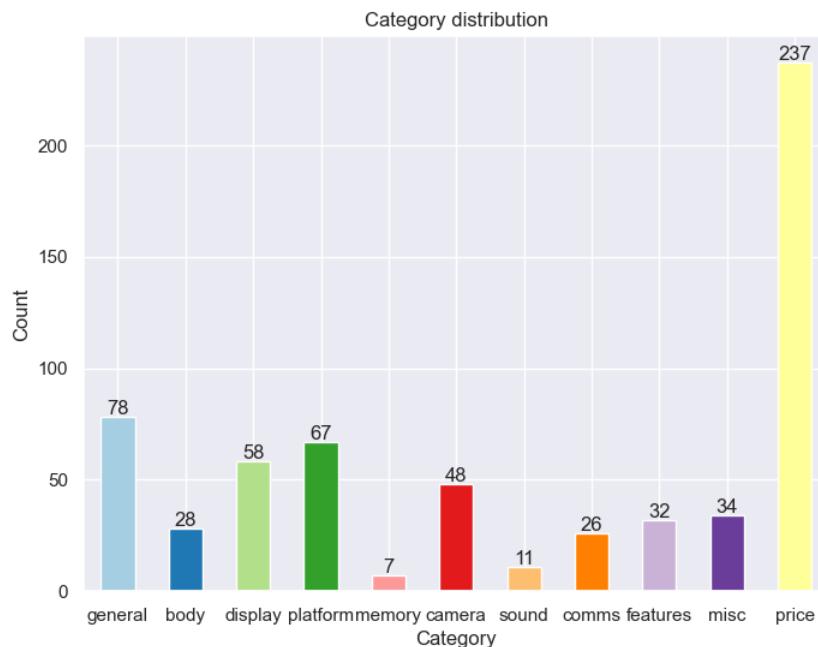


Hình 3.36. Phân phối về số điểm phản ứng.

Nhận xét: các thành viên ít quan tâm đến phản ứng cảm xúc, phần lớn các thành viên có ít hơn 2000 tương tác.

Phân tích dữ liệu đạt quy ước gán nhãn (507 mẫu)

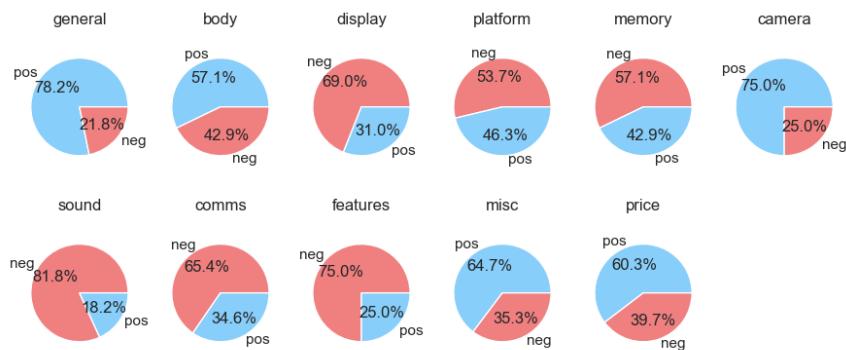
Thống kê các khía cạnh:



Hình 3.37. Thống kê số lượng từng khía cạnh.

Nhận xét: Như đã chỉ ra trong phần giới thiệu về diễn đàn, yếu tố về giá được quan tâm hơn cả.

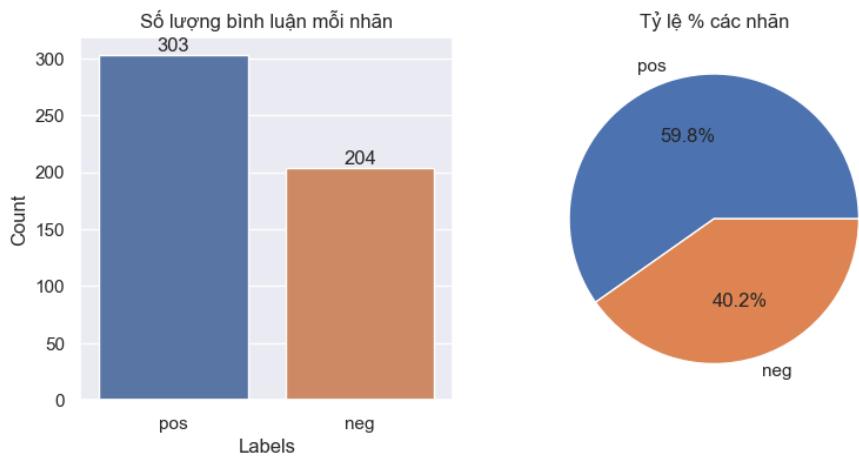
Tỷ lệ tích cực và tiêu cực từng khía cạnh:



Hình 3.38. Phân tích tỷ lệ tích cực và tiêu cực từng khía cạnh.

Nhận xét: dựa trên những khía cạnh này, người quan tâm về Google Pixel có thể cân nhắc có nên mua điện thoại không. Nhà sản xuất cũng rút ra được các chiến lược cải tiến sản phẩm. Như biểu đồ thể hiện, điện thoại Pixel nổi bật về các yếu tố về máy ảnh và giá thành. Mặt khác, những khía cạnh chung phần lớn là tích cực nhưng có nhiều khía cạnh đánh giá sản phẩm là tiêu cực (màn hình, âm thanh, tính năng).

Kiểm tra cân bằng dữ liệu được gán nhãn tổng thể (dữ liệu để xây dựng mô hình):



Hình 3.39. Kiểm tra cân bằng tập dữ liệu đã gán nhãn.

Nhận xét: Ta thấy dữ liệu tương đối cân bằng. Dữ liệu cân bằng giúp tránh việc mô hình học lệch về một tiêu chí cụ thể.

3.3.2. Tiết xử lý dữ liệu

Bước tiền xử lý dữ liệu nhằm mục đích cải thiện chất lượng của dữ liệu bằng cách loại bỏ các thành phần không quan trọng hoặc gây hiểu nhầm, ví dụ như liên kết URL, lỗi chính tả, các ký hiệu.

Hình 3.40. Tiết xử lý văn bản bình luận.

3.3.3. Phân tách tập đào tạo và tập kiểm thử

Vì tập dữ liệu là rất ít (507 mẫu), ta sử dụng tỷ lệ 80% cho tập đào tạo và 20% cho tập kiểm thử. Nếu tỷ lệ lớn hơn, mô hình có thể gặp vấn đề quá khớp dữ liệu, nghĩa là mô hình cho kết quả cao trên tập đào tạo nhưng lại có kết quả thấp trên tập kiểm thử, và mô hình không được tổng quát hóa. Nếu tỷ lệ nhỏ hơn, mô hình có thể gặp vấn đề chưa khớp dữ liệu, nghĩa là cả kết quả đào tạo và kiểm thử đều thấp, do chưa đủ dữ liệu để mô hình có thể trích xuất đặc trưng. Ngoài ra, cần xáo trộn dữ liệu để các nhãn lớp được phân bố đồng đều trên cả tập đào tạo và kiểm thử, giúp mô hình tránh học lệch về các khía cạnh cụ thể.

```
1 # split data to train and test set
2 from sklearn.model_selection import train_test_split
3
4 train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)
5 # save to csv files
6 train_df.to_csv("data/train.csv", encoding="utf-8", index=False)
7 test_df.to_csv("data/test.csv", encoding="utf-8", index=False)
```

Hình 3.41. Tách tập dữ liệu ra thành tập đào tạo và tập kiểm thử.

Thông tin dữ liệu sau khi phân chia: Số mẫu của tập đào tạo và tập kiểm thử lần lượt là 405 và 102, ứng với tỷ lệ 80-20%. Các mẫu không có dữ liệu thiếu.

```
<class 'pandas.core.frame.DataFrame'>
Index: 405 entries, 444 to 102
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   comment   405 non-null    object 
 1   label     405 non-null    object 
dtypes: object(2)
memory usage: 9.5+ KB
<class 'pandas.core.frame.DataFrame'>
Index: 102 entries, 173 to 75
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   comment   102 non-null    object 
 1   label     102 non-null    object 
```

Hình 3.42. Thông tin tập đào tạo và tập kiểm thử.

3.4. Xây dựng mô hình

3.4.1. Thông số cấu hình máy tính để đào tạo mô hình

Mô hình học sâu yêu cầu tài nguyên tính toán lớn, Google Colab cung cấp phiên bản GPU miễn phí chạy trên web cho phép người nghiên cứu thử nghiệm mô hình.

Thông số được sử dụng để chạy mô hình:

- RAM hệ thống: 12.7GB.
- RAM GPU: 15GB.
- GPU T4.
- Hệ điều hành: Ubuntu 22.04.2 LTS.

3.4.2. Đào tạo mô hình

Kết nối Google Colab với Google Drive để thuận tiện cho việc nhập dữ liệu:

```
[2] from google.colab import drive  
drive.mount('/content/gdrive')
```

Mounted at /content/gdrive

```
[3] !ln -s /content/gdrive/MyDrive /mydrive
```

Hình 3.43. Kết nối Google Colab với Google Drive.

Sử dụng GPU, đặt epochs là 8 và chia tập đào tạo thành 10 phần:

```
device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')  
EPOCHS = 8  
N_SPLITS = 10
```

Hình 3.44. Sử dụng GPU để đào tạo mô hình.

Trong học sâu, epoch được sử dụng để mô tả một lần chuyển toàn bộ tập dữ liệu đào tạo trong quá trình đào tạo mạng. Tăng epoch giúp mô hình hội tụ đến một giải pháp tốt hơn, nhưng nó cũng có nguy cơ quá khớp dữ liệu. Vì vậy, việc tìm đúng số lượng epoch là rất quan trọng, thường liên quan đến các kỹ thuật như dừng sớm hoặc giám sát hiệu suất trên một tập dữ liệu xác thực riêng biệt.

Tập đào tạo được áp dụng phương pháp lấy mẫu kiểm thử chéo. Kiểm thử chéo thường được sử dụng để so sánh và chọn ra mô hình tốt nhất cho một bài toán. Kỹ thuật này được áp dụng trong trường hợp dữ liệu không được nhiều.

Lấy dữ liệu tập đào tạo và kiểm thử:

```
def get_data(path):
    return pd.read_csv(path)

train_df = get_data('/mydrive/data/train.csv')
test_df = get_data('/mydrive/data/test.csv')
# We will use Kfold later
skf = StratifiedKFold(n_splits=N_SPLITS)
for fold, (_, val_) in enumerate(skf.split(X=train_df, y=train_df.label)):
    train_df.loc[val_, "kfold"] = fold
```

Hình 3.45. Lấy dữ liệu từ tệp.

Sử dụng bộ mã hóa được đào tạo trước phoBERT. PhoBERT được huấn luyện trên 20Gb dữ liệu tiếng Việt nhằm giải quyết các bài toán cho tiếng Việt.

```
tokenizer = AutoTokenizer.from_pretrained("vinai/phobert-base", use_fast=False)

Downloading (...)lve/main/config.json: 100% [██████████] 557/557
Downloading (...)solve/main/vocab.txt: 100% [██████████] 895k/895
Downloading (...)solve/main/bpe.codes: 100% [██████████] 1.14M/
Downloading (...)/main/tokenizer.json: 100% [██████████] 3.13M/3.
```

Hình 3.46. Sử dụng bộ mã hóa phoBERT.

Chuẩn bị dữ liệu đầu vào cho mô hình:

```
class SentimentDataset(Dataset):
    def __init__(self, df, tokenizer, max_len=120):
        self.df = df
        self.max_len = max_len
        self.tokenizer = tokenizer

    def __len__(self):
        return len(self.df)

    def __getitem__(self, index):
        row = self.df.iloc[index]
        text, label = self.get_input_data(row)

        encoding = self.tokenizer.encode_plus(
            text,
            truncation=True,
            add_special_tokens=True,
            max_length=self.max_len,
            padding='max_length',
            return_attention_mask=True,
            return_token_type_ids=False,
            return_tensors='pt',
        )

        return {
            'text': text,
            'input_ids': encoding['input_ids'].flatten(),
            'attention_masks': encoding['attention_mask'].flatten(),
            'targets': torch.tensor(label, dtype=torch.long),
        }
```

Hình 3.47. Chuẩn bị dữ liệu đầu vào cho mô hình học sâu.

Tham số `max_len` chỉ định độ dài tối đa của đầu vào được mã hóa. Các văn bản dài hơn sẽ bị cắt bớt và các văn bản ngắn hơn sẽ được đệm.

Xây dựng mô hình phân loại:

```
class SentimentClassifier(nn.Module):
    def __init__(self, n_classes):
        super(SentimentClassifier, self).__init__()
        self.bert = AutoModel.from_pretrained("vinai/phobert-base")
        self.drop = nn.Dropout(p=0.3)
        self.fc = nn.Linear(self.bert.config.hidden_size, n_classes)
        nn.init.normal_(self.fc.weight, std=0.02)
        nn.init.normal_(self.fc.bias, 0)

    def forward(self, input_ids, attention_mask):
        last_hidden_state, output = self.bert(
            input_ids=input_ids,
            attention_mask=attention_mask,
            return_dict=False # Dropout will errors if without this
        )

        x = self.drop(output)
        x = self.fc(x)
        return x
```

Hình 3.48. Xây dựng cấu trúc mô hình học sâu.

Dropout xác định xác suất bỏ học là 0,3. Nó được sử dụng để ngăn chặn việc quá khớp dữ liệu bằng cách đặt một phần đơn vị (nút mạng) về 0 trong quá trình đào tạo. Ta cũng khởi tạo trọng số và độ lệch với phân phối chuẩn.

Hàm đào tạo mô hình:

```
def train(model, criterion, optimizer, train_loader):
    model.train()
    losses = []
    correct = 0

    for data in train_loader:
        input_ids = data['input_ids'].to(device)
        attention_mask = data['attention_masks'].to(device)
        targets = data['targets'].to(device)

        optimizer.zero_grad()
        outputs = model(
            input_ids=input_ids,
            attention_mask=attention_mask
        )

        loss = criterion(outputs, targets)
        _, pred = torch.max(outputs, dim=1)

        correct += torch.sum(pred == targets)
        losses.append(loss.item())
        loss.backward()
        nn.utils.clip_grad_norm_(model.parameters(), max_norm=1.0)
        optimizer.step()
        lr_scheduler.step()

    print(f'Train Accuracy: {correct.double()/len(train_loader.dataset)} Loss: {np.mean(losses)}')
```

Hình 3.49. Hàm đào tạo mô hình học sâu.

Hàm đào tạo chuyển các đầu vào sang thiết bị (GPU), tính toán tổn thất (loss) để tối ưu (cập nhật trọng số) của mô hình. Tính toán tổn thất bằng cách so sánh dự đoán đầu ra của mô hình với nhãn mục tiêu thực tế.

Hàm đánh giá mô hình:

```
def eval(test_data = False):
    model.eval()
    losses = []
    correct = 0

    with torch.no_grad():
        data_loader = test_loader if test_data else valid_loader
        for data in data_loader:
            input_ids = data['input_ids'].to(device)
            attention_mask = data['attention_masks'].to(device)
            targets = data['targets'].to(device)

            outputs = model(
                input_ids=input_ids,
                attention_mask=attention_mask
            )

            _, pred = torch.max(outputs, dim=1)

            loss = criterion(outputs, targets)
            correct += torch.sum(pred == targets)
            losses.append(loss.item())

    if test_data:
        print(f'Test Accuracy: {correct.double()/len(test_loader.dataset)} Loss: {np.mean(losses)}')
        return correct.double()/len(test_loader.dataset)
    else:
        print(f'Valid Accuracy: {correct.double()/len(valid_loader.dataset)} Loss: {np.mean(losses)}')
        return correct.double()/len(valid_loader.dataset)
```

Hình 3.50. Hàm đánh giá mô hình học sâu.

Tương tự như hàm đào tạo, chỉ có sự khác biệt nhỏ trong sử dụng no_grad(), điều này vô hiệu hóa tính toán độ dốc. Điều này để giảm mức sử dụng bộ nhớ và tăng tốc độ đánh giá vì không cần đến độ dốc trong quá trình đánh giá.

Tiến hành đào tạo mô hình:

```

for fold in range(skf.n_splits):
    print(f'-----Fold: {fold+1} -----')
    train_loader, valid_loader = prepare_loaders(train_df, fold=fold)
    # model = SentimentClassifier(n_classes=7).to(device)
    model = SentimentClassifier(n_classes=2).to(device)
    criterion = nn.CrossEntropyLoss()
    # Recommendation by BERT: lr: 5e-5, 2e-5, 3e-5
    # Batchsize: 16, 32
    optimizer = AdamW(model.parameters(), lr=2e-5)

    lr_scheduler = get_linear_schedule_with_warmup(
        optimizer,
        num_warmup_steps=0,
        num_training_steps=len(train_loader)*EPOCHS
    )
    best_acc = 0
    for epoch in range(EPOCHS):
        print(f'Epoch {epoch+1}/{EPOCHS}')
        print('-'*30)

        train(model, criterion, optimizer, train_loader)
        val_acc = eval()

        if val_acc > best_acc:
            torch.save(model.state_dict(), f'phobert_fold{fold+1}.pth')
            best_acc = val_acc

```

Hình 3.51. Đào tạo mô hình học sâu.

Có các tham số quan trọng để đào tạo mô hình:

- Độ đo: xác định tồn thât cho quá trình đào tạo. Ta thường dùng CrossEntropyLoss là một lựa chọn phổ biến cho các nhiệm vụ phân loại, khi xác suất dự đoán gần với nhãn thực thì tồn thât thấp và ngược lại.
- Trình tối ưu: Tối ưu trọng số của mạng để giảm thiểu hàm mất mát được xác định trước. AdamW là một biến thể của Adam, được thiết kế để hoạt động tốt với các mô hình như BERT.
- Tốc độ học: Kiểm soát kích thước bước được sử dụng để cập nhật trọng số của mô hình trong mỗi lần lặp đào tạo. Tốc độ học nhỏ hơn (ví dụ: 2e-5, tương đương với 0,00002) dẫn đến cập nhật tham số nhỏ hơn, có khả năng dẫn đến quá trình đào tạo ổn định hơn, nhưng có thể cần nhiều lần lặp đào tạo hơn để hội tụ.

3.5. Đánh giá mô hình

Kết quả đào tạo mô hình:

Epoch 8/8

Train Accuracy: 0.8821917808219178 Loss: 0.361324658860331
Valid Accuracy: 0.775 Loss: 0.4620371063550313

Hình 3.52. Kết quả đào tạo mô hình học sâu từ một K-fold.

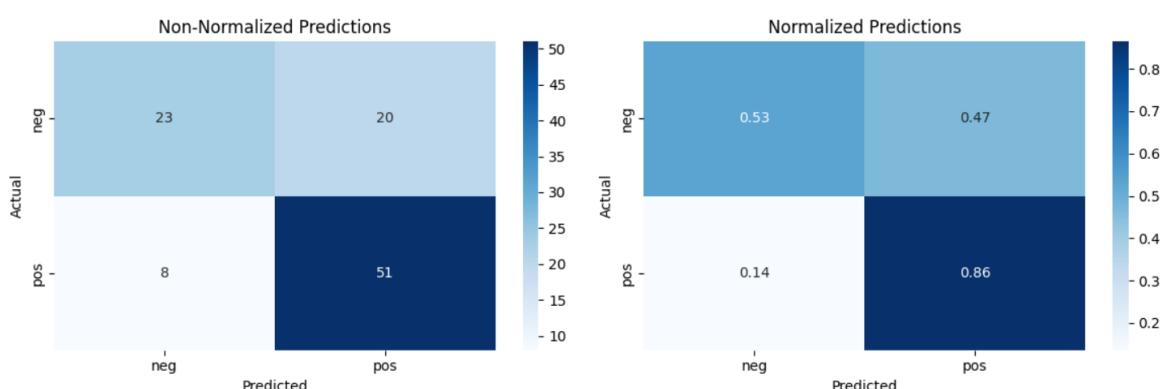
Nhận xét: dù được đào tạo trên tập dữ liệu rất ít, mô hình vẫn cho độ chính xác khá tốt, không gặp vấn đề về quá khóp dữ liệu cũng như chưa khớp dữ liệu.

Kết quả đánh giá mô hình:

	precision	recall	f1-score	support
0	0.74	0.53	0.62	43
1	0.72	0.86	0.78	59
accuracy			0.73	102
macro avg	0.73	0.70	0.70	102
weighted avg	0.73	0.73	0.72	102

Hình 3.53. Kết quả đánh giá mô hình học sâu.

Trực quan hóa kết quả đánh giá mô hình:



Hình 3.54. Trực quan hóa kết quả đánh giá mô hình.

Nhận xét kết quả đánh giá mô hình:

Lớp 0 (neg, tiêu cực) cho precision, recall, F1-score lần lượt là 0.74, 0.53, 0.62. Precision bằng 0.74, nghĩa là mô hình dự đoán là “neg” đúng khoảng 74%.

Recall bằng 0.53, nghĩa là mô hình chỉ phát hiện được khoảng 53% trường hợp thực sự là “neg”. F1-score là 0.62, cho biết bộ phân loại trung bình khá.

Tương tự, lớp 1 (pos, tích cực) cho precision, recall, F1-score lần lượt là 0.72, 0.86, 0.78. Precision bằng 0.72, nghĩa là mô hình dự đoán là “pos” đúng khoảng 72%. Recall là 0.86, nghĩa là mô hình tốt trong việc phát hiện các trường hợp thực sự là “pos” và nó đã phát hiện được khoảng 86%. F1-score là 0.78, cho thấy bộ phân loại khá tốt.

Accuracy là 0.73, mô hình đoán đúng khoảng 73% trên toàn bộ dự đoán.

Macro avg là trung bình của các giá trị precision, recall và F1-score của cả hai lớp. Trong trường hợp này, kết quả lần lượt là 0.73, 0.70, 0.70.

Weighted avg: mỗi lớp được coi trọng dựa trên tỷ lệ số lượng mẫu của lớp đó so với tổng số lượng mẫu. Các lớp có số lượng mẫu lớn hơn sẽ có trọng số lớn hơn trong tính toán. Trong trường hợp này, kết quả lần lượt là 0.73, 0.73, 0.72.

Trong kết quả, Macro avg và weighted avg khá đồng đều, điều này nghĩa là sự không cân bằng trong số lượng mẫu giữa các lớp không ảnh hưởng đáng kể đến giá trị trung bình của các số liệu đánh giá hiệu suất của mô hình.

3.6. Dự đoán với mô hình đã xây dựng được

```
def infer(text, tokenizer, max_len=120):
    encoded_review = tokenizer.encode_plus(
        text,
        max_length=max_len,
        truncation=True,
        add_special_tokens=True,
        padding='max_length',
        return_attention_mask=True,
        return_token_type_ids=False,
        return_tensors='pt',
    )

    input_ids = encoded_review['input_ids'].to(device)
    attention_mask = encoded_review['attention_mask'].to(device)

    output = model(input_ids, attention_mask)
    _, y_pred = torch.max(output, dim=1)

    print(f'Text: {text}')
    print(f'Sentiment: {class_names[y_pred]}')
    print('*'*50)
```

Hình 3.55. Hàm dự đoán dữ liệu mới.

Kết quả dự đoán trên tập dữ liệu mới:

Text: con 6 nhanh nóng máy lắm, nhất là trời nóng thế này. bác cõi lên con 7 ý.
Sentiment: pos

Text: tầm 6h được đó bác, 7 7pro giờ nó khá mát, pin cũng ổn rom gốc rất mượt.
Sentiment: pos

Text: em thấy ai cũng chê mà em dùng cũng mát, công sở nhiều khi sáng hôm trước đến trưa hôm
Sentiment: pos

Text: pixel 7 7 pro giờ rom sau 1 năm tối ưu khá tốt rồi bác, gần như không có mấy cái lỗi r
Sentiment: pos

Text: same, lỗi quá nhiệt là lý do em bán con pixel 6 pro đi. mặc dù em ko hề chơi game nhé
Sentiment: pos

Text: con đó 8a nhìn cầm vừa vặn mà bạn. màn hình phẳng nữa chắc dễ cầm nắm
Sentiment: pos

Text: ngon bổ khá rẻ bác nhé, độ mượt thì max level
Sentiment: pos

Text: con này nó bo góc nhiều nhưng cái cam lại kè 2 vạch thẳng nên thấy lạc quẻ, làm cái cá
Sentiment: neg

Text: con 5a cũng từ nhiều mà, nóng quá nên toi main, bọn 6 sang năm cứ coi chừng
Sentiment: pos

Hình 3.56. Kết quả dự đoán mô hình học sâu trên dữ liệu mới.

Nhận xét: Các kết quả được tô sáng là các dự đoán đúng với thực tế. Như vậy, có 6 trong tổng số 9 bình luận được dự đoán đúng. Ta tính được độ chính xác accuracy bằng 0.67, thấp hơn so với kết quả kiểm thử mô hình (0.73).

Như vậy, mô hình vẫn chưa dự đoán tốt trong thực tế. Để khắc phục điều này, ta cần tăng dữ liệu đầu vào, tinh chỉnh dữ liệu để cải thiện chất lượng của dữ liệu và đào tạo mô hình mới tốt hơn.

3.7. So sánh với mô hình học máy truyền thống

Để có cái nhìn khái quát hơn về độ hiệu quả của mô hình cũng như khả năng của mô hình, trong phần này tôi sẽ sử dụng mô hình học máy TF-IDF (viết tắt của từ tiếng Anh: term frequency – inverse document frequency, tạm dịch: Tần suất thuật ngữ - Nghịch đảo tần suất văn bản) kết hợp với MultinomialNB (viết tắt của từ: Multinomial Naive Bayes, thuật toán phân loại theo xác suất dựa trên định lý Bayes) để đưa ra so sánh.

Lấy dữ liệu và mã hóa bình luận thành các vectơ sử dụng TF-IDF:

```

1 import pandas as pd
2 from sklearn.feature_extraction.text import TfidfVectorizer
3
4 train_df = pd.read_csv("data/train.csv", encoding="utf-8")
5 test_df = pd.read_csv("data/test.csv", encoding="utf-8")
6
7 train_x, valid_x = train_df["comment"], test_df["comment"]
8 train_y, valid_y = train_df["label"], test_df["label"]
9
10 vectorizer = TfidfVectorizer(max_df=0.9, min_df=0.01, max_features=5000)
11 X_train = vectorizer.fit_transform(train_x)
12 X_valid = vectorizer.transform(valid_x)

```

Hình 3.57. Mã hóa bình luận với TF-IDF.

Các tham số quan trọng của TfidfVectorizer:

- max_df: Tham số này chỉ định rằng các từ xuất hiện nhiều trong các tài liệu sẽ bị bỏ qua trong quá trình vector hóa TF-IDF. Nó giúp lọc ra những từ phổ biến và ít thông tin.
- min_df: Tham số này chỉ định rằng các từ xuất hiện trong ít tài liệu sẽ bị bỏ qua trong quá trình vector hóa TF-IDF. Nó giúp loại bỏ những từ rất hiếm có thể không đóng góp nhiều cho việc phân tích.
- max_features: Tham số này giới hạn số lượng tính năng (từ) hàng đầu có điểm TF-IDF cao nhất. Điều này làm giảm số chiều của không gian đặc trưng.

Hàm xây dựng mô hình:

```

7 def multinomialBN_model(X_train, train_y, X_valid, valid_y, alpha=1.0):
8     model = MultinomialNB(alpha=alpha).fit(X_train, train_y)
9     y_pred = model.predict(X_valid)
10    (prec, recall, f1, class_size) = precision_recall_fscore_support(
11        valid_y, y_pred, average=None, labels=model.classes_
12    )
13    scores = {
14        "class_order": model.classes_,
15        "precision": prec,
16        "recall": recall,
17        "f1": f1,
18        "avg prec": np.mean(prec),
19        "avg recall": np.mean(recall),
20        "avg f1": np.mean(f1),
21    }
22    return model, scores, y_pred

```

Hình 3.58. Hàm xây dựng mô hình MultinomialNB.

Tìm alpha để chọn mô hình. Các giá trị khác nhau của alpha có thể mang lại kết quả khác nhau. Ta có thể so sánh kết quả để chọn tốt nhất [5].

```
25 models = {}
26 for alpha in [0.1, 0.2, 0.4, 0.6, 0.8, 1.0]:
27     models[alpha] = multinomialBN_model(X_train, train_y, X_valid, valid_y, alpha=alpha)
28 f1_max = max([models[alpha][1]["avg f1"] for alpha in models])
29 best_alpha, best_model, best_score, y_pred = [
30     (alpha, models[alpha][0], models[alpha][1], models[alpha][2])
31     for alpha in models
32     if models[alpha][1]["avg f1"] == f1_max
33 ][0]
34 print(
35     f"""
36     Best alpha      : {best_alpha}
37     Avg. Precision : {best_score["avg prec"]}
38     Avg. Recall    : {best_score["avg recall"]}
39     Avg. F1        : {best_score["avg f1"]}"""
40 )
41 print(
42     f"""
43     \nPer class evaluation
44     Classes       : {best_score["class_order"]}
45     Precision     : {best_score["precision"]}
46     Recall        : {best_score["recall"]}
47     F1            : {best_score["f1"]}"""
48 )
```

Hình 3.59. Tìm giá trị alpha cho mô hình MultinomialNB.

Kết quả kiểm thử mô hình học máy:

```
Best alpha      : 0.1
Avg. Precision : 0.6769400110071546
Avg. Recall    : 0.6267244777296019
Avg. F1        : 0.619235836627141
```

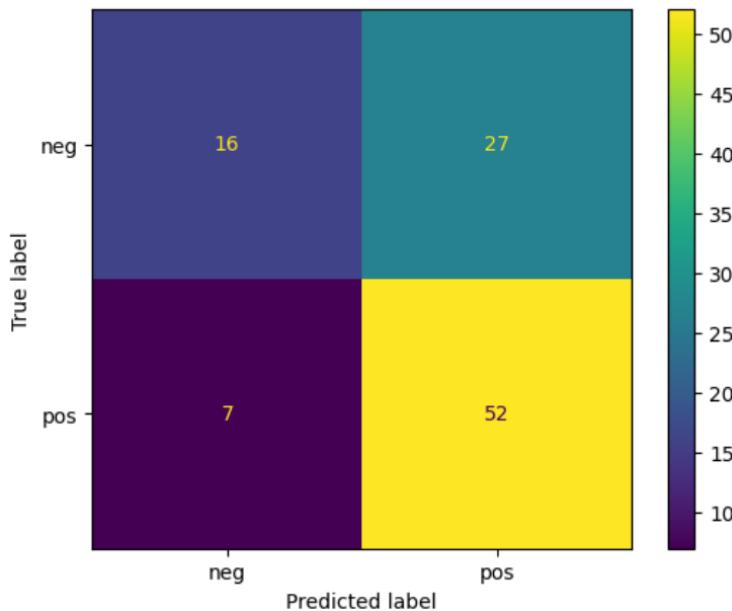
```
Per class evaluation
Classes       : ['neg' 'pos']
Precision     : [0.69565217 0.65822785]
Recall        : [0.37209302 0.88135593]
F1            : [0.48484848 0.75362319]
```

Hình 3.60. Kết quả precision, recall, F1-score của mô hình học máy.

```
1 from sklearn import metrics
2
3 metrics.accuracy_score(y_pred, valid_y)
✓ 0.0s
0.6666666666666666
```

Hình 3.61. Kết quả độ chính xác accuracy của mô hình học máy.

Trực quan hóa kết quả kiểm thử mô hình học máy:



Hình 3.62. Trực quan hóa kết quả kiểm thử mô hình học máy.

Kết quả dự đoán của mô hình trên tập dữ liệu mới:

```

1 new_samples = [
2     'Con 6 nhanh nóng máy lắm, nhất là trời nóng thế này. Bác cố lén con 7 ý.',
3     'Tầm 6h được đó bác, 7-7Pro giờ nó khá mát, pin cũng ổn + rom gốc rất mượt.'
4     'Em thấy ai cũng chê mà em dùng cũng mát, công sở nhiều khi sáng hôm trước để
5     'pixel 7 - 7 pro giờ ROM sau 1 năm tối ưu khá tốt rồi bác, gần như không có m
6     'mình dùng 7 pro lock adb, dùng 3 tháng rồi thấy vậy.',
7     'Same, lỗi quá nhiệt là lý do em bán con Pixel 6 Pro đi. Mặc dù em ko hề chơi
8     'con đó 8a nhìn cầm vừa vặn mà bạn. màn hình phẳng nữa chắc dễ cầm nắm',
9     'Ngon - bổ - khá rẻ bác nhé, độ mượt thì max level ',
10    'con này nó bo góc nhiều nhưng cái cam lại kẽ 2 vạch thẳng nên thấy lạc que,
11    'con 5a cũng tử nhiều mà, nóng quá nên toi main, bọn 6 sang năm cứ coi chừng'
12 ]
13 sample_vects = vectorizer.transform([clean(doc) for doc in new_samples])
14 print("Predicted class for samples: ", best_model.predict(sample_vects))
15 print(
16     "Probabilities: \n",
17     best_model.classes_,
18     "\n",
19     best_model.predict_proba(sample_vects),
20 )

```

Predicted class for samples: ['neg' 'pos' 'neg' 'pos' 'pos' 'pos' 'pos' 'pos' 'neg']

Hình 3.63. Kết quả dự đoán của mô hình học máy trên dữ liệu mới.

Nhận xét: Các kết quả được tô sáng là các dự đoán đúng với thực tế. Như vậy, có 6 trong tổng số 9 bình luận được dự đoán đúng. Ta tính được độ chính xác Accuracy bằng 0.67, kết quả này giống với kết quả kiểm thử của mô hình.

So sánh mô hình

Bảng 3.1. Kết quả mô hình học sâu và mô hình học máy trên tập kiểm thử.

Thông số đánh giá	Mô hình học sâu	Mô hình học máy
Accuracy	0.73	0.67
Precision	0.73	0.68
Recall	0.70	0.63
F1-score	0.70	0.62
Precision lớp 0 (neg)	0.74	0.70
Precision lớp 1 (pos)	0.72	0.66
Recall lớp 0	0.53	0.37
Recall lớp 1	0.86	0.88
F1-score lớp 0	0.62	0.48
F1-score lớp 1	0.78	0.75

Nhận xét: Như được hiển thị trong bảng trên, mô hình học sâu đã đạt được kết quả tốt hơn trên hầu hết các thông số đánh giá.

KẾT LUẬN

1. Đánh giá kết quả đạt được

Trong quá trình nghiên cứu thực hiện đồ án, tìm hiểu về bài toán xây dựng mô hình phân loại bình luận, tôi đã thu được những kinh nghiệm quý báu và đạt được những kết quả chính như sau:

- Nghiên cứu tổng quan về xử lý ngôn ngữ tự nhiên và cụ thể là ngôn ngữ tiếng Việt.
- Biết được quy trình xây dựng bài toán phân loại văn bản. Bao gồm các bước thu thập dữ liệu, tiền xử lý dữ liệu, phân tích dữ liệu và xây dựng một mô hình học sâu để giải quyết bài toán phân loại bình luận.
- Mô hình tùy chỉnh mặc dù trên tập dữ liệu rất ít nhưng đã đạt được kết quả mong đợi là 73% trên tập kiểm thử. Điều này cho thấy tinh chỉnh mô hình học sâu mang lại kết quả tốt, hữu ích với các miền chuyên ngành, những lĩnh vực nhỏ (ngách) có dữ liệu khiêm tốn.
- Có những nhận xét, thông kê và tìm hiểu về diễn đàn VOZ và những đánh giá hữu ích về dòng điện thoại Google Pixel.

2. Hướng phát triển của đề tài

Trong tương lai, tôi sẽ tiếp tục nghiên cứu tìm hiểu để giải quyết một số vấn đề, mà trong thời gian thực hiện đồ án chưa làm được:

- Tìm hiểu thêm về các công cụ và kỹ thuật thu thập dữ liệu khác.
- Tiếp tục gán nhãn bình luận để tăng khối lượng dữ liệu và cải thiện được kết quả mô hình phân loại.
- Nghiên cứu thêm các phương pháp tiền xử lý dữ liệu để cải thiện chất lượng dữ liệu.
- So sánh với các mô hình phân loại văn bản khác để chọn lựa mô hình có kết quả tốt nhất.
- Nghiên cứu về bài toán phân loại dựa trên khía cạnh.

TÀI LIỆU THAM KHẢO

- [1] H. Phê, Từ điển tiếng Việt, Hà Nội: NXB Đà Nẵng, 2003.
- [2] Đ. Điện, Giáo trình "Xử lý ngôn ngữ tự nhiên", NXB Đại học Quốc gia - HCM, 2006.
- [3] Mai Ngọc Chù, Vũ Đức Nghiêm, Hoàng Trọng Phiên, Cơ sở ngôn ngữ học và tiếng Việt, NXB Giáo dục.
- [4] "Đặc điểm tiếng Việt," 21 11 2010. [Online]. Available: <https://vnlp.net/ti%E1%BA%BFng-vi%E1%BB%87t-c%C6%A1-b%E1%BA%A3n/d%E1%BA%B7c-di%E1%BB%83m-ti%E1%BA%BFng-vi%E1%BB%87t/>.
- [5] J. Singh, Natural Language Processing in the Real World, CRC Press, 2023.
- [6] Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana, Practical Natural Language Processing, O'Reilly Media, Inc., 2020.
- [7] T. Ganegedara, Natural Language Processing with TensorFlow, Packt Publishing, 2022.
- [8] S. Ravichandiran, Getting Started with Google BERT, Packt Publishing, 2021.
- [9] "Các phương pháp đánh giá một hệ thống phân lớp," [Online]. Available: <https://machinelearningcoban.com/2017/08/31/evaluation/>.
- [10] A. Chapagain, Hands-On Web Scraping with Python, Packt Publishing.
- [11] "đánh giá - Google pixel - xuất sắc ở tầm giá 4->7 triệu (trải nghiệm sử dụng, cách mua)," Công ty TNHH Thật Vi Diệu, [Online]. Available: <https://voz.vn/t/google-pixel-xuat-sac-o-tam-gia-4-7-trieu-trai-nghiem-su-dung-cach-mua.122469/>.
- [12] "Website Traffic - Check and Analyze Any Website | Similarweb," Similarweb LTD, [Online]. Available: <https://www.similarweb.com/>.
- [13] "Google Pixel 4a 5G - Full phone specifications," [Online]. Available: https://www.gsmarena.com/google_pixel_4a_5g-10385.php.