1. ĐẠI HỌC QUỐC GIA HÀ NỘI
2. TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



1. Họ Và Tên Sinh Viên:
2. Phan Lương Huân
3. Nguyễn Việt Hà
4. Lê Tuấn Minh
5. DỰ đOÁN ĐÁNH GIÁ CỦA KHÁCH HÀNG TỪ CÁC BÌNH LUẬN TRÊN CÁC TRANG THƯƠNG MẠI ĐIỆN TỬ TIẾNG VIỆT
6. Báo cáo học phần môn dự án
7. Ngành: Công nghệ thông tin
8. HÀ NỘI – 2021

1. ĐẠI HỌC QUỐC GIA HÀ NỘI
2. TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
3. Họ Và Tên Sinh Viên:   
   1. Phan Lương Huân  
   2. Nguyễn Việt Hà  
   3. Lê Tuấn Minh

**DỰ ĐOÁN ĐÁNH GIÁ CỦA KHÁCH HÀNG TỪ CÁC BÌNH LUẬN TRÊN CÁC TRANG THƯƠNG MẠI ĐIỆN TỬ TIẾNG VIỆT**

1. Báo cáo học phần môn dự án
2. Ngành: Công nghệ thông tin

**Cán bộ hướng dẫn: ThS. Cấn Duy Cát**

1. HÀ NỘI – 2021

Tóm tắt

Hiện nay việc mua bán hàng online đang phát triển với tốc độ rất nhanh và nó là một xu thế tất yếu của cuộc sống. Với những mặt tích cực mà nó mang đến là sự tiện lợi, nhanh chóng, mọi lúc, mọi nơi, v.v… thì vẫn có đó những mặt hạn chế. Ví dụ như việc chúng ta khó mà có thể an tâm về sản phẩm có đúng chất lượng như mô tả, quảng cáo trước đó không để tin tưởng và chọn mua sản phẩm đó. Người dùng hay người cung cấp sản phẩm cũng không thể đọc hết các bình luận để dựa vào đó có cái nhìn tổng quan về thái độ, ý kiến của những người đã mua sản phẩm, từ đó có thể đưa ra quyết định có nên mua hay không, có nên cải thiện thêm điều gì về sản phẩm của mình nữa không. Từ ý tưởng này, nhóm đã cố gắng tạo ra một hệ thống sử dụng AI để phân loại sắc thái bình luận của người dùng. Dựa vào hệ thống phân loại này, người dùng có thể thống kê phân loại được những bình luận tích cực, tiêu cực liên quan đến sản phẩm.

Báo cáo sử dụng dữ liệu được crawls từ trang tiki.vn, shopee.vn

**Từ khoá**: *Support Vector Machine, Long Short-Term Memory, Recurrent Neural Network.*

Lời cảm ơn

Để hoàn thành được bài báo cáo này, em xin gửi lời cảm ơn đến thầy Cấn Duy Cát đã tận tình hướng dẫn, chỉ bảo em trong suốt quá trình học tập và nghiên cứu.

Em xin gửi đến toàn bộ quý thầy, cô giáo trong khoa Công nghệ thông tin nói riêng và trường Đại học Công Nghệ - Đại Học Quốc Gia Hà Nội nói chung lời cảm ơn chân thành. Em xin kính chúc quý thầy, cô dồi dào sức khỏe và thành công trong sự nghiệp cao quý.

Lời cam đoan

Em xin cam đoan báo cáo này do chính em và các bạn trong nhóm thực hiện. Em xác nhận rằng bài báo cáo này là của riêng em và các bạn trong nhóm, ngoại trừ trường hợp trích dẫn tham khảo từ các nguồn khác nhau. Mọi đóng góp của các tác giả khác được trích dẫn trong báo cáo này đã được chính tác giả cho phép và đã liệt kê rõ ở cuối bài báo cáo.

Em xác nhận rằng bài báo cáo của em không vi phạm bản quyền của bất kỳ ai cũng như không vi phạm bất kỳ quyền sở hữu nào. Bất kỳ ý tưởng, nghiên cứu, trích dẫn hoặc bất kỳ tài liệu nào khác được đưa vào bài báo cáo đã được các chủ sở hữu cho phép bằng văn bản và em có đầy đủ quyền tác giả để cải thiện các tài liệu này.

Sinh viên thực hiện

**Phan Lương Huân**

Mục lục

[Tóm tắt iii](#_Toc74184140)

[Lời cảm ơn iv](#_Toc74184141)

[Lời cam đoan v](#_Toc74184142)

[Mục lục vi](#_Toc74184143)

[Danh mục từ viết tắt viii](#_Toc74184144)

[Danh mục hình ảnh ix](#_Toc74184145)

[Danh mục bảng biểu x](#_Toc74184146)

[Chương 1. Giới thiệu 1](#_Toc74184147)

[1.1. Động lực 1](#_Toc74184148)

[1.2. Đặt vấn đề 1](#_Toc74184149)

[1.3. Khó khăn và thử thách 2](#_Toc74184150)

[1.4. Cách tiếp cận 2](#_Toc74184151)

[1.5. Quy mô và giới hạn 3](#_Toc74184152)

[Chương 2. Cơ sở lý thuyết 4](#_Toc74184153)

[2.1. Khai phá dữ liệu 4](#_Toc74184154)

[2.1.1. Khái niệm 4](#_Toc74184155)

[2.1.2. Quá trình khai phá dữ liệu 4](#_Toc74184156)

[2.1.3. Các bài toán của khái phá dữ liệu 6](#_Toc74184157)

[2.2. Các mô hình được sử dụng 6](#_Toc74184158)

[2.2.1. Support Vector Machine (SVM) 7](#_Toc74184159)

[2.2.2. Recurrent Neural Network (RNN) 8](#_Toc74184160)

[2.2.3. Long Short Term Memory Networks (Mạng LSTM) 9](#_Toc74184161)

[2.3. Phương pháp phân chia dữ liệu 10](#_Toc74184162)

[2.3.1. Phân chia ngẫu nhiên 10](#_Toc74184163)

[2.3.2. Xác thực chéo 10](#_Toc74184164)

[Chương 3. Thực nghiệm và kết quả 12](#_Toc74184165)

[3.1. Cài đặt và cấu hình 12](#_Toc74184166)

[3.1.1. Triển khai mô hình 12](#_Toc74184167)

[3.1.2. Môi trường huấn luyện và kiểm thử 12](#_Toc74184168)

[3.2. Tổng quan về mô hình 13](#_Toc74184169)

[3.3. Chuẩn bị dữ liệu 13](#_Toc74184170)

[3.4. Word Embedding 15](#_Toc74184171)

[3.4.1. Term Frequency Inverse Document Frequency 15](#_Toc74184172)

[3.4.2. PhoBERT 15](#_Toc74184173)

[3.5. Kết quả thực nghiệm và đánh giá 16](#_Toc74184174)

[3.5.1. Đánh giá mô hình phân lớp 16](#_Toc74184175)

[3.5.2. Kết quả đánh giá mô hình 18](#_Toc74184176)

[Kết luận 19](#_Toc74184177)

[Tài liệu tham khảo 20](#_Toc74184178)

[Phân công công việc 21](#_Toc74184179)

Danh mục từ viết tắt

BERT Bidirectional Encoder Representations from Transformers

KDD Knowledge Discovery in Database

LSTM Long Short-Term Memory

RNN Recurrent Neural Network

SVC Support Vector Classification

SVM Support Vector Machine

TF-IDF Term Frequency-Inverse Document Frequency

Danh mục hình ảnh

[Hình 2.1. Quá trình khám phá tri thức. 5](#_Toc74157169)

[Hình 2.2. Các bài toán của khái phá dữ liệu. 6](#_Toc74157170)

[Hình 2.3. Dữ liệu được phân lớp với các siêu phẳng. 7](#_Toc74157171)

[Hình 2.4. Sử dụng kernel xây dựng siêu phẳng. 8](#_Toc74157172)

[Hình 2.5. Module lặp lại trong một mạng LSTM chứa 4 lớp tương tác. 9](#_Toc74157173)

[Hình 2.6. Kỹ thuật xác thực chéo. 11](#_Toc74157174)

[Hình 3.1. Mô hình hóa quá trình thực hiện 13](#_Toc74157175)

[Hình 3.2. Ma trận lỗi. 16](#_Toc74157186)

Danh mục bảng biểu

[Bảng 3.1. Số lượng các nhãn trong bộ dữ liệu. 14](#_Toc74157187)

[Bảng 3.2. Ví dụ các bước trong phần Preprocessing. 14](#_Toc74157188)

[Bảng 3.3. Công thức tính micro-average và macro-average. 17](#_Toc74157189)

[Bảng 3.4. Kết quả khi sử dụng TF-IDF với các mô hình. 18](#_Toc74157191)

[Bảng 3.5. Kết quả khi sử dụng PhoBERT với các mô hình. 18](#_Toc74157241)

# Giới thiệu

## Động lực

Thu thập thông tin phản hồi từ khách hàng là một phương pháp giúp cho các doanh nghiệp biết được những điểm mạnh, yếu trong sản phẩm, dịch vụ của mình. Từ đó nhanh chóng nắm bắt được tâm lý, cảm súc của khách hàng để mang lại cho họ sản phẩm, dịch vụ tốt hơn theo nhu cầu.

Ngày nay, với sự phát triển của Internet, đặc biệt là sự bùng nổ của mạng xã hội và sự phát triển như một điều tất yếu của thương mại điện tử, ... đã cho phép mọi người không chỉ chia sẻ thông tin trên đó mà còn thể hiện thái độ, quan điểm của mình đối với các sản phẩm, dịch vụ và các vấn đề xã hội khác. Vì vậy mà Internet đã trở lên vô cùng quan trọng và là nguồn cung cấp một lượng thông tin vô cùng lớn và quan trọng.

Thông qua những dữ liệu được cung cấp qua Internet:

* Người dùng sử dụng nó để tìm kiếm, tham khảo trước khi đưa ra quyết định về sử dụng một sản phẩm hay dịch vụ nào đó.
* Các nhà cung cấp dịch vụ cũng có thể sử dụng những nguồn thông tin này để đánh giá về sản phẩm của mình, từ đó có thể đưa ra những cải tiến phù hợp hơn với người dùng, mang lại lợi nhuận cao hơn, tránh các rủi ro đáng tiếc xảy ra. Đặc biệt, khi 1 doanh nghiệp có 1 sản phẩm mới ra mắt thị trường thì việc lấy ý kiến phản hồi là vô cùng cần thiết.
* Các cơ quan chức năng có thể sử dụng những thông tin này để tìm hiểu xem quan điểm và thái độ của cộng đồng để có thể kịp thời sửa đổi, ban hành các chính sách cho hợp lý hơn.

## Đặt vấn đề

Một câu hỏi đặt ra đối với những trang thương mại điện tử là thái độ, đánh giá và quan điểm của người dùng với sản phẩm họ cung cấp là như thế nào để có thể đưa ra những chỉnh sửa, cải tiến cho hệ thống thích hợp hơn với người dùng. Bên cạnh đó, người dùng cũng có thể dự vào những phản hồi của người mua để quyết định, lựa chọn có nên mua sản phẩm đó hay không. Để trả lời cho những câu hỏi đó, cũng như giải quyết tất cả những vấn đề trên không phải là một chuyện đơn giản. Một cách truyền thống là có thể đi hỏi từng người sử dụng một, sử dụng phiếu đánh giá, ... Nhưng thực tế cho thấy rằng những phương pháp này là không khả thi hoặc không hiệu quả. Các công ty nghiên cứu thị trường hiện này vẫn thường sử dụng các phương pháp truyền thống này tuy nhiên độ tin cậy không cao (do nhân viên tự làm giả dữ liệu) và độ tính cập nhật khá thấp (10-20 ngày cho thu thập thông tin, 10-20 ngày cho phân tích dữ liệu).

Để trả lời câu hỏi trên một cách hiệu quả và nhanh chóng thì cần phải tận dụng phân tích được chính những thông tin mà người dùng để lại qua internet như các bình luận, đánh giá, bài chia sẻ, ... Vì đó, dẫn tới việc phân loại các bình luận theo quan điểm tích cực, tiêu cực hay trung tính.

Sau đây, nhóm sẽ cùng giải quyết bài toán phân loại sắc thái bình luận của người dùng bằng việc sử dụng bộ dữ liệu theo thang đánh giá 5 sao và cũng là bộ dữ liệu đó nhưng được gán nhăn thành 3 loại: Tích cực(Positive), Tiêu cực(Negative) và Trung tính(Neutral)

## Khó khăn và thử thách

Không giống như việc phân loại các văn bản, bài viết theo chủ đề, việc phân loại các bình luận gặp nhiều khó khăn trong việc xử lý dữ liệu đầu vào. Việc thu thập dữ liệu từ các trang thương mại điện tử gặp nhiều vấn đề khi bộ dữ liệu có thể khiến việc huấn luyện và kết quả thu được không được như mong đợi. Các vấn đề liên quan đến bộ dữ liệu như sau:

* Sai chính tả: Các bình luận trên mạng thường dùng nhiều từ nóng, từ viết tắt hay viết không dấu hoặc dấu đánh lộn xộn (Ví dụ: k, ko, k0 tương đương với ‘không’ hay ‘bt’ tức là ‘bình thường’). Bên cạnh đó, xuất hiện nhiều trường hợp người dùng viết láy âm tiết (Ví dụ như: Ủng hộ shopppp nhaaaa !!!!!). Ngoài ra, các bình luận thường ngắn, một số câu bình luận chỉ có một vài từ hay một vài chữ.
* Mất cân bằng: Đa phần các bình luận lấy từ các trang thương mại điện từ lớn như shopee hay tiki đều là các bình luận được đánh giá 5 sao. Cụ thể là các bình luận đánh giá 5 sao chiếm đến hơn 90%, còn lại các bình luận được đánh giá 1, 2, 3 sao thì không nhiều vì đa phần đều là các sản phẩm tốt.

## Cách tiếp cận

Giải pháp sẽ tập trung vào sử dụng các phương pháp và thuật toán học máy nhằm phân loại sắc thái của các bình luận. Đầu tiên, nhiều cách phân loại sẽ được phân tích và so sánh với nhau về mặt lý thuyết để chọn ra một số cách phù hợp nhất với bài toán được đặt ra. Tiếp theo, các thuật toán đã chọn sẽ được áp dụng trên tập dữ liệu để có được kết quả đánh giá. Trong quá trình lặp đi lặp lại này, các chỉ số và thuật toán được sử dụng sẽ liên tục được thay đổi và tối ưu hóa để đạt được kết quả tốt nhất có thể. Cuối cùng, quá trình và kết quả sẽ được đánh giá và so sánh với nhau.

## Quy mô và giới hạn

* Báo cáo sẽ tập trung vào lý thuyết phương pháp học máy được sử dụng, cũng như những kết quả đạt được trong đánh giá mô hình. Các chi tiết về lập trình và vận hành sẽ bị bỏ qua.
* Các phần dữ liệu chỉ được lấy từ các bình luận của một số nhóm sản phẩm trên một nền tảng nhất định nên vẫn còn nhiều cách có thể phát triển thêm.
* Một số chi tiết khác có liên quan đến bình luận như thông tin sản phẩm, thông tin người viết, thời gian đăng sẽ không được dùng trong việc huấn luyện mô hình.
* Dữ liệu đầu vào chỉ bao gồm các bình luận tiếng Việt trên một số trang thương mại điện tử của Việt Nam.

Báo cáo gồm có 3 chương. Trong chương tiếp theo (chương 2), báo cáo sẽ tập trung làm rõ cơ sở lý thuyết để giải quyết vấn đề này. Tại chương 3, chương cuối cùng sẽ là phần thực nghiệm và kết quả thu được sau quá trình thực hiện.

# Cơ sở lý thuyết

Trong chương này, chúng ta sẽ cùng tìm hiểu về khai phá dữ liệu và các mô hình được sử dụng để giải quyết bài toán đã đặt ra. Phần 2.1 cung cấp một cái nhìn tổng quan về những khái niệm cơ bản, quy trình và các bài toán trong khai phá dữ liệu. Phần 2.2 em sẽ giới thiệu về các mô hình được sử dụng trong hệ thống phân loại sắc thái bình luận này.

## Khai phá dữ liệu

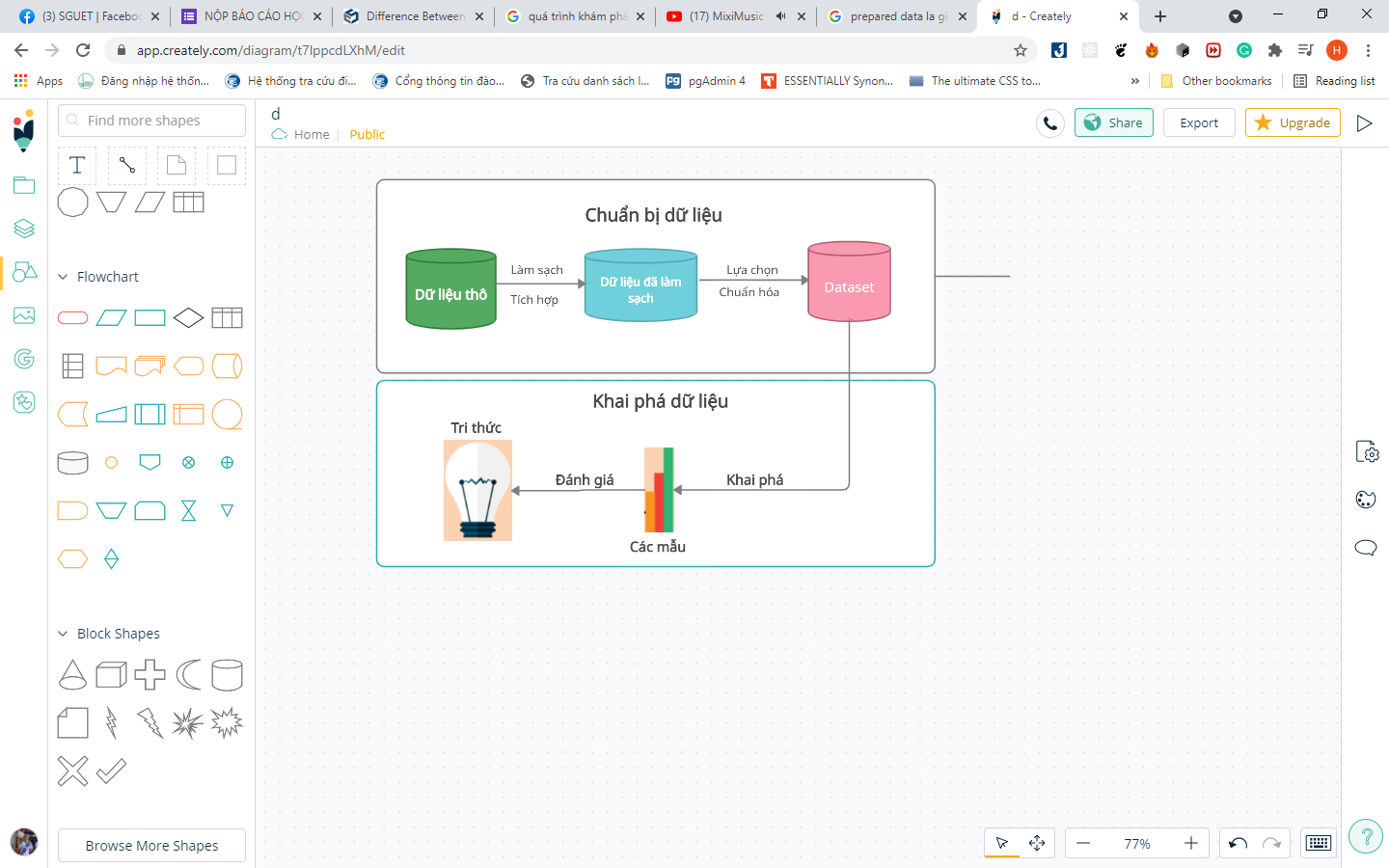
Kỷ nguyên của dữ liệu lớn, đa dạng và phức hợp. Ngày nay, số lượng người dùng các thiết bị thông minh như điện thoại, máy tính bảng, laptop, … có kết nối internet để tìm kiếm thông tin, giải trí, … có tốc độ gia tăng nhanh chóng. Hơn nữa, sự xuất hiện của IoT (Internet of Things) kết nối giữa các thiết bị với nhau bằng internet, cho phép trao đổi, truyền tải dữ liệu. Chính vì vậy mà khối lượng dữ liệu ngày càng lớn, tốc độ bùng nổ dữ liệu ngày càng nhanh. Chúng ta có thể làm gì với khối lượng dữ liệu đó? Chúng có đem lại giá trị, lợi ích gì không? Khai phá dữ liệu (Data mining) sẽ giúp ta hiểu được giá trị của dữ liệu.

### Khái niệm

Khai phá dữ liệu (Data mining) là quá trình tính toán để tìm ra các mẫu (patterns) trong các bộ dữ liệu lớn dựa vào học máy, thống kê và các hệ thống cơ sở dữ liệu. Đây là một lĩnh vực liên ngành của khoa học máy tính. Mục tiêu tổng thể của quá trình khai phá dữ liệu là trích xuất thông tin từ một bộ dữ liệu và chuyển nó thành một cấu trúc dễ hiểu để sử dụng tiếp.

### Quá trình khai phá dữ liệu

Khai phá dữ liệu (Data mining) là một phần quan trọng trong quá trình KDD (Knowledge Discovery in Database) – Khám phá tri thức trong Cơ sở dữ liệu. Quá trình khai phá tri thức bao gồm hai giai đoạn chính là chuẩn bị dữ liệu (Data preparation) và khai phá dữ liệu (Data mining).



Quá trình khám phá tri thức.

Quá trình chuẩn bị dữ liệu:

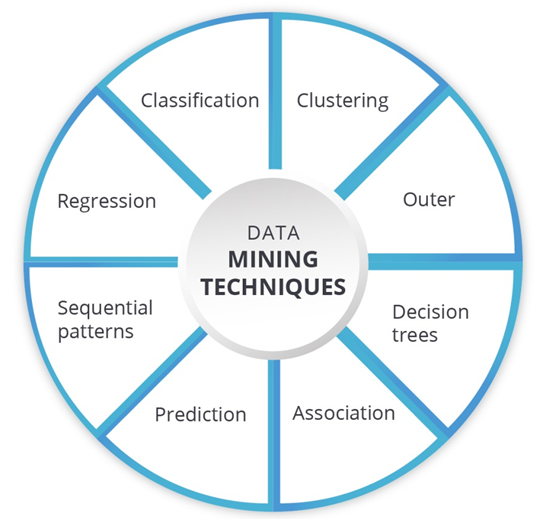
* Thu thập dữ liệu: Dữ liệu được thu thập theo mục đích đề ra.
* Làm sạch dữ liệu (Data cleaning): Mục đích là loại bỏ các dữ liệu “nhiễu” (noise), những dữ liệu không cần thiết, không liên quan hay không đầy đủ thông tin (missing value)
* Tích hợp dữ liệu (Data integration): Mục đích của việc làm này là để tích hợp dữ liệu từ nhiều nguồn khác nhau vào một hệ cơ sở dữ liệu, một nguồn dữ liệu nhất định, … Đây là quá trình phức tạp, thường sử dụng “metadata” hay còn gọi là “siêu dữ liệu” nhằm giảm bớt sai sót trong quá trình tích hợp.
* Lựa chọn dữ liệu (Data selection): Chọn ra những dữ liệu, thuộc tính có thể có tiềm năng, có giá trị phân tích và loại bỏ dữ liệu, thuộc tính dư thừa, trùng lặp.
* Chuyển đổi dữ liệu (Data transformation): Dữ liệu ban đầu thường luôn là dữ liệu thô chưa được chuẩn hóa nên việc chuyển đổi dữ liệu, khái quát và tổng hợp dữ liệu theo một định dạng phù hợp là việc làm vô cùng quan trọng và cần thiết.

Quá trình khai phá dữ liệu:

* Xây dựng mô hình học máy: Sau khi dữ liệu được làm sạch, lựa chọn mô hình học máy phù hợp với dữ liệu, chạy mô hình với bộ dữ liệu đó và đánh giá mô hình.
* Tối ưu mô hình: Mục đích là đưa ra được một mô hình tốt nhất.
* Đánh giá mẫu dữ liệu: Xây dựng các giả thuyết và tiến hành kiểm định dựa trên mức độ tin cậy. Nếu kết quả phân tích không hợp lý, có sai sót, cần quay lại các bước trước, còn nếu kết quả phân tích đã hợp lý, độ tin cậy cao thì đưa ra mẫu thông tin tri thức.
* Biểu diễn tri thức: Đưa ra các tri thức, thông tin hữu ích đến mọi người.

### Các bài toán của khái phá dữ liệu

* Phân lớp
* Phân cụm
* Phân tích tìm ngoại lệ
* Cây quyết định
* Luật kết hợp
* Phương pháp dự đoán
* Khai phá mẫu
* Hồi quy



Các bài toán của khái phá dữ liệu.

## Các mô hình được sử dụng

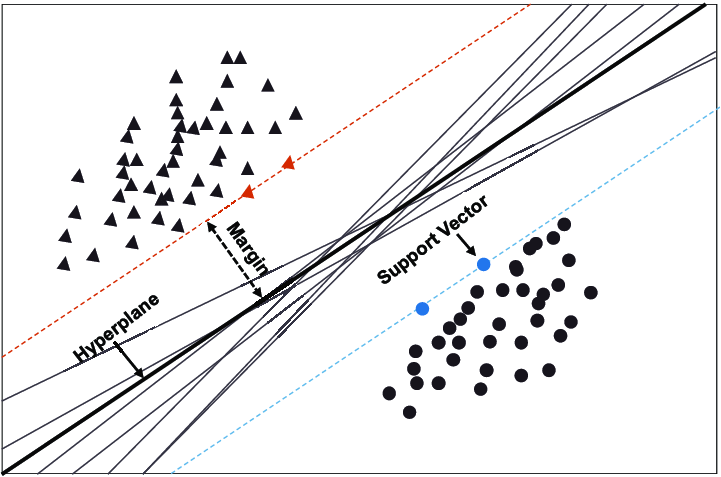
Trong phần này, em sẽ giới thiệu một cách tổng quan về các mô hình được sử dụng trong báo cáo này là: Support Vector Machine (Phần 2.2.1) và Long Short-Term Memory (Phần 2.2.2)

### Support Vector Machine (SVM)

Máy vector hỗ trợ (SVM) là 1 thuật toán học máy thuộc loại học có giám sát, nghĩa là nó tạo ra một hàm sẽ ánh xạ một đầu vào nhất định với một đầu ra. SVM nhận dữ liệu đầu vào và phân loại chúng vào hai lớp khác nhau. Do đó, SVM là một thuật toán phân loại nhị phân.

Mục tiêu của máy vector hỗ trợ là xây dựng một siêu phẳng, sau đó sử dụng nó để phân loại dữ liệu.

SVM xuất phát từ bộ phân loại lề tối đa. Bộ phân loại lề tối đa sử dụng khoảng cách từ ranh giới nhất định để phận loại đầu vào đến 2 điểm dữ liệu gần nhất tương ứng với 2 phân lớp. Khoảng cách đó được gọi là lề (margin). Hai điểm gần siêu phẳng nhất của 2 lớp được gọi là Vector hỗ trợ. Trong không gian hai chiều, ranh giới được coi là một đường thẳng. Trong không gian ba chiều, nó là một mặt phẳng, nhưng sau đó, nó trở nên khó xác định trong không gian nhiều chiều. Ranh giới này có thể coi là một siêu phẳng (hyperplane). Để có thể phân loại một cách tốt nhất thì ranh giới này phải nằm càng xa các điểm dữ liệu của tất cả các lớp, tức là lề (margin) càng to thì phân lớp càng hiệu quả.



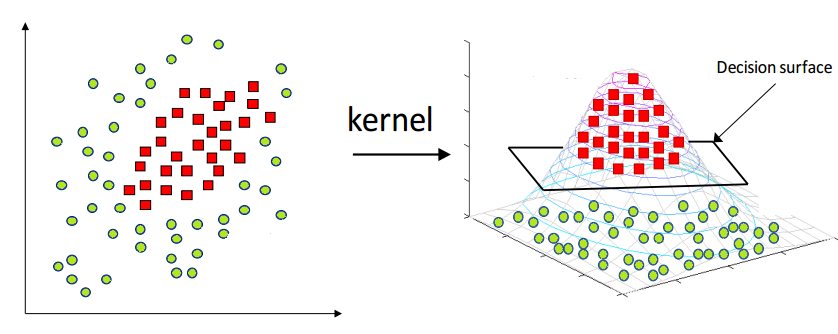
Dữ liệu được phân lớp với các siêu phẳng.

Trong hình trên, siêu phẳng tốt nhất là đường thẳng được tô đậm.

Bộ phận loại lề tối đa hoạt động ở một mức độ nhất định. Nếu tập dữ liệu không thể phân tách được bằng siêu phẳng, không thể sử dụng được tập dữ liệu này. Lúc này, hãy xem xét một bộ phân loại không hoàn hảo, nhưng nó có thể hoạt động một phần hoặc hầu hết thời gian.

Bằng cách sử dụng bộ phân loại không hoàn hảo, có thể xử lý hầu hết các quan sát và đưa ra mức độ thích hợp ứng với mô hình khi nó được hiển thị với dữ liệu mới. Sự phát triển này của bộ phân loại lề tối đa được gọi là bộ phân loại vector hỗ trợ (SVC – Support Vector Classifier). Thay vì phân loại chính xác, SVC cho phép một số quan sát nằm ở phía sai của lề và/hoặc siêu phẳng. Cần phải tìm giải pháp cho vấn đề tối ưu hóa sử dụng tham số điều chỉnh, độ rộng của lề và biến slack. Tham số điều chỉnh được sử dụng để kiểm soát sự cân bằng phương sai lệch. Biến slack cho phép các điểm dữ liệu về phía lề hoặc siêu phẳng. Các giá trị của slack cho biết về hoạt động của một điểm dữ liệu nhất định. Nếu biến slack của một điểm dữ liệu nhất định bằng 0 thì điểm đó nằm ở phía bên phải của lề. Nếu biến slack lớn hơn 0 nhưng nhỏ hơn 1, điểm dữ liệu nằm ở phía bên trái của lề và bên phải của siêu phẳng. Nếu biến slack lớn hơn 1, điểm dữ liệu nằm ở phía trái của siêu phẳng.

SVM xây dựng dựa trên sự tối ưu hóa trong Bộ phân loại vector hỗ trợ bằng cách tăng không gian đặc trưng sử dụng kernel (hạt nhân). Kernel tương tự như tối ưu hóa trước đó. Kernel tính toán các sản phẩm bên trong giữa tất cả các đầu ra của tất cả các cặp điểm dữ liệu. Bằng cách sử dụng kernel, thuật toán có thể hiệu quả hơn nhiều.



Sử dụng kernel xây dựng siêu phẳng.

Có nhiều loại kernel như RBF kernel (nhân RBF), linear kernel (nhân tuyến tính), polynomial kernel (nhân đa thức), sigmoid kernel.

### Recurrent Neural Network (RNN)

Ý tưởng chính của RNN là sử dụng chuỗi các thông tin. Trong các mạng nơ-ron truyền thống tất cả các đầu vào và cả đầu ra là độc lập với nhau. Tức là chúng không liên kết thành chuỗi với nhau. Nhưng các mô hình này không phù hợp trong rất nhiều bài toán. Ví dụ, nếu muốn đoán từ xuất hiện tiếp theo trong một câu thì ta cũng cần biết các từ xuất hiện trước đó. RNN được gọi là hồi quy (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó.

Trong loại mạng thần king cơ bản, tất cả các đầu vào và đầu ra nằm độc lập với nhau. Mô hình này không thích hợp để sử dụng trên các loại dữ liệu yêu cầu về thứ tự như các từ trong một câu, các khung hình trong một video, hồ sơ bệnh án… RNN được ra đời để giải quyết những bài toán đó.

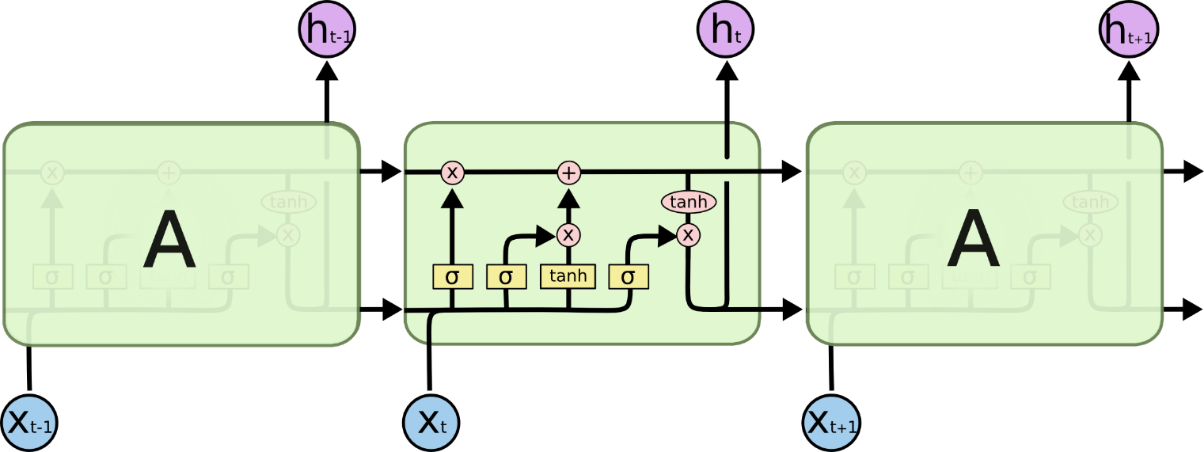
Trong một mạng thần kinh hồi quy, mỗi lớp ẩn sẽ được tính toán bằng cách kết hợp đầu vào với một lớp ẩn trước đó bằng một hàm bao gồm có trọng số và độ mất mát của mỗi tầng trước đó. Như vậy kết quả gần sẽ được "nhớ", giúp tăng độ chính xác cho mỗi bước tiếp theo, và các kết quả xa sẽ bị "quên".

### Long Short Term Memory Networks (Mạng LSTM)

Mạng bộ nhớ dài ngắn (Long Short Term Memory networks) là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa. LSTM được giới thiệu bởi [Hochreiter & Schmidhuber (1997)](http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf), sau đó đã được cải tiến và trở nên hiệu quả với nhiều bài toán khác nhau.

LSTM được thiết kế để giải quyết vấn đề phụ thuộc xa (long-term dependency) của RNN. Trong RNN, việc nhớ thông tin càng xa thì việc học sẽ càng trở lên khó khăn. Ở mạng LSTM, việc nhớ thông tin là một đặc tính mặc định, tức là không cần phải huấn luyện để có thể nhớ được. Ngay trong cấu trúc của nó đã có thể ghi nhớ được thông tin mà không cần bất kì can thiệp nào như RNN.

Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại. LSTM cũng có dạng như vậy nhưng thay vì chỉ có một tầng mạng nơ-ron như mạng RNN chuẩn, LSTM có tới 4 tầng tương tác với nhau như trong hình 2.5.

[](https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-chain.png)

Module lặp lại trong một mạng LSTM chứa 4 lớp tương tác.

## Phương pháp phân chia dữ liệu

### Phân chia ngẫu nhiên

Phân chia dữ liệu thành 2 tập dữ liệu độc lập: tập dữ liệu huấn luyện và tập dữ liệu kiểm thử.

Tập dữ liệu huấn luyện gồm 2 phần:

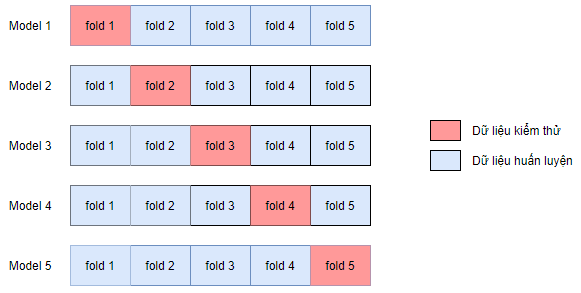
* Training set: là tập hợp gồm những dữ liệu đã có nhãn, được đưa vào để huấn luyện, đào tạo mô hình.
* Validation set: cũng là tập hợp gồm những dữ liệu có nhãn nhưng nhãn này không được sử dụng để nhằm mục đích đánh giá hiệu suất của mô hình sau khi huấn luyện, giúp thử nghiệm để tinh chỉnh các tham số của mô hình, từ đó chọn mô hình hiệu quả nhất. Có những mô hình phân loại không cần đến Validation set.

Thông thường, tỷ lệ phân chia cho training set là 80% và validation test là 20%.

Tập dữ liệu kiểm thử là tập hợp gồm những dữ liệu chưa được gán nhãn, được sử dụng để đánh giá độ hiệu quả của mô hình sau quá trình huấn luyện.

### Xác thực chéo

Xác thực chéo là kỹ thuật phân chia tập dữ liệu ban đầu thành tập dữ liệu huấn luyện (training data) – và tập dữ liệu kiểm thử (test data). Có nhiều phương pháp để phân chia tập dữ liệu nhưng phương pháp được sử dụng phổ biến nhất là K – fold. Trong K-fold, bộ dữ liệu ban đầu được chia đều thành k phần, mỗi phần là một “fold” có kích cỡ bằng nhau. Một chuỗi các mô hình sẽ được đào tạo. Mô hình đầu tiên được đào tạo sử dụng fold đầu tiên làm tập dữ liệu kiểm thử, các fold còn lại làm tập dữ liệu huấn luyện. Mô hình thứ hai sử dụng fold thứ hai làm tập dữ liệu kiểm thử và các fold còn lại làm tập dữ liệu huấn luyện. Tương tự và lặp lại, mô hình thứ k sử dụng fold thứ k làm tập kiểm thử và các tập còn lại làm tập huấn luyện.



Kỹ thuật xác thực chéo.

Như vậy, mọi điểm dữ liệu đều sẽ nằm trong tập kiểm thử một lần và được ở trong tập huấn luyện (k – 1) lần. Và điều này cũng làm xáo trộn các điểm dữ liệu trong tập dữ liệu ban đầu. Vì vậy, đối với dữ liệu chuỗi thời gian, phương pháp này không thể sử dụng do tính chất của chuỗi thời gian là liên tục và không thể bị xáo trộn được.

# Thực nghiệm và kết quả

## Cài đặt và cấu hình

### Triển khai mô hình

Mô hình được xây dựng sử dụng Python phiên bản 3.9 và TensorFlow phiên bản 2.5. TensorFlow là nền tảng open-source và hoàn toàn miễn phí được thiết kế bởi Google Brain Team cho các tác vụ trong học máy. Nó có một hệ sinh thái toàn diện, linh hoạt gồm các công cụ, thư viện và nhiều tài nguyên mở được sử dụng trong nhiều tác vụ học máy hiện đại nhất hiện nay. TensorFlow cũng có thể sử dụng trong các nghiên cứu và báo cáo khoa học.

Các gói Python khác được sử dụng trong dự án:

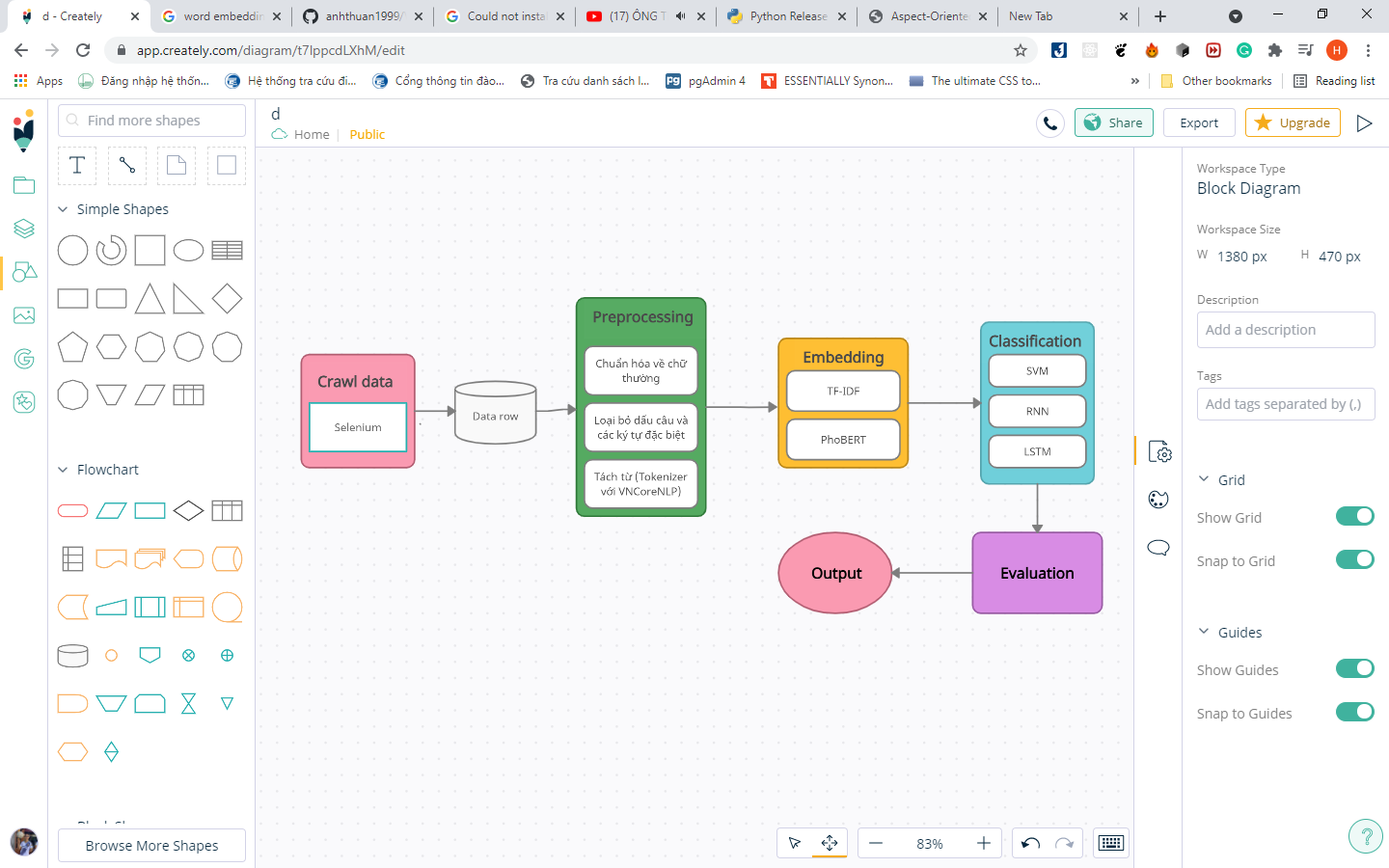
* numpy
* scipy
* pandas
* Keras
* Sklearn
* Selenium
* BeautifulSoup
* VnCoreNLP

### Môi trường huấn luyện và kiểm thử

Mô hình được huấn luyện và kiểm thử được thực hiện trên máy tính cá nhân Asus A566U. Máy tính chạy hệ điều hành Window 10 Pro. Cấu hình của máy tinh như sau:

* Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
* 8 GB Ram DDR4 @ 2400MHz

## Tổng quan về mô hình



Mô hình hóa quá trình thực hiện

Mô hình phân loại sắc thái bình luận bao gồm 3 phần chính là Preprocessing, Embedding, Classification được minh họa trong Hình 3.1. Trong phần Preprocessing, dữ liệu thô thu được từ các trang thương mại điện tử sẽ được làm sạch trước khi tiến tới các giai đoạn tiếp theo. Trong quá trình Embedding, dữ liệu sau khi làm sạch sẽ được chuyển từ dạng word sang vector bằng 2 phương pháp là TF-IDF và PhoBERT. Sau đó, dữ liệu ở dạng vector này sẽ được đưa vào các mô hình phân loại là SVM, RNN và LSTM. Sau cùng, chúng ta sẽ đánh giá mô hình cùng với các phương pháp được sử dụng.

## Chuẩn bị dữ liệu

Để giải quyết bài toán đặt ra, nhóm sử dụng thư viện Selenium để crawl dữ liệu các bình luận trên các trang web là: <https://shopee.vn/> và <https://tiki.vn/>.

Dữ liệu là các bình luận của các sản phẩm thuộc các danh mục sau: Điện thoại, Máy tính laptop, Thời trang nam, Thời trang nữ, Điện gia dụng, Bách hóa trước ngày 10/5/2021, bao gồm 15030 dữ liệu với 2 cột:

* Comment: Bình luận của khách hàng
* Rate: Đánh giá của khách hàng về sản phẩm (Từ 1 đến 5 sao)

Số lượng các nhãn trong bộ dữ liệu.

|  |  |
| --- | --- |
| Nhãn | Số lượng |
| 1 | 4114 |
| 2 | 3207 |
| 3 | 2935 |
| 4 | 2775 |
| 5 | 1999 |
| Tổng cộng | 15030 |

Dữ liệu sau khi thu thập được sẽ chuẩn hóa thông qua một số tiền xử lý cơ bản sau:

* Bước 1: Loại bỏ dấu câu và các ký tự đặc biệt.
* Bước 2: Loại bỏ những khoảng trắng, dấu cách không cần thiết.
* Bước 3: Chuẩn hóa về chữ thường.
* Bước 4: Tách từ (tokenizer sử dụng VNCoreNLP).

Ví dụ các bước trong phần Preprocessing.

|  |  |
| --- | --- |
| Input | Giao nhanh , đúng sản phẩm nhưng loa kém, KHÔNG NGHE THẤY GÌ!!! |
| Bước 1 | Giao nhanh đúng sản phẩm nhưng loa kém KHÔNG NGHE THẤY GÌ |
| Bước 2 | Giao nhanh đúng sản phẩm nhưng loa kém KHÔNG NGHE THẤY GÌ |
| Bước 3 | giao nhanh đúng sản phẩm nhưng loa kém không nghe thấy gì |
| Bước 4 | Giao nhanh đúng sản\_phẩm nhưng loa kém không nghe thấy gì |

## Word Embedding

### Term Frequency Inverse Document Frequency

Sau khi chuẩn hóa dữ liệu từ dữ liệu thô như bên trên, nhóm có sử dụng kỹ thuật TF-IDF để embedding từ dạng văn bản sang dạng vector để phụ vụ cho quá trình huấn luyện và kiểm thử phía sau.

**TF-IDF** (Term Frequency – Inverse Document Frequency) là 1 kĩ thuật sử dụng trong khai phá dữ liệu văn bản. TF-IDF có 2 trọng số là TF và IDF. Các trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Giá trị của các trọng số này phụ thuộc vào số lần xuất hiện của một từ trong văn bản và tần suất xuất hiện của từ đó trong tập dữ liệu. TF-IDF và một vài biến thể của nó thường được sử dụng trong các hệ thống tìm kiếm ví dụ như Elasticsearch - một search engine nổi tiếng. Tf-idf cũng được sử dụng để lọc những từ stopwords – những từ có giá trị thấp trong văn bản, giúp tăng hiệu quả của các mô hình trong xử lý ngôn ngữ tự nhiên.

### PhoBERT

BERT (Bidirectional Encoder Representations from Transformers) được hiểu là một mô hình học sẵn hay còn gọi là pre-train model, học các vector đại diện theo ngữ cảnh 2 chiều của từ, được sử dụng để transfer sang các bài toán khác trong lĩnh vực xử lý ngôn ngữ tự nhiên. BERT giúp cải thiện việc tìm ra đại diện của từ thông qua ngữ cảnh của nó.

Về mặt lý thuyết, các kỹ thuật khác như Word2vec, FastText hay Glove cũng tìm ra đại diện của từ thông qua ngữ cảnh chung của chúng. Tuy nhiên, những ngữ cảnh này là đa dạng trong dữ liệu tự nhiên. Ví dụ các từ như "con chuột" có ngữ nghĩa khác nhau ở các ngữ cảnh khác nhau như "Con chuột máy tính này thật đẹp!!" và "con chuột này to thật." Trong khi các mô hình như Word2vec, fastText tìm ra 1 vector đại diện cho mỗi từ dựa trên 1 tập ngữ liệu lớn nên không thể hiện được sự đa dạng của ngữ cảnh. Việc tạo ra một biểu diễn của mỗi từ dựa trên các từ khác trong câu sẽ mang lại kết quả ý nghĩa hơn nhiều. Như trong trường hợp trên ý nghĩa của từ con chuột sẽ được biểu diễn cụ thể dựa vào phần trước hoặc sau nó trong câu. Nếu đại diện của từ "con chuột" được xây dựng dựa trên những ngữ cảnh cụ thể này thì ta sẽ có được biểu diễn tốt hơn. BERT mở rộng khả năng của các phương pháp trước đây bằng cách tạo các biểu diễn theo ngữ cảnh dựa trên các từ trước và sau đó để dẫn đến một mô hình ngôn ngữ với ngữ nghĩa phong phú hơn.

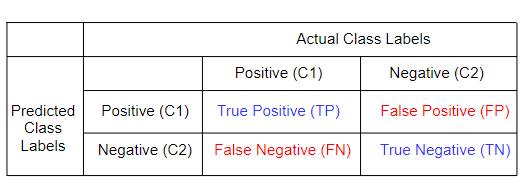
PhoBERT là là một pre-trained được huấn luyện monolingual language, tức là chỉ huấn luyện dành riêng cho tiếng Việt. Việc huấn luyện dựa trên kiến trúc và cách tiếp cận giống RoBERTa của Facebook được Facebook giới thiệu giữa năm 2019. Đây là một cái tiến so với BERT trước đây. Tương tự như BERT, PhoBERT cũng có 2 phiên bản là *PhoBERTbase*​ với 12 transformers block và *PhoBERTlarge*​ với 24 transformers block. PhoBERT được train trên khoảng 20GB dữ liệu bao gồm khoảng 1GB Vietnamese Wikipedia corpus và 19GB còn lại lấy từ Vietnamese news corpus. Đây là một lượng dữ liệu khả ổn để train một mô hình như BERT. PhoBERT sử dụng RDRSegmenter của [VnCoreNLP](https://github.com/vncorenlp/VnCoreNLP) để tách từ cho dữ liệu đầu vào trước khi qua BPE encoder.

## Kết quả thực nghiệm và đánh giá

Sau khi huấn luyện mô hình, làm sao để ta có thể biết được hiệu suất của mô hình đó như thế nào? Cần đánh giá để xem mô hình đó có thực sự hiệu quả không. Mức độ hiệu quả được đánh giá trên tập dữ liệu kiểm thử. Ta cần so sánh giữa kết quả mô hình dự đoán được và kết quả thực của nó. Có rất nhiều cách đánh giá mô hình phân lớp. Tùy vào những bài toán khác nhau mà ta sử dụng các phương pháp khác nhau.

### Đánh giá mô hình phân lớp

**Ma trận lỗi** là một ma trận tổng quát thể hiện kết quả phân lớp chính xác (gán nhãn đúng) và kết quả phân lớp sai (gán nhãn sai) được dự đoán bới mô hình phân lớp bằng cách so sánh với giá trị thực (nhãn thực) của nó. Ma trận cỡ NxN, trong đó N là số lượng dữ liệu cần gán nhãn. Đối với bài toán phân lớp nhị phân, ta có 2 lớp C1 – lớp dương (positive), C2 – lớp âm (negative), ma trận lỗi có dạng:



Ma trận lỗi.

* Nếu nhãn dự đoán của mô hình là “Positive” đúng với nhãn thực là “Positive” thì ta có True Positive (TP).
* Nếu nhãn dự đoán của mô hình là “Positive” sai với nhãn thực là “Negative” thì ta có False Positive (FP), được gọi là Cảnh báo nhầm.
* Nếu nhãn dự đoán của mô hình là “Negative” sai với nhãn thực là “Positive” thì ta có False Negative (FN), được gọi là Lỗi bỏ sót.
* Nếu nhãn dự đoán của mô hình là “Negative” đúng với nhãn thực là “Negative” thì ta có True Negative (TN).

**Độ chính xác** (Accuracy) là số phần trăm (%) đối tượng được đoán nhận đúng nhãn trên tổng số đối tượng cần gán nhãn.

**Tỷ lệ lỗi** (Error Rate) là số phần trăm đối tượng bị đoán nhận sai nhãn trên tổng số đối tượng cần gán nhãn.

**Độ đặc hiệu** (Precision) là tỷ lệ các trường hợp Positive được phân loại đúng.

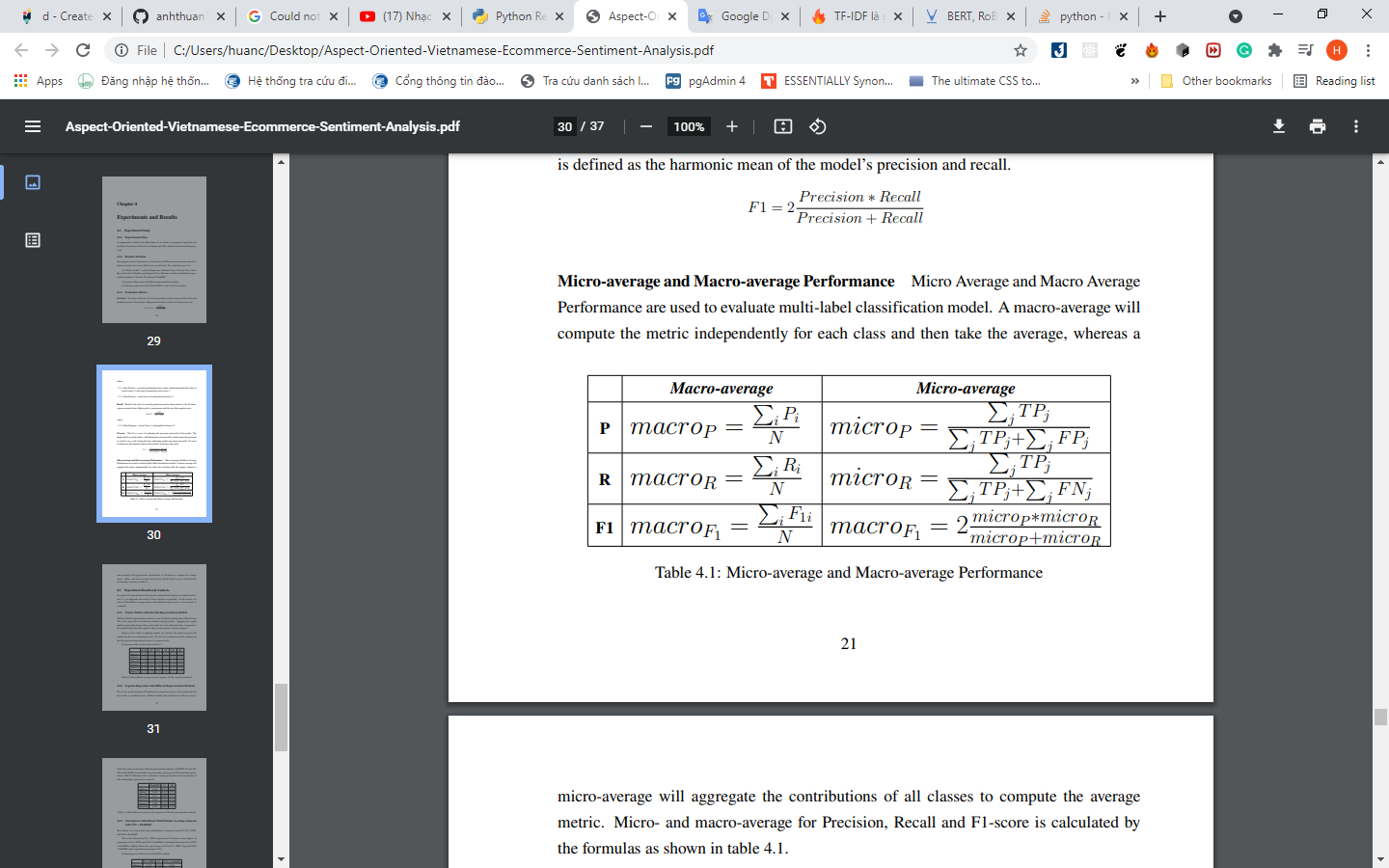
**Độ hồi tưởng** (Recall) là tỷ lệ các trường hợp Positive thực đã được phân loại đúng

**Chỉ số F1** (F1 Score) kết hợp 2 độ đo độ đặc hiệu và độ hồi tưởng để đánh giá hiệu quả của mô hình phân lớp. Chỉ số F1 cân bằng tầm quan trọng của cả độ đặc hiệu và độ hổi tưởng. Hay trong những trường hợp không phân biệt được độ đặc hiệu hay độ hồi tưởng tốt hơn thì nên sử dụng F1 score.

F1 score càng lớn thì hiệu suất mô hình phân lớp càng cao.

**Micro-average and Macro-average** được sử dụng để đánh giá mô hình phân lớp với nhiều lớp. Công thức tính micro và macro average cho Precision, Recall và F1-Score được biểu diễn như trong bảng 3.3.

Công thức tính micro-average và macro-average.



### Kết quả đánh giá mô hình

Với bộ dữ liệu như trên, nhóm triển khai và sử dụng vào quá trình huấn luyện và kiểm thử. Bộ dữ liệu được sẽ được dùng với các mô hình phân lớp với 3 lớp và 5 lớp.

Bảng bên dưới là kết quả thu được sau khi áp dụng TF-IDF ở bước Embedding.

Kết quả khi sử dụng TF-IDF với các mô hình.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | SVM  (3 lớp) | SVM (5 lớp) | RNN (3 lớp) | RNN (5 lớp) | LSTM(3 lớp) | LSTM(5 lớp) |
| micro-p | 0.80 | 0.61 | 0.77 | 0.56 | 0.83 | 0.59 |
| micro-r | 0.69 | 0.52 | 0.66 | 0.47 | 0.68 | 0.48 |
| micro-f1 | 0.73 | 0.55 | 0.70 | 0.49 | 0.71 | 0.51 |
| macro-p | 0.72 | 0.52 | 0.69 | 0.45 | 0.71 | 0.50 |
| macro-r | 0.61 | 0.39 | 0.55 | 0.40 | 0.63 | 0.39 |
| macro-f1 | 0.67 | 0.41 | 0.61 | 0.37 | 0.65 | 0.38 |

Bảng bên dưới là kết quả thu được sau khi áp dụng PhoBERT ở bước Embedding.

Kết quả khi sử dụng PhoBERT với các mô hình.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | SVM  (3 lớp) | SVM (5 lớp) | RNN (3 lớp) | RNN (5 lớp) | LSTM(3 lớp) | LSTM(5 lớp) |
| micro-p | 0.81 | 0.59 | 0.75 | 0.52 | 0.84 | 0.61 |
| micro-r | 0.66 | 0.53 | 0.61 | 0.42 | 0.69 | 0.50 |
| micro-f1 | 0.70 | 0.56 | 0.68 | 0.41 | 0.72 | 0.52 |
| macro-p | 0.72 | 0.51 | 0.69 | 0.47 | 0.74 | 0.53 |
| macro-r | 0.63 | 0.42 | 0.51 | 0.41 | 0.62 | 0.43 |
| macro-f1 | 0.68 | 0.39 | 0.59 | 0.39 | 0.68 | 0.44 |

Kết luận

Trong báo cáo này, nhóm có sử dụng 3 mô hình là SVM, RNN và LSTM để áp dụng vào bài toán phân loại sắc thái bình luận với dữ liệu được thu thập từ các trang thương mại điện tử lớn như shopee và tiki. Kết quả thu được khi phân loại với số lớp bằng 3 cho kết quả tương đối khả quan. So với 2 mô hình SVM và LSTM thì mô hình RNN tỏ ra kém hiệu quả hơn. Với phương pháp TF-IDF thì mô hình SVM cho kết quả tốt hơn là mô hình LSTM. Cụ thể là chỉ số mirco-f1 thu được ở mô hình SVM là 0.73 trong khi mô hình LSTM là 0.71, chỉ số macro-f1 ở mô hình SVM là 0.67 trong khi mô hình LSTM là 0.65. Ngược lại với TF-IDF thì PhoBERT cho kết quả với LSTM tốt hơn SVM. Trong khi chi số macro-f1 của 2 mô hình này đề bằng nhau là 0.68 thì chỉ số micro-f1 của LSTM là 0.72 cao hơn micro-f1 của SVM 2%.

Với kết quả thu được, ta thấy những cải tiến của LSTM so với RNN là tương đối hiệu quả khi đều cho kết quả tốt với cả TF-IDF và PhoBERT. Với bộ dữ liệu nhỏ khoảng hơn 10000 bình luận thì kết quả phân loại của SVM + TF-IDF là tương đối tốt khi so sánh với LSTM + PhoBERT. Khi có một bộ dữ liệu lớn hơn, mô hình LSTM với PhoBERT có thể sẽ cho kết quả chính xác hơn.

Trong tương lai, nhóm sẽ cố gắng cải thiện phần tiền xử lý dữ liệu để giúp quá trình huấn luyện có thể đạt kết quả cao hơn.

Tài liệu tham khảo

[1] Pham Huu Quang, “Vietnamese Sentiment Analysis” on viblo.asia website. October 14, 2018.

[2] Tran Anh Thuan, Nguyen Anh Nhat, Bui Xuan Thanh, Vo Nguyen Tam An and Le Hoanh Su, “A proposed method for opinion mining online media of smartphone with Vietnamese text” in The 6th international Conference for Small and Medium Business 2020(ICSMB 2020).

[3] Pham Huu Quang, “BERT- bước đột phá mới trong công nghệ xử lý ngôn ngữ tự nhiên của Google”, “BERT, RoBERTa, PhoBERT, BERTweet: Ứng dụng state-of-the-art pre-trained model cho bài toán phân loại văn bản” on viblo.asia website. November 27, 2018

[4] “Khai phá dữ liệu”, “Máy vector hỗ trợ” on vi.wikipedia.org website. June 09, 2021.

[5] Christopher Olah, “Understanding LSTM Networks” on colah.github.io website. August 27, 2015.

Phân công công việc

[1] Phan Lương Huân: Thu thập xử lý dữ liệu, tiến hành cài đặt train và test, làm báo cáo.

[2] Nguyễn Việt Hà: Làm slide.

[3] Lê Tuấn Minh: train và test, làm báo cáo phần RNN.

Mã nguồn của dự án

[1] Github: <https://github.com/huancnttuet/CommentClassification>

[2] Azure Devops: <https://dev.azure.com/bambookite/Comment%20Rate>