TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

**KHOA ĐIỆN TỬ**

**Bộ Môn: Công Nghệ Thông Tin**



**BÀI TẬP LỚN**

MÔN HỌC

**KHOA HỌC DỮ LIỆU**

*Đề tài: Dự đoán bình luận trên Facebook*

|  |  |
| --- | --- |
| Sinh viên: | NGUYỄN THỌ DUY |
| Lớp: | 55KMT |
| GVHD: | TS.NGUYỄN VĂN HUY |

**Thái Nguyên – 2023**

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP LỚN**

**MÔN HỌC**: **KHOA HỌC DỮ LIỆU**

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

*Sinh viên:* ***Nguyễn Thọ Duy*** *Lớp: K55KMT*

*Ngành: Kỹ thuật máy tính*

*Ngày giao đề: 15/05/2023 Ngày hoàn thành:*

1.Tên đề tài : Dự đoán bình luận trên Facebook

2. Nội dung :

3. Các bản vẽ, chương trình và đồ thị: Kiểm thử chương trình

|  |  |
| --- | --- |
| TRƯỞNG BỘ MÔN | GIÁO VIÊN HƯỚNG DẪN |
| *(Ký và ghi rõ họ tên)* | *(Ký và ghi rõ họ tên)* |

**TS.Nguyễn Văn Huy**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Thái Nguyên, ngày….tháng…..năm 2023*

**GIÁO VIÊN HƯỚNG DẪN**

(Ký ghi rõ họ tên)

**NHẬN XÉT CỦA GIÁO VIÊN CHẤM**

*Thái Nguyên, ngày… tháng…..năm 2023*

**GIÁO VIÊN CHẤM**

*(Ký ghi rõ họ tên)*

**MỤC LỤC**

[LỜI NÓI ĐẦU 6](#_Toc136389152)

[CHƯƠNG I : GIỚI THIỆU 7](#_Toc136389153)

[1.1 Dự đoán bình luận spam là gì ? 7](#_Toc136389154)

[1.2 Ứng dụng của việc dự đoán bình luận spam 8](#_Toc136389155)

[1.3 Phân biệt bình luận spam ? 9](#_Toc136389156)

[1.4 Phương pháp để dự đoán bình luận spam 10](#_Toc136389157)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 11](#_Toc136389158)

[2.1 Tokenizer là gì ? 11](#_Toc136389159)

[2.2 Mô hình Phobert 12](#_Toc136389160)

[2.3 Phobert Embedding 13](#_Toc136389161)

[2.4 PhoBERT classification 14](#_Toc136389162)

[CHƯƠNG III: XÂY DỰNG CHƯƠNG TRÌNH 15](#_Toc136389163)

[3.1 Xử lý dữ liệu 15](#_Toc136389164)

[3.2 Huấn luận mô hình 17](#_Toc136389165)

[3.3 Demo 21](#_Toc136389166)

[CHƯƠNG IV: KẾT LUẬN 22](#_Toc136389167)

[TÀI LIỆU THAM KHẢO 24](#_Toc136389168)

# LỜI NÓI ĐẦU

Hiện nay, trong thời đại mạng xã hội phát triển những bình luận spam trở thành một vấn đề nghiêm trọng và đáng lo ngại. Người dùng đang gặp phải rất nhiều bình luận spam trên các nền tảng truyền thông xã hội, diễn đàn và các hệ thống gửi tin nhắn. Những bình luận này không chỉ gây phiền toái, mất thời gian và làm giảm trải nghiệm sử dụng, mà còn có thể chứa các liên kết độc hại, lừa đảo hoặc vi rút.

Vì vậy, việc xây dựng một hệ thống dự đoán bình luận spam là cực kỳ quan trọng và cần thiết. Mục tiêu của chúng ta là tạo ra một giải pháp tự động, sử dụng mô hình PhoBERT, để phát hiện và loại bỏ các bình luận spam khỏi nền tảng truyền thông xã hội và các hệ thống gửi tin nhắn.

Trong báo cáo này, chúng ta sẽ tìm hiểu về việc sử dụng mô hình PhoBERT để dự đoán bình luận spam. Mô hình đã được huấn luyện trên dữ liệu tiếng Việt và có khả năng hiểu và rút trích đặc trưng từ các văn bản tiếng Việt. Chúng ta sẽ thực hiện các bước tiền xử lý dữ liệu, xây dựng mô hình và đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.

Thông qua việc áp dụng mô hình PhoBERT, chúng ta có thể giúp tạo ra một môi trường trực tuyến an toàn và sạch sẽ hơn, nơi mà người dùng có thể trải nghiệm và tương tác một cách tích cực và bình yên hơn.

# CHƯƠNG I : GIỚI THIỆU

## 1.1 Dự đoán bình luận spam là gì ?



Hình 1: Spam bình luận trên Facebook

Spam là từ viết tắt của cụm từ tiếng Anh : Stupid – Pointless -Annoying - Messages nhằm để chỉ những thư có nội dung phiền toái, vô nghĩa đối với người nhận. Đó là những tin được gửi đồng loạt cho nhiều người với mục đích quảng cáo là chính.

Spam bắt đầu được sử dụng vào năm 1978, khi mà một người đàn ông thay vì gửi tin nhắn cho từng người đã sử dụng Spam để gửi tin nhắn cùng một lúc cho hơn 300 người. Càng về sau, những tin nhắn Spam càng phổ biến hơn. Tuy nhiên, vấn đề nằm ở chỗ những tin nhắn Spam đều vô nghĩa đối với người nhận và khiến họ cảm thấy khó chịu. Vậy nên, trên Facebook hay Email đều có mục Spam để chứa những tin nhắn không cần thiết đối với người dùng.

Bình luận spam là các bình luận không mong muốn hoặc không liên quan, thường chứa thông tin quảng cáo, lừa đảo, hoặc các liên kết độc hại. Những bình luận spam thường được tạo ra tự động hoặc gửi từ các tài khoản giả mạo, và có thể gây phiền toái và rủi ro cho người dùng.

Các loại spam có thể bao gồm:

* Email spam: Đây là hình thức phổ biến nhất của spam, trong đó các tin nhắn không mong muốn được gửi đến hộp thư điện tử của người dùng. Thông điệp trong email spam thường chứa quảng cáo sản phẩm, thông tin lừa đảo, hoặc các liên kết độc hại.
* Spam trên mạng xã hội: Đây là hình thức spam mà người dùng gặp phải trên các nền tảng truyền thông xã hội. Các bình luận spam, tin nhắn riêng tư hoặc các thông báo không mong muốn có thể chứa các liên kết độc hại hoặc nội dung quảng cáo không liên quan.
* Spam trên diễn đàn: Đây là hình thức spam mà người dùng gặp phải trên các diễn đàn trực tuyến. Những bài đăng spam thường chứa các liên kết quảng cáo, nội dung không liên quan hoặc thông tin lừa đảo.
* Spam trên tin nhắn điện thoại di động: Đây là hình thức spam gửi qua tin nhắn SMS hoặc ứng dụng nhắn tin. Những tin nhắn spam này có thể chứa quảng cáo, thông tin lừa đảo hoặc các dịch vụ không mong muốn.

Dự đoán bình luận spam là quá trình sử dụng một mô hình hoặc một hệ thống tự động để phân loại các bình luận vào hai nhóm: spam và không spam. Mục tiêu là xác định xem một bình luận cụ thể có phải là bình luận spam hay không.

Quá trình dự đoán bình luận spam thường dựa trên việc xây dựng một mô hình máy học hoặc một hệ thống dựa trên quy tắc. Mô hình này được huấn luyện trên dữ liệu đã được gán nhãn là spam hoặc không spam, và từ đó có thể học được các đặc trưng và mẫu của các bình luận spam. Khi một bình luận mới được đưa vào, mô hình sẽ đánh giá bình luận và dự đoán xem liệu có phải là spam hay không. Tuỳ vào mục đích spam còn chia thành các mức độ khác nhau.

## 1.2 Ứng dụng của việc dự đoán bình luận spam

Dự đoán bình luận spam có nhiều ứng dụng trong các lĩnh vực liên quan đến truyền thông trực tuyến và quản lý nội dung. Dưới đây là một số ứng dụng quan trọng của việc dự đoán bình luận spam:

* **Bảo vệ người dùng:** Dự đoán bình luận spam giúp bảo vệ người dùng khỏi sự quấy rối, lừa đảo và các liên kết độc hại. Bằng cách loại bỏ các bình luận spam, người dùng có thể trải nghiệm một môi trường trực tuyến an toàn hơn và không bị phiền hà.
* **Cải thiện trải nghiệm người dùng:** Spam có thể làm giảm chất lượng và giá trị của các nền tảng truyền thông xã hội, diễn đàn và hệ thống gửi tin nhắn. Bằng cách dự đoán và loại bỏ bình luận spam, ta tăng cường trải nghiệm người dùng bằng cách giảm thiểu nhiễu và nội dung không liên quan.
* **Tăng cường an ninh mạng:** Spam có thể chứa các liên kết độc hại hoặc nội dung lừa đảo có thể gây nguy hiểm cho người dùng. Bằng cách phát hiện và ngăn chặn bình luận spam, ta giúp bảo vệ người dùng khỏi các mối đe dọa an ninh mạng.
* **Quản lý nội dung:** Việc dự đoán bình luận spam có thể hỗ trợ quản lý nội dung trên các nền tảng truyền thông xã hội, diễn đàn và trang web. Các hệ thống dự đoán spam có thể tự động phân loại và loại bỏ bình luận spam, giúp duy trì chất lượng và tính toàn vẹn của nội dung trên mạng.
* **Nghiên cứu và phân tích dữ liệu:** Dữ liệu từ các bình luận spam có thể được sử dụng để nghiên cứu và phân tích xu hướng, mô hình hành vi spam và các chiến lược spam. Những thông tin này có thể giúp định hình các chiến lược phòng ngừa và đối phó với spam trong tương lai.

Như vậy, việc dự đoán bình luận spam đóng vai trò quan trọng trong việc bảo vệ người dùng, cải thiện trải nghiệm và an toàn trực tuyến.

## 1.3 Phân biệt bình luận spam ?

Để phân biệt bình luận spam, có thể sử dụng các phương pháp và công cụ:

Phân tích từ khóa: Kiểm tra các từ khóa phổ biến trong các bình luận spam, như "quảng cáo," "giảm giá," "lừa đảo," "liên kết độc hại," và nhiều hơn nữa. Nếu một bình luận chứa nhiều từ khóa spam, có khả năng cao là một bình luận spam.

Phân tích cú pháp: Phân tích cú pháp của bình luận để xác định các mẫu ngữ pháp hoặc câu trùng lặp, có thể cho thấy bình luận là spam. Các bình luận spam thường có cấu trúc lạ và không tự nhiên.

Kiểm tra độ dài: Xem xét độ dài của bình luận. Các bình luận spam thường ngắn và không có nội dung cụ thể, trong khi các bình luận hợp lệ thường có nội dung dài hơn và chứa thông tin cụ thể.

Phân tích URL và liên kết: Kiểm tra các liên kết có trong bình luận để xem liệu chúng có phải là liên kết độc hại, không liên quan hoặc liên kết đến trang web spam hay không.

Đánh giá đáng tin cậy của tài khoản: Xem xét độ tin cậy của tài khoản gửi bình luận. Các tài khoản mới được tạo gần đây hoặc không có lịch sử hoạt động có thể là dấu hiệu của một tài khoản giả mạo gửi bình luận spam.

Sử dụng mô hình học máy: Xây dựng một mô hình học máy dựa trên dữ liệu đã được gán nhãn là spam hoặc không spam để tự động phân loại các bình luận. Mô hình này có thể học các đặc trưng và mẫu của các bình luận spam và không spam, từ đó dự đoán xem một bình luận mới có phải là spam hay không.

Sử dụng công cụ phân loại spam: Có sẵn các công cụ và thư viện phân loại spam như SpamAssassin, Akismet, hoặc Google CAPTCHA, có thể được tích hợp vào hệ thống để tự động phát hiện và loại bỏ bình luận spam.

## 1.4 Phương pháp để dự đoán bình luận spam

Để dự đoán một bình luận spam có thể sử dụng các đặc điểm:

Độ dài: Bình luận spam thường có tend to dài hơn và chứa nhiều từ ngữ không liên quan, lặp đi lặp lại hoặc vô nghĩa.

Từ khóa: Có thể xây dựng một danh sách từ khóa liên quan đến spam và kiểm tra xem bình luận có chứa các từ khóa này hay không.

Liên kết (URL): Bình luận spam thường chứa các liên kết không mong muốn hoặc đường dẫn đến trang web đáng ngờ.

Ngôn ngữ: Các bình luận spam có thể chứa ngôn ngữ thiếu văn hóa, lời lẽ không phù hợp hoặc nội dung không phù hợp.

Ký tự đặc biệt: Các bình luận spam có thể chứa nhiều ký tự đặc biệt, biểu tượng hoặc dấu câu không cần thiết.

Đánh giá: Xem xét đánh giá hoặc đánh giá của người dùng khác về bình luận. Bình luận spam thường có đánh giá thấp hoặc phản hồi tiêu cực.

Từ các đặc điểm trên có thể xây dựng được bộ dữ liệu để phân biệt ra làm hai loại bình luận riêng biệt với các nhãn spam (Label = 1) và không phải spam (Label = 0)

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT

## 2.1 Tokenizer là gì ?

Tokenizer là một thuật toán có nhiệm vụ tách từ, cụm từ trong văn bản và là 1 nhánh con trong xử lý ngôn ngữ tự nhiên (NLP). Hay hiểu đơn giản Tokenize là quá trình chia đoạn (segmentation) một văn bản thành các đơn vị nhỏ hơn gọi là "token". Trong xử lý ngôn ngữ tự nhiên, các token thường tương ứng với các từ, subword units hoặc ký tự.

Trong ngôn ngữ từ có thể được coi là các yếu tố nhỏ nhất trong ngôn ngữ tự nhiên. Tokenizer là một thuật toán có nhiệm vụ tách từ, cụm từ trong văn bản. Quá trình tokenize nhằm tách một chuỗi văn bản thành các đơn vị có ý nghĩa nhỏ hơn để tiện cho việc xử lý và phân tích ngôn ngữ. Các token có thể được sử dụng để xây dựng từ điển, đếm tần suất từ và n-gram, rút trích đặc trưng và thực hiện các tác vụ xử lý ngôn ngữ tự nhiên khác.

Ví dụ: Phiên điều trần luận tội → Phiên/điều trần/luận tội. Có 2 loại tokenizer thường dùng là POS và NER. Có thể nói tokenizer chính là vấn đề rất quan trọng trong xử lý ngôn ngữ tự nhiên, giải quyết được bài toán này rất quan trọng và ứng dụng rất nhiều trong các bài toán phức tạp hơn như phân loại (classification), hồi quy (regression).

Trong xử lý ngôn ngữ tự nhiên sẽ có một số kiểu tokenize như sau:

* **Tokenize theo word level:** Phân tách câu thành các token được ngăn cách bởi khoảng trắng hoặc dấu câu. Khi đó mỗi token là một từ đơn âm tiết. Đây là phương pháp token được sử dụng trong các thuật toán nhúng từ truyền thống như GloVe, word2vec.
* **Tokenize theo multi-word level:** Tiếng Việt và một số ngôn ngữ khác tồn tại từ đơn âm tiết (từ đơn) và từ đa âm tiết (từ ghép). Do đó nếu token theo từ đơn âm tiết sẽ làm nghĩa của từ bị sai khác. Ví dụ cụm từ vô xác định nếu được chia thành vô, xác và định sẽ làm cho từ bị mất đi nghĩa phủ định của câu. Do đó để tạo ra được các từ với nghĩa chính xác thì chúng ta sẽ sử dụng thêm từ điển bao gồm cả từ đa âm tiết và đơn âm để tokenize câu. Trong Tiếng Việt có khá nhiều các module hỗ trợ tokenize dựa trên từ điển như VnCoreNLP, pyvivn, underthesea.
* **Tokenize theo character level:** Tokenize theo word level thường sinh ra một từ điển với kích thước rất lớn, điều này làm gia chi phí tính toán. Hơn nữa nếu tokenize theo word level thì đòi hỏi từ điển phải rất lớn thì mới hạn chế được những trường hợp từ nằm ngoài từ điển. Tuy nhiên nếu phân tích ta sẽ thấy hầu hết các từ đều có thể biểu thị dưới một nhóm các ký tự là chữ cái, con số, dấu xác định. Như vậy chỉ cần sử dụng một lượng các ký tự rất nhỏ có thể biểu diễn được mọi từ. Từ được token dựa trên level ký tự sẽ có tác dụng giảm kích thước từ điển mà vẫn biểu diễn được các trường hợp từ nằm ngoài từ điển. Đây là phương pháp được áp dụng trong mô hình fasttext.

## 2.2 Mô hình Phobert

PhoBERT (Pre-trained Language Model for Vietnamese) là một mô hình ngôn ngữ tiên tiến được huấn luyện trên dữ liệu tiếng Việt rất lớn sử dụng kiến trúc của BERT (Bidirectional Encoder Representations from Transformers). BERT là một trong những mô hình ngôn ngữ tiền huấn luyện nổi tiếng được phát triển bởi Google, nó đã mang lại tiến bộ đáng kể trong nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên (NLP).

PhoBERT được huấn luyện trên một tập dữ liệu tiếng Việt rộng lớn và sử dụng kiến trúc Transformer, cho phép nắm bắt được sự phụ thuộc ngữ nghĩa và ngữ cảnh trong ngôn ngữ tiếng Việt. Mô hình này đã học cách biểu diễn các từ, câu và văn bản trong không gian vectơ nhiều chiều.

PhoBERT mang lại nhiều lợi ích cho việc xử lý ngôn ngữ tiếng Việt trong các tác vụ NLP. Điều này bao gồm:

Biểu diễn ngữ nghĩa chính xác: PhoBERT hiểu được ngữ nghĩa của từ và câu trong tiếng Việt, giúp cải thiện hiệu suất cho các tác vụ như phân loại văn bản, trích xuất thông tin, và dịch máy.

Xử lý ngôn ngữ tự nhiên tiếng Việt phức tạp: Với việc huấn luyện trên dữ liệu tiếng Việt, PhoBERT có khả năng xử lý ngôn ngữ tự nhiên phức tạp như từ ghép, dấu thanh, và cấu trúc ngữ pháp của tiếng Việt.

Các phiên bản của Phobert bao gồm:

* **PhoBERT-base:** Đây là phiên bản cơ bản của PhoBERT, được huấn luyện trên dữ liệu lớn tiếng Việt. Nó có kích thước lớp nhúng (embedding layer) là 768, có tổng cộng 12 layer transformer và 110 triệu tham số.
* **PhoBERT-large:** Đây là phiên bản mở rộng của PhoBERT, có kích thước lớp nhúng là 1,024, có tổng cộng 24 layer transformer và khoảng 210 triệu tham số. Phiên bản này được huấn luyện với một lượng dữ liệu lớn hơn so với PhoBERT-base, mang lại khả năng biểu diễn ngôn ngữ tốt hơn.
* **PhoBERT-base v3:** Phiên bản này được cải tiến từ PhoBERT-base ban đầu, được huấn luyện trên dữ liệu lớn hơn và mang lại hiệu suất tốt hơn trong các tác vụ như phân loại văn bản, trích xuất thông tin, dịch máy, và tương tự.
* **PhoBERT-softmax:** Phiên bản này được tối ưu hóa cho tác vụ phân loại văn bản. Thay vì sử dụng lớp nhúng cuối cùng để lấy vectơ biểu diễn của câu, nó sử dụng lớp softmax để tính toán xác suất của từng lớp phân loại.
* **PhoBERT-tokenizer:** Đây là phiên bản chỉ chứa tokenizer của PhoBERT, được sử dụng để tách từ và chuẩn hóa văn bản thành các thành phần ngôn ngữ nhỏ hơn (ví dụ: từ, câu).
* **Phobert-for-G2T:** Đây là mô hình được tinh chỉnh đặc biệt cho tác vụ dịch máy từ tiếng Việt sang tiếng Anh (G2T - General-to-Text). Mô hình này được huấn luyện trên dữ liệu tiếng Việt và tiếng Anh song ngữ, giúp cải thiện hiệu suất dịch máy tiếng Việt sang tiếng Anh.
* **Phobert-for-Sequence-Classification:** Đây là mô hình được tinh chỉnh cho tác vụ phân loại chuỗi (sequence classification) trong đó mục tiêu là dự đoán nhãn cho các câu hoặc văn bản. Mô hình này có thể được sử dụng cho các tác vụ như phân loại cảm xúc, phân loại tin tức, phân loại spam và nhiều tác vụ khác.

## 2.3 Phobert Embedding

PhoBERT embedding là biểu diễn vector của một văn bản sử dụng PhoBERT. Khi sử dụng PhoBERT, mỗi từ trong văn bản sẽ được mã hóa thành một vector số thực đa chiều. Tổng hợp các vector từ các từ thành biểu diễn vector của cả văn bản đó.

PhoBERT sử dụng kiến trúc mô hình BERT (Bidirectional Encoder Representations from Transformers) để tạo ra các embedding. Mô hình BERT sử dụng một lớp nhúng (embedding layer) đầu tiên để biểu diễn các từ thành các vectơ số thực. Sau đó, các đoạn (sentences) hoặc câu (sentences) được mã hóa thông qua một loạt các lớp transformer, đại diện cho ngữ cảnh và ý nghĩa của từng từ trong văn bản.

PhoBERT embedding là kết quả cuối cùng của quá trình mã hóa văn bản bằng mô hình PhoBERT. Nó cung cấp một biểu diễn số học của văn bản, cho phép các thuật toán máy học và các mô hình phân loại khác nhau sử dụng thông tin ngôn ngữ của văn bản đó để thực hiện các tác vụ như phân loại, dự đoán, và xử lý ngôn ngữ tự nhiên.

## 2.4 PhoBERT classification

PhoBERT classification là quá trình sử dụng mô hình PhoBERT để phân loại các văn bản vào các nhãn (labels) tương ứng. Mô hình PhoBERT được huấn luyện trên một tập dữ liệu lớn để hiểu và biểu diễn thông tin ngôn ngữ, sau đó có thể được sử dụng để thực hiện các tác vụ phân loại văn bản.

Quá trình PhoBERT classification thường được thực hiện bằng cách đưa mỗi văn bản vào mô hình PhoBERT để lấy embedding của văn bản đó. Embedding là biểu diễn vector của văn bản trong không gian số học. Sau khi có được embedding, ta có thể đưa nó qua một hoặc nhiều lớp mạng nơ-ron để dự đoán nhãn của văn bản.

Các bước thực hiện PhoBERT classification:

* **Tiền xử lý dữ liệu:** Chuẩn hóa và làm sạch dữ liệu văn bản, bao gồm việc loại bỏ các ký tự đặc biệt, chuyển đổi văn bản về chữ thường, loại bỏ stop words, và thực hiện các phép xử lý khác cần thiết.
* **Trích xuất embedding:** Sử dụng mô hình PhoBERT để trích xuất embedding cho mỗi văn bản. Đầu vào là các câu hoặc đoạn văn bản, và đầu ra là một biểu diễn vector đa chiều của văn bản.
* **Huấn luyện mô hình phân loại**: Sử dụng các phương pháp huấn luyện máy học như học có giám sát (supervised learning) để huấn luyện một mô hình phân loại sử dụng embedding của văn bản làm đầu vào và nhãn làm đầu ra. Các lớp mạng nơ-ron thông thường như Multilayer Perceptron (MLP), Support Vector Machines (SVM), hoặc các mô hình deep learning như Convolutional Neural Networks (CNN) hoặc Recurrent Neural Networks (RNN) có thể được sử dụng.
* **Đánh giá mô hình:** Đánh giá hiệu suất của mô hình phân loại trên tập dữ liệu kiểm tra hoặc sử dụng các phương pháp đánh giá như độ chính xác (accuracy), độ phủ (recall), độ chính xác gia tăng (precision), và F1-score.

# CHƯƠNG III: XÂY DỰNG CHƯƠNG TRÌNH

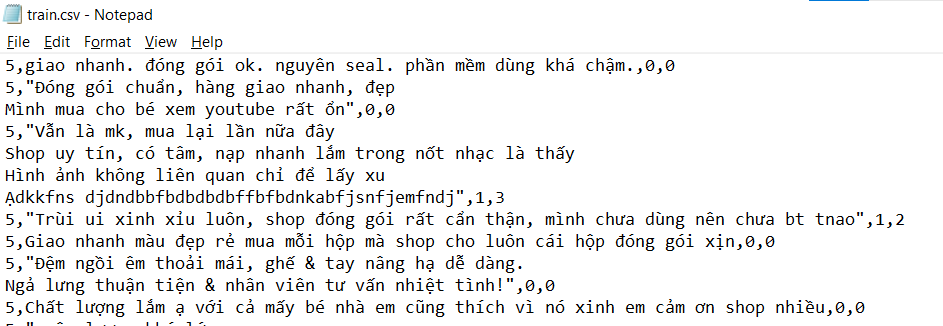
## 3.1 Xử lý dữ liệu

Dữ liệu được chuẩn bị từ rất nhiều nguồn khác nhau và được chia thành 3 file lớn là:

Tập huấn luyện (train.csv): Được sử dụng để huấn luyện mô hình, là tập dữ liệu mà mô hình sẽ học từ đó.

Tập kiểm tra (test.csv): Được sử dụng để kiểm tra hiệu suất của mô hình đã được huấn luyện. Dữ liệu trong tập kiểm tra không được sử dụng trong quá trình huấn luyện mô hình.

Tập phát triển (dev.csv): Được sử dụng để đánh giá hiệu suất của mô hình trong quá trình điều chỉnh siêu tham số (hyperparameters) và lựa chọn mô hình tốt nhất. Dữ liệu trong tập phát triển cũng không được sử dụng trong quá trình huấn luyện mô hình.



*Hình 2: Mô tả dữ liệu file train*

Các bước tiền xử lý dữ liệu:

* Sử dụng thư viện VnCoreNLP để tách từ và phân tích từ tiếng Việt.
* Loại bỏ stop words.
* Xử lý các biểu tượng cảm xúc trong dữ liệu.
* Chuyển đổi văn bản thành dạng chữ thường (lowercase).

Sử dụng thư viện VnCoreNLP để tách từ và phân tích từ tiếng Việt: Thư viện VnCoreNLP được sử dụng để phân tích và tách từ tiếng Việt. Quá trình này giúp tách các từ trong văn bản thành các từ riêng lẻ, tạo cơ sở cho các bước xử lý và phân loại sau này.

Loại bỏ stop words từ file vietnamese-stopwords-dash.txt: Stop words là những từ phổ biến và không mang ý nghĩa đặc biệt trong quá trình xử lý ngôn ngữ tự nhiên. Bước loại bỏ stop words giúp loại bỏ những từ này khỏi văn bản để giảm nhiễu và tăng độ chính xác của quá trình phân loại.

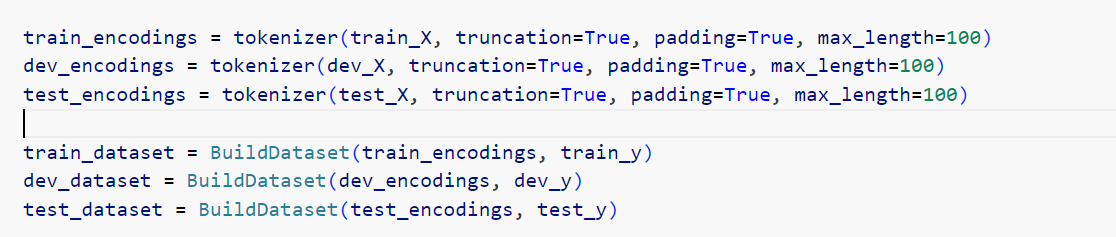
Xử lý biểu tượng cảm xúc trong dữ liệu bằng hàm deEmojify(): Emoji là biểu tượng cảm xúc được sử dụng trong văn bản. Việc xử lý biểu tượng cảm xúc bằng hàm deEmojify() giúp loại bỏ hoặc thay thế các emoji trong văn bản, đồng thời giúp giảm nhiễu và tăng tính chính xác của quá trình phân loại.

Chuyển đổi văn bản thành dạng lowercase: Chuyển đổi văn bản thành lowercase có thể giúp đồng nhất dữ liệu và giảm độ phức tạp của từ vựng. Điều này đảm bảo rằng các từ viết hoa và viết thường được coi là giống nhau, giúp mô hình hiểu và xử lý dữ liệu tốt hơn.



*Hình 3: Tiền xử lý dữ liệu*

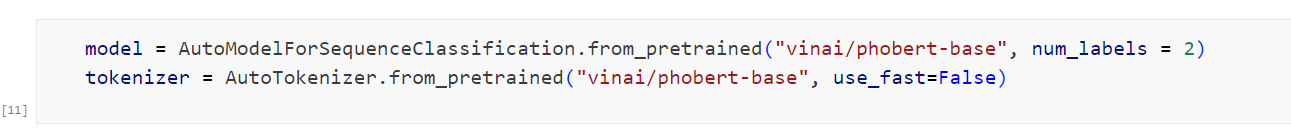
Tiếp theo là sử dụng tokenizer để mã hóa các câu trong tập dữ liệu train, dev và test thành dạng vector.



*Hình 4: Xây dựng dữ liệu đầu vào cho model*

## 3.2 Huấn luận mô hình

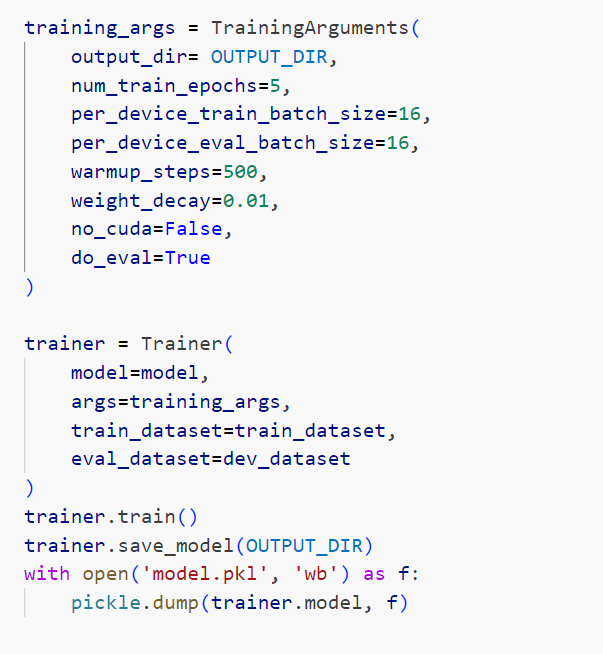
Model sử dụng của Phobert với số nhãn đâu vào là 2.



*Hình 5: Model xử dụng*

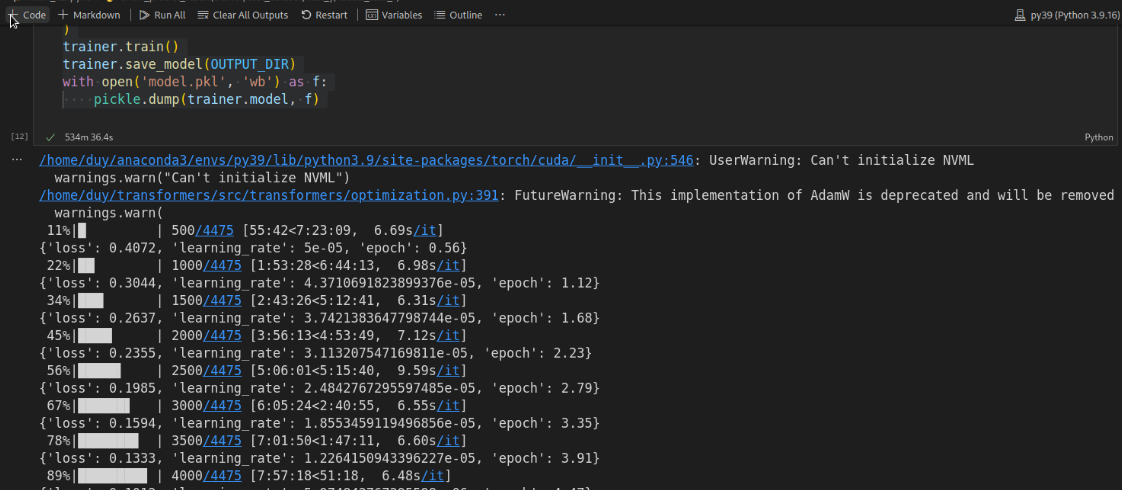
Một số các thông số:

* output\_dir: Đường dẫn đến thư mục lưu trữ kết quả huấn luyện.
* num\_train\_epochs: Số lượng epoch (vòng lặp) huấn luyện.
* per\_device\_train\_batch\_size: Số lượng mẫu huấn luyện được đưa vào mô hình trên mỗi thiết bị (GPU) trong mỗi batch.
* per\_device\_eval\_batch\_size: Số lượng mẫu đánh giá được đưa vào mô hình trên mỗi thiết bị (GPU) trong mỗi batch.
* warmup\_steps: Số lượng bước huấn luyện ban đầu dùng để làm nóng mô hình trước khi tăng tỉ lệ học (learning rate).
* weight\_decay: Hệ số giảm trọng lượng (weight decay) để tránh quá khớp (overfitting).
* no\_cuda: Có sử dụng GPU hay không. Đặt giá trị False để sử dụng GPU.
* do\_eval: Có đánh giá kết quả trên tập validation (dev) hay không.
* model: Mô hình cần huấn luyện.
* args: Đối tượng TrainingArguments chứa các thông số cấu hình huấn luyện.
* train\_dataset: Bộ dữ liệu huấn luyện.
* eval\_dataset: Bộ dữ liệu đánh giá (validation).

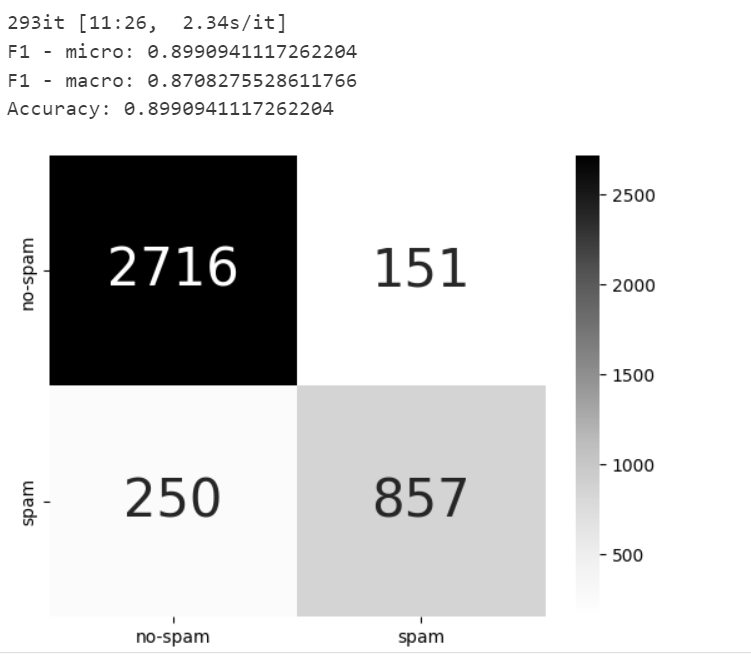


*Hình 6: Train model*

Quá trình huấn luyện mô hình mất khá nhiều thời gian hơn 500 phút với số lần train là 5.

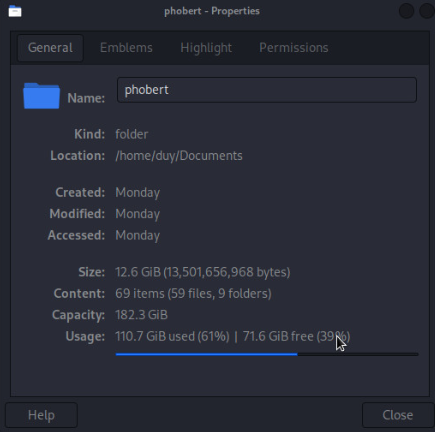


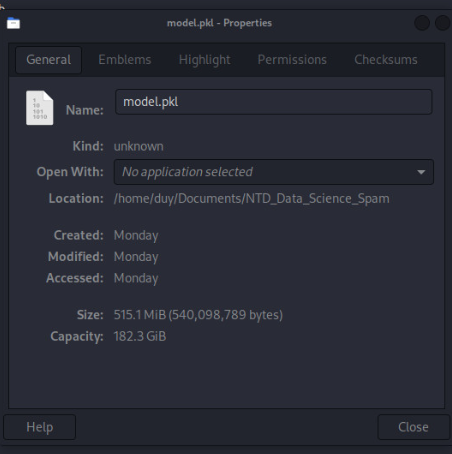
*Hình 7: Quá trình huấn luyện*



*Hình 8: Kết quả sau khi huấn luyện.*

Kết quả thu được sau khi huấn luyện với Accuracy gần 90% và file huấn luyện khá là nặng khoảng 13 GB.

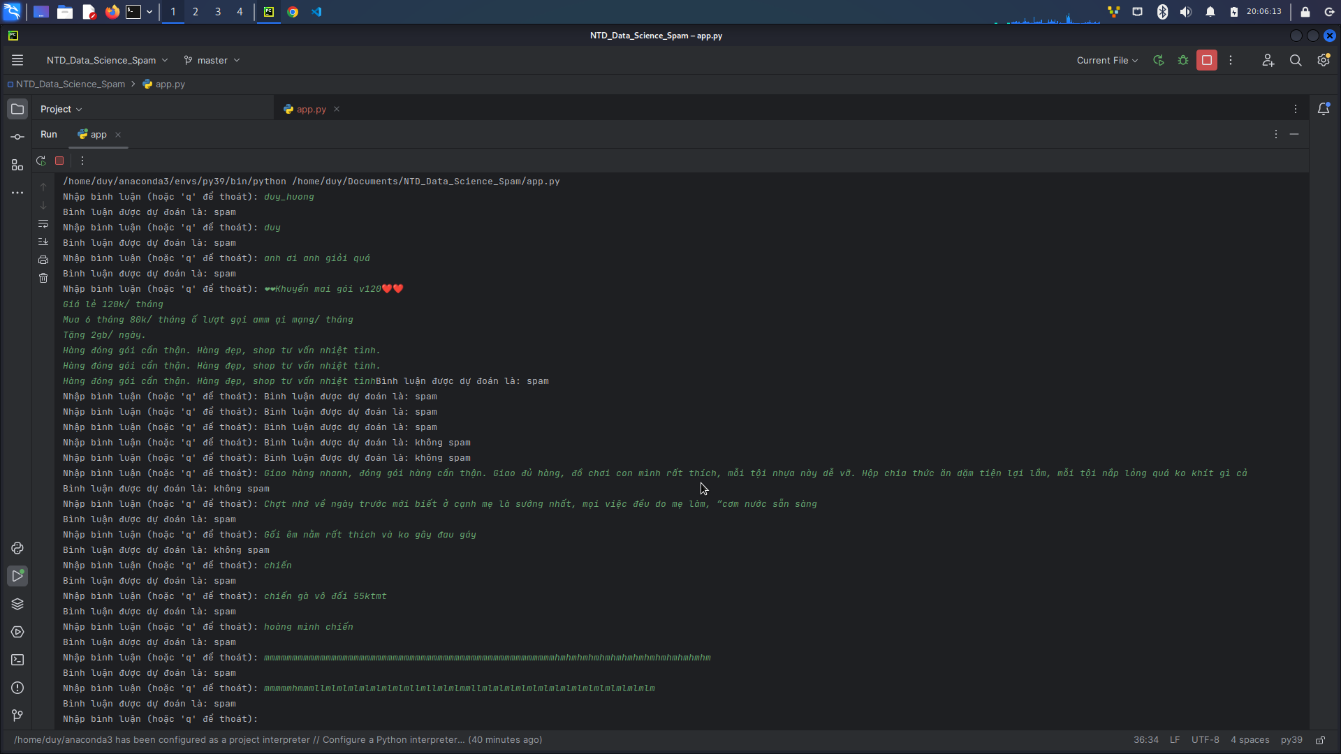




*Hình 9: File model.*

## 3.3 Demo

Sử dụng file model.pkl để kiểm thử, kết quả thu được.



*Hình 10: Kết quả chạy demo*

Nhìn chung kết quả thì đã phân biệt được bình luận spam và đâu là bình luận nào là không spam. Nhưng độ chính xác còn chưa cao.

# CHƯƠNG IV: KẾT LUẬN

Tổng kết lại, sử dụng mô hình PhoBERT để dự đoán bình luận spam. Kết quả thực nghiệm cho thấy mô hình đạt được hiệu suất tốt trong việc phân loại bình luận là spam hay không spam. Đây là một nền tảng để xây dựng các hệ thống tự động phát hiện và ngăn chặn bình luận spam trong các ứng dụng thực tế. Tuy nhiên, việc nghiên cứu và cải thiện các phương pháp phân loại spam vẫn là một lĩnh vực nghiên cứu tiềm năng và cần được tiếp tục khám phá.

**Ưu điểm:**

Hiệu suất cao: Mô hình PhoBERT đã được huấn luyện trên tập dữ liệu lớn và có khả năng hiểu và biểu diễn ngôn ngữ tiếng Việt tốt. Do đó, nó có thể mang lại hiệu suất tốt trong việc dự đoán bình luận spam.

Đa năng: Mô hình PhoBERT có thể áp dụng cho nhiều tác vụ xử lý ngôn ngữ tự nhiên khác nhau, không chỉ giới hạn trong việc dự đoán bình luận spam. Điều này tạo ra sự linh hoạt và tiềm năng ứng dụng rộng rãi.

Tiếng Việt tự nhiên: PhoBERT là một trong số ít các mô hình pre-trained dành riêng cho tiếng Việt, nên nó có khả năng xử lý và hiểu các ngữ cảnh và biểu đạt ngôn ngữ tiếng Việt tự nhiên một cách tốt.

**Nhược điểm:**

Đòi hỏi tài nguyên tính toán: Mô hình PhoBERT là một mô hình sâu có số lượng tham số lớn, do đó, việc huấn luyện và sử dụng mô hình này đòi hỏi tài nguyên tính toán cao, đặc biệt là khi áp dụng cho dữ liệu lớn.

Phụ thuộc vào dữ liệu huấn luyện: Mô hình PhoBERT yêu cầu một tập dữ liệu huấn luyện lớn và mang tính đại diện cao để đạt được hiệu suất tốt. Việc thiếu dữ liệu hoặc dữ liệu không đủ đại diện có thể ảnh hưởng đến khả năng dự đoán chính xác của mô hình.

Khó khăn trong việc giải thích kết quả: Mô hình PhoBERT là một mô hình mạng neural sâu, do đó, việc giải thích cơ chế ra quyết định của mô hình có thể gặp khó khăn. Điều này làm cho việc hiểu và giải thích lý do mô hình phân loại một bình luận là spam trở nên khó khăn.

**Phương hướng phát triển:**

Tăng cường tập dữ liệu huấn luyện: Một tập dữ liệu huấn luyện lớn và đa dạng là một yếu tố quan trọng để cải thiện hiệu suất của mô hình. Tiếp tục thu thập và gán nhãn các bình luận spam và không spam mới để làm giàu tập dữ liệu huấn luyện có thể cải thiện khả năng dự đoán của mô hình.

Cải tiến bước tiền xử lý dữ liệu:

* Loại bỏ dữ liệu không cần thiết.
* Xử lý từ đồng nghĩa và từ viết sai.
* Xử lý từ viết tắt và ngôn ngữ chat.
* Xử lý các từ văng tục.

Việc dự đoán một bình luận là spam hay không cần dựa vào rất nhiều yếu tố cũng như mục dích khác nhau nên mô hình còn cần rất nhiều cải tiến để đưa vào sử dụng. Nhưng có thể thấy mô hình đã đạt được một số kết quả đơn giản nhất có thể sử dụng thử nghiệm.

# TÀI LIỆU THAM KHẢO

<https://github.com/VinAIResearch/PhoBERT>

<https://viblo.asia/p/bert-roberta-phobert-bertweet-ung-dung-state-of-the-art-pre-trained-model-cho-bai-toan-phan-loai-van-ban-4P856PEWZY3>

<https://phamdinhkhanh.github.io/2020/06/04/PhoBERT_Fairseq.html>