BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC KINH TẾ TP HỒ CHÍ MINH
TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ

REPORT

ĐỀ TÀI: CRAWL DỮ LIỆU TỪ FACEBOOK, THỰC HIỆN ĐÁNH GIÁ CẢM XÚC BÀI VIẾT VÀ BÌNH LUẬN. XÂY DỰNG HỆ THỐNG ĐỀ XUẤT BÀI VIẾT CHO CÁ NHÂN VÀ THỰC HIỆN TÓM TẮT CÁC BÀI VIẾT.

Sinh Viên: Phan Trần Sơn Bảo
Chuyên Ngành: KHOA HỌC DỮ LIỆU
Khóa: K46

Giảng Viên: TS. Đặng Nhân Cách

TP. Hồ Chí Minh, Ngày 01 tháng 04 năm 2023

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU

1.1 GIỚI THIỆU VỀ ĐỀ TÀI.

1.1.1.Giới thiệu tổng quan.

- Trong thời đại số hoá ngày nay, dữ liệu là một tài nguyên quan trọng và không thể thiếu trong nhiều lĩnh vực để có thể đánh giá sản phẩm, hành vi khách hàng, giúp doanh nghiệp tăng thêm lợi nhuận,…. Và để có thể có được nguồn dữ liệu lớn thì các nền tảng mạng xã hội chính là những nơi giúp ta có được nguồn dữ liệu lớn đó.

- Chính vì vậy, việc crawl dữ liệu ngày nay đóng một vai trò rất quan trọng trong việc phân tích dữ liệu và xây dựng các mô hình để có thể đưa ra quyết định giúp cá nhân hoặc doanh nghiệp hoạt đồng hiệu quả và trở nên tốt hơn. Các công ty hay các tổ chức thường sử dụng dữ liệu để nghiên cứu thị trường, phân tích dữ liệu khách hàng hoặc của đối thủ cạnh tranh, xây dựng các hệ thống đề xuất hay tạo ra những sản phẩm phù hợp với nhu cầu của khách hàng để có thể tiếp cận và giữ lấy khách hàng dễ hơn.

- Tóm lại, crawl dữ liệu đã trở thành một phần không thể thiếu trong thời đại hiện nay bởi nó đóng vai trò rất quan trọng trong việc phân tích dữ liệu, đưa ra quyết định, theo dõi thông tin, tin tức, sự kiện trong thời gian thực để giúp cho các doanh nghiệp và cá nhân có thể đưa ra những quyết định chính xác và hiệu quả hơn.

1.1.2. Giới thiệu đề tài.

- Facebook là một trong các nền tảng xã hội phổ biến đối với mọi người ở Việt Nam và nó cũng là một nền tảng mạng xã hội tiềm năng để ta có được những dữ liệu về các quán ăn, những quan tâm của mọi người về một vấn đề, về mọi thứ… để ta có thể khai thác, phân tích và đánh giá nó. Chính vì vậy, em đã chọn Facebook để tiến hành cào dữ liệu về - cụ thể là cào các bài viết và bình luận cũng như các trường dữ liệu khác như tương tác, url bài viết, url ảnh, url video,… từ trang TheAnh28 Entertainment, và xử lí và phân tích Sentiemnt các dữ liệu (bài viết và bình luận) và phân tích reactions. Đồng thời, em cũng xây dựng riêng cho mình một Personal Recommender System để có thể có được những post liên quan đến vấn đề mà em quan tâm và thực hiện việc Summerization Text các post đã được recommend bằng Personal Recommender System.

1.2. MỤC ĐÍCH NGHIÊN CỨU

- Mục đích của đề tài trên là thành công trong việc cào dữ liệu từ Page TheAnh28 Entertainment trong Facebook, và tiến hành xử lí các dữ liệu đã được crawl để tiến hành phân tích cảm xúc của từng bài viết hay từng comment qua 2 model pre-trained, model pre-train PhoBert –base và model pre-train Bert-base cho đa ngôn ngữ. Sau khi đã có được kết quả phân tích cảm xúc của các bài viết và bình luận của 2 mô hình PhoBert-base và Bert-base cho đa ngôn ngữ thì sẽ tiến hành phân tích và kết luận mô hình nào hiệu quả hơn trong việc phân tích cảm xúc bài viết và bình luận trên page này và mở rộng là trên Facebook.

- Tiếp theo, em sẽ xây dựng một hệ thống đề xuất các bài viết riêng cho mình để có thể lấy được những bài viết và url của những bài viết đó cho những vấn đề mà em quan tâm.

- Cuối cùng, khi lấy được những bài viết thì em muốn tóm tắt nó lại để em chỉnh lấy được những ý chính của bài viết đó.

1.3. MỘT SỐ HƯỚNG TIẾP CẬN GIẢI QUYẾT BÀI TOÁN.

- Để có thể cào dữ liệu từ Facebook thì ta có thể xin API từ Facebook để cào dữ liệu từ chính nó nhưng vì phương án này có tốn chi phí để xin API và đợi facebook duyệt khá lâu nên em sẽ bỏ qua phương án này. Ngoài việc sử dụng API của Facebook để cào thì ta có thể sử dụng các thư viện khác như selenium, facebook scraper,.. và kết hợp sử dụng cookies để cào dữ liệu từ Facebook. Trong bài này, em sẽ sử dụng facebook scraper và cookies của page TheAnh28 Entertainment để tiến crawl từng bài post và các thuộc tính của bài post đó.

- Vì dữ liệu được cào không có nhãn nên sau khi có kết quả phân tích cảm xúc từ 2 mô hình pre-train thì em sẽ tiến hành chọn mô hình hiệu quả hơn trong việc phân tích cảm xúc bằng cách lấy 10 bài viết đầu tiên và đặt nhãn cho nó theo ý kiến riêng của bản thân mình sau đó so sánh 10 nhãn của 10 bài viết đầu tiên với 10 kết quả của từng model. Model nào có số nhãn giống với nhãn mà em tự cho là đúng thì em sẽ cho rằng model đó sẽ tốt hơn trong việc phân tích cảm xúc khi ta phân tích cảm xúc dữ liệu của Facebook.

- Đối với việc xây dựng mô hình đề xuất bài viết thì em sẽ tiến hành xây dựng mô hình để chọn ra các bài viết tương đồng với yêu cầu của em bằng cách gom các bài viết tương tự nhau thành một cụm. Sauk hi có được các cụm chứa các bài viết tương đồng thì em sẽ chọn ra cụm có các bài viết tương tự với yêu cầu mà em đưa ra, và output sẽ là các bài viết tương đồng với yêu cầu của em.

- Cuối cùng, sau khi đã có các bài viết tương đòng với yêu cầu mà em đưa ra thì em sẽ xây dựng mô hình để cho nó học

1.4. CÁC CÔNG CỤ VÀ THƯ VIỆN CẦN DÙNG.

- Một số thư viện cần dùng: facebook_scraper, numpy, pandas, nltk, sklearn, re, demoji, pyvi, torch, transformers…

- Một số model cần dùng: pre-train model của PhoBert-base, pre-train model của Bert-base cho đa ngôn ngữ, K-nearest neighbors,…

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÁC VẤN ĐỀ LIÊN QUAN

2.1. GIỚI THIỆU VỀ COOKIES.

- Cookies là một đoạn văn bản ghi thông tin được tạo ra và lưu trên trình duyệt của máy người dùng. Cookies giúp các trang web nhớ thông tin về người dùng, về những gì họ đã làm trên trang web đó.

- Các cookies có thể được sử dụng để giúp trang web hoạt động hiệu quả hơn bằng cách lưu trữ thông tin về người dùng, giảm thời gian phản hồi và cải thiện trải nghiệm người dùng. Các trang web có thể sử dụng cookies để lưu trữ thông tin đăng nhập của người dùng, như tên đăng nhập và mật khẩu, để người dùng không cần phải nhập lại thông tin này mỗi khi truy cập vào trang web. Tuy nhiên, các cookies cũng có thể được sử dụng để thu thập thông tin về người dùng và hoạt động của họ trên trang web.

- Chính vì vậy, nên em sẽ dùng cookies để có thể cào được dữ liệu của từng bài viết của page và các bình luận trong từng bài viết thông qua cookies của page.

2.2. TỔNG QUAN CÁC THƯ VIỆN QUAN TRỌNG.

2.2.1. Thư viện facebook_scraper.

- Thư viện facebook_scraper là một thư viện giúp ta trích xuất, cào dữ liệu từ facebook mà không cần dùng API từ facebook. Chính điều này cũng khiến cho việc cào dữ liệu từ facebook trở nên khá rủi ro và hạn chế vì việc trích xuất, cào dữ liệu từ facebook mà không có sư cho phép từ facebook sẽ vi phạm qui định của facebook về việc bảo mật và riêng tư của người dùng. Tuy không dùng API của facebook để trích xuất, cào dữ liệu từ nó nhưng ta vẫn cần phải có cookies của page ta cần cào thì ta mới có thể tiến hành cào dữ liệu của page đó trên facebook được.

- Tuy nhiên, việc trích xuất, cào dữ liệu từ facebook bằng thư viện facebook_scraper cũng gặp nhiều vấn đề khác như không thể cào được hết các giá trị của trường dữ liệu (ví dụ không thể cào hết các comment của bài viết, tuy nhiên vẫn sẽ cào được hết tất cả các bài viết đó) nếu số bài viết cào quá lớn và đôi khi việc cào dữ liệu liên tục sẽ bị facebook khóa tài khoản và mất tài khoản facebook.

2.2.2. Thư viện pyvi.

- Vì dữ liệu text mà em dùng để phân tích là dữ liệu ngôn ngữ tiếng Việt nên sẽ không thể sử dụng các thư viện nên em sẽ kết hợp thư viện nltk và re để có thể xử lí văn bản tiếng việt dễ dàng và chính xác hơn.

- Như đã nói ở trên thì pyvi là một thư viện dành riêng cho việc xử lí văn bản tự nhiên dành riêng cho tiếng Việt. Nó được phát triển bởi nhóm dự án Underthesea của trường đại học Công nghệ Thông tin – đại học Quốc gia thành phố Hồ Chí Minh. Nó là một thư viện mạnh và được sử dụng rộng rãi trong việc phân tích, xử lí ngôn ngữ tiếng Việt như: tách câu, chuyển đổi chữ hoa sang chữ thường, phân tích cú pháp, gán nhãn từ loại,…

2.2.3. Thư viện nltk.

- Thư viện nltk là một thư viện phổ biến trong việc tiền xử lí với dữ liệu ở định dạng văn bản. Nó là một thư viện cung cấp một bộ công cụ cho các việc xử lí dữ liệu văn bản như tách từ, phân tích cú pháp, chuẩn hóa, chuyển đổi, phân loại văn bản,... để ta có thể làm sạch được dữ liệu trước khi đưa dữ liệu vào các mô hình để tiến hành phân tích đánh giá. Đây là một thư viện hết sức cần thiết trong đề tài này vì đề tài này chủ yếu làm việc trên dữ liệu văn bản nên việc xử lí văn bản một cách hợp lí và tốt thì mới có thể cho vào các mô hình để huấn luyện vì khi xử lí được thành dữ liệu sạch thì các mô hình chạy mới cho ra kết quả tốt nhất được.

2.2.4. Thư viện sklearn.

- Thư viện sklearn là một thư viện nổi tiếng nhất trong lĩnh vực Machine Learning và Data Mining. Thư viện này nổi tiếng như vậy bới các tính năng nó đem lại khá quan trọng đối với những người sử dụng như: xây dựng các mô hình máy học, cung cấp các công cụ để ta có thể đánh giá các mô hình sau khi ta chạy mô hình, cung cấp các công cụ để ta có thể xử lí dữ liệu chuyển sang dạng vector hay giảm chiều dữ liệu,.. để khi đưa vào mô hình thì có thể chạy mô hình hiệu quả hơn. Chính vì những tính năng đó nên đây cũng là một thư viện quan trọng trong việc xây dựng các mô hình như recommender system hay text summerization trong đề tài này, vì 2 mô hình trên đều cần phải chuyển hóa dữ liệu sang dạng feature sau đó mới đưa vào các mô hình để huấn luyện mô hình và đưa ra kết quả.

2.2.5. Thư viện transformers.

- Thư viện Transformers là một thư viện mã nguồ mở cho việc xử lí ngôn ngữ tự nhiên được phát triển bởi Hugging Face. Thư viện này cho phép chúng ta tạo, huấn luyện lại hay sử dụng các mô hình như Bert, GPT-2,…

- Ta có thể sử dụng thư viện transformers cho nhiều mục đích khác nhau như phân loại văn bản, dò tìm thực thể hay tóm tắt văn bản và phát sinh văn bản… Ngoài ra, thư viện này cũng cung cấp các công cụ để fine-tuning các mô hình NLP trên các tập dữ liệu đặc biệt của ta và để thực hiện các nhiệm vụ NLP với các mô hình đã được đào tạo sẵn.

- Ngoài ra, Transformers cũng cho phép chúng ta chia sẻ và tải các mô hình NLP đã được đào tạo sẵn từ một thư viện mô hình được lưu trữ trực tuyến, được gọi là Hugging Face Model Hub. Điều này giúp cho việc sử dụng các mô hình NLP đã được đào tạo sẵn trở nên dễ dàng và nhanh chóng hơn.

- Chính vì vậy, đây sẽ là một thư viện quan trọng trong đề tài này, vì nó cho phép dùng 2 mô hình đã được huấn luyện sẵn mà em sẽ dùng để phân tích cảm xúc văn bản là mô hình Bert cho đa ngôn ngữ và mô hình PhoBert được phát triển cho tiếng Việt để phân tích cảm xúc văn bản tiếng Việt.

2.3. CÁC MÔ HÌNH ĐƯỢC SỬ DỤNG.

2.3.1. Mô hình pre-train PhoBert.

- PhoBert là một mô hình học sâu pre-trained - được huấn luyện chỉ dành riêng cho tiếng Việt. PhoBert được xây dựng dựa trên kiến trúc mã hóa của Bert, và nó được huấn luyện và đã được kết quả tốt để có thể giải quyết các vấn đề ngôn ngữ tự nhiên như phân tích cảm xúc văn bản một cách tốt nhất. Ngoài việc phân tích cảm xúc, PhoBert còn là một mô hình ta có thể dùng để thực hiện các tác vụ như phân loại văn bản hay dịch máy,...

- PhoBERT được phát triển bởi nhóm nghiên cứu của trường Đại học Công nghệ - Đại học Quốc gia Hà Nội và VinAI Research. Mô hình này có 2 phiên bản khác nhau: PhoBert-base và PhoBert-large, với số lượng tham số và độ chính xác khác nhau. Đối với PhoBert-base thì nó dùng để thực hiện các tác vụ đơn giản như phân tích cảm xúc văn bản, phân loại văn bản, tìm kiếm thông tin,... Tuy nhiên, đối với PhoBert-large thì ta có thể thực hiện các tác vụ khó hơn như phân tích ngữ cảnh, hiểu bài đọc tự động,..

- Đối với đề tài này thì em sẽ chọn mô hình PhoBert-base để tiến hành đánh giá cảm xúc văn bản.

2.3.1.1. Mô hình pre-train PhoBert-base.

- PhoBert-base là một mô hình ngôn ngữ tiếng Việt dựa trên mô hình BERT, nó được huấn luyện trên một lượng lớn dữ liệu tiếng Việt. Mô hình này có kích thước tương đối nhỏ với 110 triệu tham số. PhoBERT-base được huấn luyện trên 3 nguồn dữ liệu tiếng Việt khác nhau: Wikipedia, tờ báo và tiểu thuyết. Với kích thước dữ liệu lên tới 30GB, mô hình này đã học được rất nhiều tri thức ngôn ngữ Việt Nam và có thể đáp ứng các yêu cầu NLP cơ bản như phân tích cảm xúc văn bản, phân loại văn bản, ...

- Kiến trúc của PhoBert-base bao gồm 12 lớp Transformer và các lớp Fully Connected để đưa ra dự đoán cuối cùng. Mô hình này sử dụng kỹ thuật Fine-tuning để có thể áp dụng vào các tác vụ NLP khác nhau. Ngoài ra, mô hình này còn có thể thực hiện các công việc khác như tách từ, chuyển đổi chữ hoa/thường và mã hóa văn bản.

- Một số ứng dụng của PhoBert-base đã được triển khai trong thực tế như chatbot, hệ thống phân tích ý kiến khách hàng trên mạng xã hội và hệ thống gợi ý sản phẩm. PhoBert-base cũng được cung cấp dưới dạng một API mở trên nền tảng Transformers của Hugging Face, cho phép các nhà phát triển tích hợp mô hình này vào các ứng dụng của họ.

2.3.2. Mô hình pre-train Bert-base cho đa ngôn ngữ.

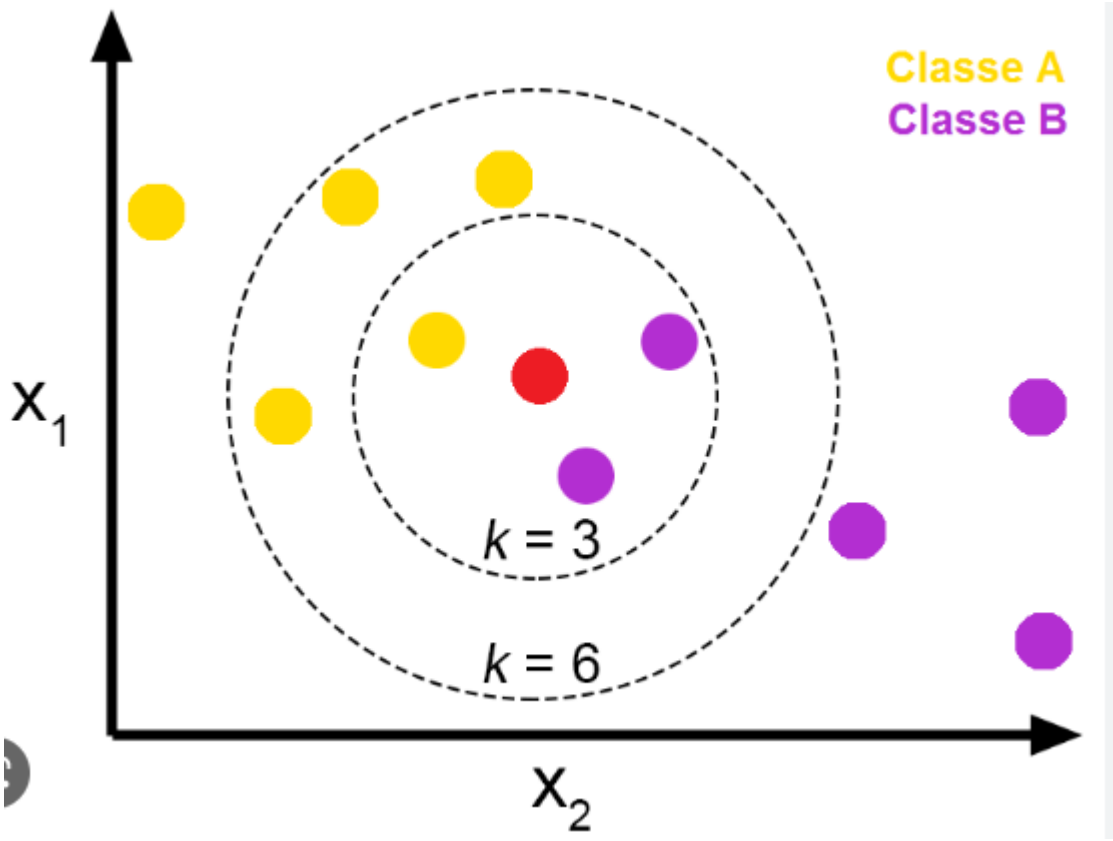
- Bert-base cho đa ngôn ngữ là một mô hình pre-trained trên dữ liệu lớn của Google, được thiết kế để hỗ trợ nhiều ngôn ngữ khác nhau. Mô hình này được xây dựng dựa trên kiến trúc Transformer, có khả năng học các mối quan hệ phức tạp giữa các từ trong văn bản và xử lý nhiều ngôn ngữ khác nhau cùng một lúc. Bert-base cho đa ngôn ngữ có khả năng học được cách dùng ngôn ngữ khác nhau và xử lý các câu văn đa ngôn ngữ.

- Mô hình Bert-base multilingual có thể được sử dụng để giải quyết nhiều nhiệm vụ ngôn ngữ khác nhau, bao gồm phân loại văn bản, dịch máy, trích xuất thông tin và tóm tắt văn bản. Vì mô hình này được huấn luyện trên nhiều ngôn ngữ, nó có thể hỗ trợ các dự án đa ngôn ngữ và giảm thiểu sự cần thiết của các mô hình đặc biệt được huấn luyện cho mỗi ngôn ngữ.

2.3.3. Mô hình K-Nearest Neighbors.

- Mô hình K-Nearest Neighbors (KNN) là một mô hình học máy đơn giản và phổ biến được sử dụng trong các bài toán phân loại và dự đoán. Mô hình KNN không có quá trình huấn luyện, mà chỉ cần phân loại hoặc dự đoán dựa trên các điểm dữ liệu đã được phân loại trước đó.

- Cơ chế hoạt động của KNN là: KNN sẽ tìm kiếm K điểm dữ liệu gần nhất (từ khoảng cách Euclidean, khoảng cách Mahalanobis hoặc khoảng cách cosine, ...) với điểm dữ liệu cần phân loại/dự đoán, và sau đó áp dụng phương pháp đa số phiếu để xác định lớp của điểm dữ liệu đó. Điều này có nghĩa là lớp được chọn cho điểm dữ liệu cần phân loại/dự đoán là lớp xuất hiện nhiều nhất trong các K điểm dữ liệu gần nhất.



- Mô hình KNN có thể được sử dụng cho cả bài toán phân loại và dự đoán. Để sử dụng KNN cho bài toán phân loại, ta cần xác định lớp của các điểm dữ liệu trong tập huấn luyện trước đó, trong khi đối với bài toán dự đoán, ta cần xác định giá trị mong muốn của biến mục tiêu cho các điểm dữ liệu trong tập huấn luyện trước đó.

- Ưu điểm của KNN là đơn giản và dễ hiểu, cũng như có thể được áp dụng cho nhiều loại dữ liệu khác nhau. Tuy nhiên, KNN có thể trở nên rất chậm khi số lượng điểm dữ liệu lớn và thời gian tính toán để tìm các điểm gần nhất sẽ tăng lên đáng kể. Điều này cũng có nghĩa là việc lưu trữ toàn bộ tập dữ liệu cần thiết cho mô hình KNN có thể rất tốn kém và không thực tế.

- Nhược điểm của KNN: Chi phí tính toán liên quan cao vì nó lưu trữ tất cả dữ liệu đào tạo, yêu cầu bộ nhớ lưu trữ cao, cần xác định giá trị của K, dự đoán chậm nếu giá trị của N cao, nhạy cảm với các tính năng không liên quan.

2.3.3.1. Khoảng cách cosine.

- Khoảng cách cosine là phương pháp đo độ tương đồng giữa hai vectơ trong không gian đa chiều và nó được tính theo công thức sau:

$$cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}||||\vec{b}||}$$

- Các bước để thực hiện Khoảng cách cosine KNN như sau:

- Tính toán khoảng cách cosine giữa điểm dữ liệu mới và tất cả các điểm dữ liệu trong tập huấn luyện.
- Tìm k điểm dữ liệu gần nhất với điểm dữ liệu mới dựa trên khoảng cách cosine.
- Lấy phân loại của các điểm dữ liệu gần nhất để phân loại điểm dữ liệu mới.

- Khoảng cách cosine KNN có ưu điểm là nó giảm thiểu hiện tượng "chiều dài vector ảo" trong không gian đa chiều và giúp cải thiện độ chính xác của KNN trong các bài toán phân loại và gợi ý dữ liệu.

2.3.4. Mô hình T5.

- Mô hình T5 (Text-to-Text Transfer Transformer) là một mô hình xử lý ngôn ngữ tổng quát và mạnh mẽ được đề xuất bởi Google AI Language vào năm 2019. Mô hình T5 là một kiến trúc transformer, được xây dựng trên cơ sở của kiến trúc transformer lớn nhất hiện nay - GPT-2.

- Mô hình T5 được huấn luyện để giải quyết nhiều tác vụ xử lý ngôn ngữ tổng quát khác nhau bằng cách chuyển đổi đầu vào văn bản sang đầu ra văn bản thông qua một loạt các phép chuyển đổi khác nhau. Các tác vụ mà mô hình T5 có thể giải quyết bao gồm dịch máy, tổng hợp văn bản, trả lời câu hỏi, tóm tắt văn bản, và nhiều tác vụ khác.

- Một ưu điểm của T5 là nó không chỉ là một mô hình dự đoán mà còn là một mô hình tổng quát để xử lý nhiều loại tác vụ khác nhau. Điều này cho phép T5 được sử dụng để giải quyết các vấn đề xử lý ngôn ngữ tổng quát mà không cần phải huấn luyện riêng cho từng tác vụ.

- Mô hình T5 đã đạt được nhiều kết quả tốt trên các tập dữ liệu khác nhau và được sử dụng rộng rãi trong các ứng dụng xử lý ngôn ngữ như tóm tắt văn bản, trả lời câu hỏi, dịch máy và tổng hợp văn bản.

2.4. PHƯƠNG PHÁP TF-IDF.

- TF-IDF (hay Term Frequency - Inverse Document Frequency) giúp xác định mức độ quan trọng của một từ trong một tài liệu bằng cách so sánh sự xuất hiện của từ đó trong tài liệu đó với tần suất xuất hiện của từ đó trong tập hợp các tài liệu khác. Kỹ thuật này được sử dụng để giải quyết các vấn đề như tìm kiếm thông tin, phân loại văn bản và gợi ý sản phẩm.

- Term frequency: trong document d, frequency (tần số) biểu diễn số lần xuất hiện của từ t. Trọng số của từ xuất hiện trong document.

$$tf(t,d) = \frac{\text{count of t in d}}{\text{number of words in d}}$$

- IDF (invert document frequency) dùng để đánh giá múc độ quan trọng của 1 từ trong băn bản. Khi tính tf mức độ quan trọng của các từ coi là như nhau. Tuy nhiên trong văn bản thường xuất hiện nhiều từ không quan trọng xuất hiện với tần suất cao:

$$idf(t,D) = \log \frac{|D|}{|d \in D : t \in d|}$$

- TF-IDF là một trong những metric tốt nhất để xác định độ quan trọng của từ trong một đoạn text (document) trong một corpus. TF-IDF là hệ thống trọng số cái mà gán trọng số cho mỗi từ trong document dựa trên term frequency (tf) và document frequency (idf). Từ có weight cao hơn sẽ có ý nghĩa nhiều hơn.

- Nó tính toán mức độ quan trọng của một từ trong một tài liệu bằng cách tính toán hai yếu tố: tần suất của từ đó trong tài liệu và tần suất của từ đó trong tập hợp các tài liệu khác. Cụ thể, TF-IDF được tính bằng cách nhân hai giá trị sau đây:

$$\text{tfidf}(t,d,D) = \text{tf}(t,d) \times \text{idf}(t,D)$$

- TF là tần số xuất hiện của từ trong tài liệu.
- IDF là tần suất nghịch đảo của từ trong tập hợp các tài liệu.

CHƯƠNG 3. ĐỀ XUẤT MÔ HÌNH TRIỂN KHAI

3.1. GIỚI THIỆU VỀ DATASET VÀ CÁC BƯỚC TIỀN XỬ LÍ.

3.1.1. Các bước cào dữ liệu.

- Trước tiên, ta cần phải có cookies của page ta cần cào (Page TheAnh28 Entertainment). Ta có thể lấy cookies của page TheAnh28 Entertainment thông qua extension get cookies của Chrome và tải nó về dưới dạng file json.

- Sau khi đã có được cookies của page TheAnh28 Entertainment ta sẽ tiến hành install và khai báo thư viện facebook_scrapper để tiến hành việc cào dữ liệu từ page này. Trong phần options ta sẽ được tùy chỉnh các tham số như comments, reactions, allow_extra_requests,... để có thể tùy ý cào dữ liệu facebook theo ý muốn. Và tùy chọn số pages để có thể cào được số bài viết khác nhau. Lưu ý: không nên cào quá nhiều bài viết vì việc cào như thế sẽ không đảm bảo được các trường dữ liệu như comments_full khi cào sẽ có đầy đủ giá trị vì tính năng hạn chế của Facebook, ngoài ra việc cào quá nhiều cũng sẽ khiến Facebook khóa tài khoản chúng ta.

```
FANPAGE_LINK ="Theanh28"
FOLDER_PATH = "/content/drive/MyDrive/Project/Crawldata/"
COOKIE_PATH = "/content/drive/MyDrive/Project/Crawldata/cookies.json"

post_list = []
for post in get_posts(FANPAGE_LINK,
                      options={
        "comments": True,
        "reactions": True,
        "allow_extra_requests": True,
    }, extra_info=True, pages=300, cookies=COOKIE_PATH):
    post_list.append(post)
```

```
Kết quả truyền trực tuyến bị cắt bớt đến 5000 dòng cuối.
ERROR:facebook_scraper.extractors:Unable to parse comment <Element 'div' class=('_2a_i',) data-store='{"token":"7281125485266642_725133252320997"}' id='725133252320997' data-uniqueid
ERROR:facebook_scraper.extractors:Unable to parse comment <Element 'div' class=('_2a_i',) data-store='{"token":"7281125485266642_1988346394309 81"}' id='1988346394309 81' data-uniqueid
ERROR:facebook_scraper.extractors:Unable to parse comment <Element 'div' class=('_2a_i',) data-store='{"token":"7281125485266642_997037924827748"}' id='997037924827748' data-uniqueid
ERROR:facebook_scraper.extractors:Unable to parse comment <Element 'div' class=('_2a_i',) data-store='{"token":"7281125485266642_1253288725586455"}' id='1253288725586455' data-unique
ERROR:facebook_scraper.extractors:Unable to parse comment /comment/replies/?ctoken=7281125485266642_5868114796559400&count=19&curr&pc=1&isinline&initcomp&ft_ent_identifier=pfbid02fAv
ERROR:facebook_scraper.extractors:Unable to parse comment /comment/replies/?ctoken=7281125485266642_1271119736468 42&count=6&curr&pc=1&isinline&initcomp&ft_ent_identifier=pfbid02fAvV
WARNING:facebook_scraper.extractors:[7280933791952478] Extract method extract_video_meta didn't return anything
WARNING:facebook_scraper.extractors:[7280933791952478] Extract method extract_factcheck didn't return anything
WARNING:facebook_scraper.extractors:[7280933791952478] Extract method extract_share_information didn't return anything
WARNING:facebook_scraper.extractors:[7280933791952478] Extract method extract_listing didn't return anything
WARNING:facebook_scraper.extractors:[7280933791952478] Extract method extract_with didn't return anything
Traceback (most recent call last):
  File "/usr/local/lib/python3.9/dist-packages/facebook_scraper/utils.py", line 279, in safe_consume
    for item in generator:
  File "/usr/local/lib/python3.9/dist-packages/facebook_scraper/extractors.py", line 708, in extract_reactors
    data = json.loads(response.text[prefix_length:])  # Strip 'for (;;);'
  File "/usr/lib/python3.9/json/__init__.py", line 346, in loads
    return _default_decoder.decode(s)
  File "/usr/lib/python3.9/json/decoder.py", line 337, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
  File "/usr/lib/python3.9/json/decoder.py", line 355, in raw_decode
    raise JSONDecodeError("Expecting value", s, err.value) from None
json.decoder.JSONDecodeError: Expecting value: line 1 column 2 (char 1)
```

- Sau khi đã có được dữ liệu cào về, ta thu được các trường dữ liệu như sau:

```
<class 'dict'>
dict_keys(['post_id', 'text', 'post_text', 'shared_text', 'original_text', 'time', 'timestamp', 'image', 'image_lowquality', 'images', 'images_description',
'images_lowquality', 'images_lowquality_description', 'video', 'video_duration_seconds', 'video_height', 'video_id', 'video_quality', 'video_size_MB',
'video_thumbnail', 'video_watches', 'video_width', 'likes', 'comments', 'shares', 'post_url', 'link', 'links', 'user_id', 'username', 'user_url', 'is_live',
'factcheck', 'shared_post_id', 'shared_time', 'shared_user_id', 'shared_username', 'shared_post_url', 'available', 'comments_full', 'reactors', 'w3_fb_url',
'reactions', 'reaction_count', 'with', 'page_id', 'sharers', 'image_id', 'image_ids', 'was_live', 'fetched_time'])
```

- Tiếp theo, ta chuyển đổi dữ liệu đã được cào về thành 1 dataframe có tên post_df_full để tiện lợi cho việc xử lí và phân tích cho đề tài.

```python
# Initialize dataframe to scrape Facebook post
post_df_full = pd.DataFrame(columns = [])
# Start to collect Facebook post data by facebook_scraper library
for post in post_list:
    post_entry = post
    fb_post_df = pd.DataFrame.from_dict(post_entry, orient='index')
    fb_post_df = fb_post_df.transpose()
    post_df_full = post_df_full.append(fb_post_df)
    print(post['post_id']+' get')
```

3.1.2. Giới thiệu về dataset.

- Sau khi cào được dữ liệu các bài viết, bình luận và các trường thông tin khác từ Facebook thì ta có được 1 dataframe như sau:

	post_id	text	post_text	shared_text	original_text	time	timestamp	image	image_lowquality	images	...	with	page_id
0	7341596765886180	KIẾM GẦN 9 TRIỆU ĐỒNG/NGÀY\n\nTheo chia sẻ của...	KIẾM GẦN 9 TRIỆU ĐỒNG/NGÀY\n\nTheo chia sẻ của...		None	2023-03-31 08:30:25	1680251425	https://scontent-qro1-2.xx.fbcdn.net/v/t39.308...	https://scontent-qro1-2.xx.fbcdn.net/v/t39.308...	[https://scontent-qro1-2.xx.fbcdn.net/v/t39.30...	...	None	1509435412435707
0	7341544862558037	ĐỒNG CON HƠN CÁ LẠC LONG QUÂN\n\nTheo hướng dẫ...	ĐỒNG CON HƠN CÁ LẠC LONG QUÂN\n\nTheo hướng dẫ...		None	2023-03-31 08:09:31	1680250171	https://scontent-qro1-2.xx.fbcdn.net/v/t39.308...	https://scontent-qro1-2.xx.fbcdn.net/v/t39.308...	[https://scontent-qro1-2.xx.fbcdn.net/v/t39.30...	...	None	1509435412435707
0	7341504075895449	LỚP HỌC NHIỀU NGƯỜI MƠ ƯỚC, AI CŨNG XỨNG ĐÁNG ...	LỚP HỌC NHIỀU NGƯỜI MƠ ƯỚC, AI CŨNG XỨNG ĐÁNG ...		None	2023-03-31 07:54:09	1680249249	https://scontent-qro1-2.xx.fbcdn.net/v/t39.308...	https://scontent-qro1-2.xx.fbcdn.net/v/t39.308...	[https://scontent-qro1-2.xx.fbcdn.net/v/t39.30...	...	None	1509435412435707
0	7341432399235950	NGHI PHẠM ÂM TÍNH VỚI CHẤT CẤM VÀ HOÀN TOÀN TỈ...	NGHI PHẠM ÂM TÍNH VỚI CHẤT CẤM VÀ HOÀN TOÀN TỈ...		None	2023-03-31 07:27:04	1680247624	https://scontent-qro1-2.xx.fbcdn.net/v/t39.308...	https://scontent-qro1-2.xx.fbcdn.net/v/t39.308...	[https://scontent-qro1-2.xx.fbcdn.net/v/t39.30...	...	None	1509435412435707
0	7341374382575085	CẢNH BÁO: HÀNG LOẠT CÁC BẠN TRẺ BỊ L.OẠN T.HẦN...	CẢNH BÁO: HÀNG LOẠT CÁC BẠN TRẺ BỊ L.OẠN T.HẦN...		None	2023-03-31 07:05:28	1680246328	https://scontent-qro1-2.xx.fbcdn.net/v/t39.308...	https://scontent-qro1-2.xx.fbcdn.net/v/t39.308...	[https://scontent-qro1-2.xx.fbcdn.net/v/t39.30...	...	None	1509435412435707

5 rows × 54 columns

- Bao gồm 54 trường dữ liệu, và các trường dữ liệu thì số giá trị dữ liệu mà ta cào được chênh lệch nhau khá nhiều bởi vì có thể là bài viết đó không có các trưởng đó nên ta không thể cào được, và cũng có thể vì Facebook hạn chế nên khi cào dữ liệu ta không thể cào hết được mà chỉ cào một phần thôi.

```
 0   post_id                          1196 non-null   object
 1   text                             1071 non-null   object
 2   post_text                        1071 non-null   object
 3   shared_text                      1069 non-null   object
 4   original_text                    0 non-null      object
 5   time                             1196 non-null   datetime64[ns]
 6   timestamp                        1183 non-null   object
 7   image                            872 non-null    object
 8   image_lowquality                 1196 non-null   object
 9   images                           1196 non-null   object
 10  images_description               1196 non-null   object
 11  images_lowquality                1196 non-null   object
 12  images_lowquality_description    1196 non-null   object
 13  video                            323 non-null    object
 14  video_duration_seconds           0 non-null      object
 15  video_height                     0 non-null      object
 16  video_id                         323 non-null    object
 17  video_quality                    0 non-null      object
 18  video_size_MB                    0 non-null      object
 19  video_thumbnail                  323 non-null    object
 20  video_watches                    0 non-null      object
 21  video_width                      0 non-null      object
 22  likes                            877 non-null    object
 23  comments                         1196 non-null   object
 24  shares                           1196 non-null   object
 25  post_url                         1196 non-null   object
 26  link                             0 non-null      object
 27  links                            1196 non-null   object
 28  user_id                          1196 non-null   object
 29  username                         1196 non-null   object
 30  user_url                         1196 non-null   object
 31  is_live                          1196 non-null   object
 32  factcheck                        0 non-null      object
 33  shared_post_id                   4 non-null      object
 34  shared_time                      4 non-null      datetime64[ns]
 35  shared_user_id                   4 non-null      object
```

```
 36  shared_username                  4 non-null      object
 37  shared_post_url                  4 non-null      object
 38  available                        1196 non-null   object
 39  comments_full                    1196 non-null   object
 40  reactors                         319 non-null    object
 41  w3_fb_url                        319 non-null    object
 42  reactions                        319 non-null    object
 43  reaction_count                   1196 non-null   object
 44  with                             8 non-null      object
 45  page_id                          1196 non-null   object
 46  sharers                          0 non-null      object
 47  image_id                         749 non-null    object
 48  image_ids                        1196 non-null   object
 49  was_live                         1196 non-null   object
 50  fetched_time                     319 non-null    datetime64[ns]
 51  header                           8 non-null      object
 52  video_ids                        4 non-null      object
 53  videos                           4 non-null      object
dtypes: datetime64[ns](3), object(51)
memory usage: 513.9+ KB
```

- Tuy nhiên, với đề tài trên thì ta chỉ quan tâm tới các trưởng dữ liệu chính như: post_text/text, comments_full và reactions thì ta đã cào được khá nhiều và tạm đủ dùng để ta có thể tiến hành phân tích cảm xúc văn bản. Ngoài ra, ta còn có thể cần post_url để sau

khi được đề xuất bài viết và tóm tắt nó lại thì ta sẽ dùng url của bài viết đó để đánh giá xem thử mô hình có hiệu quả không.

- 3 trường dữ liệu chính mà ta sẽ làm việc thì đều thuộc kiểu dữ liệu object (string) nên ta sẽ cần phải tiền xừ lí một cách hợp lí và hiệu quả thì mới chạy mô hình tốt được.

- Tuy nhiên, khi nhìn vào cột reaction thì ta thấy nó đang là một dictionary chứa các tương tác với số tương tác của từng reaction. Chính vì vậy, để có thể phân tích được reactions thì ta sẽ cần chuyển đổi từng tương tác thành từng cột riêng biệt.

3.1.3. Các bước tiền xử lí.

3.1.3.1. Đối với reactions.

- Vì cột reactions đang chứa các giá trị giống với dạng dictionary chứa các tương tác và tổng số của từng tương tác nên ta sẽ chuyển từng giá trị của cột reactions thành 1 dictionary, đối với những cột có giá trị NaN thì sẽ thay thế nó bằng một dictionary rỗng.

- Sau đó, chuyển đổi từng tương tác của dictionary thành một cột riêng biệt, trong đó tên cột là các tương tác - các key của dictionary, còn số các tương tác sẽ là value của dictionary đó. Đoạn code dưới đây sẽ giúp ta làm điều đó:

```
post_df_full1['reactions'] = post_df_full1['reactions'].apply(lambda x: {} if pd.isna(x) else dict(eval(x)))
post_df_full_reactions = post_df_full1['reactions'].apply(pd.Series)
```

- Sau khi đã có các cột tương tác riêng biệt rồi thì ta sẽ tiến hành gộp nó lại với dataframe ban đầu bằng phương thức concat.

```
post_df_full_with_reactions = pd.concat([post_df_full1, post_df_full_reactions], axis=1).drop('reactions', axis=1)
post_df_full_with_reactions[['post_text','thích','yêu thích','haha','wow','buồn','phẫn nộ','thương thương',
                             'shares','comments','reaction_count']]
```

- Kết quả ta đạt được sẽ là một dataframe chứa đầy đủ thông tin của từng loại tương tác như thích, yêu thích, haha, wow,… cho từng bài viết như sau:

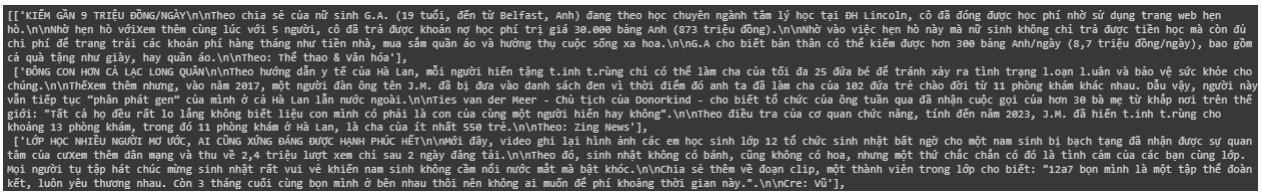
	post_text	thích	yêu thích	haha	wow	buồn	phẫn nộ	thương thương	shares	comments	reaction_count
0	KIẾM GẦN 9 TRIỆU ĐỒNG/NGÀY\n\nTheo chia sẻ của...	2704.0	15.0	1336.0	71.0	118.0	1.0	6.0	27	647	4251
1	ĐÔNG CON HƠN CẢ LẠC LONG QUÂN\n\nTheo hướng dẫ...	4165.0	17.0	3148.0	63.0	150.0	1.0	5.0	12	386	7549
2	LỚP HỌC NHIỀU NGƯỜI MƠ ƯỚC, AI CŨNG XỨNG ĐÁNG ...	8174.0	3717.0	29.0	3.0	39.0	NaN	150.0	13	256	12112
3	NGHỊ PHẠM ÂM TÍNH VỚI CHẤT CẤM VÀ HOÀN TOÀN TỈ...	7844.0	14.0	30.0	23.0	1165.0	121.0	8.0	47	1148	9205
4	CẢNH BÁO: HÀNG LOẠT CÁC BẠN TRẺ BỊ L.OẠN T.HẦN...	4834.0	18.0	94.0	26.0	770.0	1.0	3.0	149	1261	5746
...	...	...	...	...	...	...	...	...	...	...	...
1191	Hoa hậu Trái Đất 2018 - Nguyễn Phương Khánh đế...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
1192	Chỉ trong ngày 8/3, đã có 10 triệu người thử l...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
1193	RICHARLISON THẤT VỌNG VÌ HLV CHO NGỒI DỰ BỊ\n\...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
1194	SIÊU LẦY GẶP SIÊU LỪA ĐÃ CÁN MỐC 50 TỶ ĐỒNG\n\...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0
1195	NỖI ĐAU NHÂN ĐÔI, VỪA MẤT TIỀN VỪA MẤT TÌNH.....	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0

3.1.3.2. Đối với dữ liệu dạng văn bản (string data).

- Ta sẽ tiến hành tiền xử lí văn bản đối với dữ liệu text sao cho hợp lí và tùy vào tính huống mà ta sẽ xử lí nó theo những cách khác nhau.

3.1.3.2.1. Tiền xử lí bài viết.

- Trước tiên, ta sẽ nhìn sơ qua dữ liệu các bài viết.



- Ta thấy các bài viết dài thường khi cào dữ liệu sẽ bị dính thêm chữ ‘Xem thêm’ nên ta sẽ cần phải loại bỏ nó. Tiếp theo, thì các bài viết có xuất hiện dấu xuống dòng ‘\n’ khá nhiều nên ta cũng cần phải xử lí nó, và một số bài viết thì có liên quan đến vấn đề nhạy cảm nên thường tác giả bài viết sẽ viết tên của những người liên quan theo định dạng ‘A.B.C.D’ hay những từ ngữ nhạy cảm ‘tự tử’ sẽ ghi là ‘t.ự t.ử’ nên ta sẽ loại bỏ đi dấu ‘.’ Để khi gặp các từ ngữ nhạy cảm ở định dạng như thế thì nó sẽ quay trở lại dạng nguyên gốc, còn đối với tên thì ta sẽ chấp nhận việc mô hình được triển khai sẽ không thể hiểu nó bởi vì ta cũng không biết tên gốc của từng người là như thế nào.

- Ngoài ra, trong các bài viết thường sẽ có một vài từ viết tắt, và sẽ có một vài bài viết sẽ đính kèm thêm các đường link, ghi thêm cre, theo, nguồn, video, ảnh,.. được lấy từ một nơi nào đó. Nên ta cần sẽ phải loại bỏ đi hết những thứ này để bài viết trở nên ‘sạch’ hơn. Tiếp theo, ta sẽ loại bỏ đi những hastag, punctuation, số và các từ dừng vì nó không quan trọng trong việc phân tích cảm xúc văn bản nên ta cũng sẽ loại bỏ nó. Đồng thời, ta cũng sẽ loại bỏ các emoji vì các emoji này không có giá trị trong việc đánh giá cảm xúc văn bản.

- Dưới đây là một số từ viết tắt mà em tự ghi ra vì không tìm ra được cách xử lí nó nên em tự ghi tay và thay thế nó với từ hoàn chỉnh:

- Tuy nhiên, khi thay đổi từ viết tắt thành từ hoàn chỉnh của nó thì ta cũng nên xem xét ngữ cảnh lúc đó như thế nào, và từ viết tắt này có đúng với mọi người hay không thì ta

mới có thể thay thế nó thành từ hoàn chỉnh được để tránh tình trạng từ được thay thế ở trong một hoàn cảnh không đúng.

- Đây là đoạn code để thực hiện việc xử lí văn bản cho bài viết:

```
def preprocess_vietnamese_text(text):

    text = re.sub(r'(\w+)Xem thêm (\w+)', r'\1 \2', text)
    # lowercase the text
    text = text.lower()

    #replace word with this format 'a.b' with 'ab'
    text = re.sub(r'(\w)\.(\w)', r'\1\2', text)
    text = text.replace('/', ' ')
    # replace the abbreviations
    for abbr, full in abbreviations.items():
        text = re.sub(rf"\b{abbr}\b", full, text)

    text = re.sub(r'\b(cre|theo|nguồn|video|hình ảnh):\s*\S+.*', '', text)

    # remove accents
    #text = unidecode.unidecode(text)

    # remove emojis
    text = demoji.replace(text, '')

    # remove hastag
    text = re.sub(r'#\w+\b', '', text)

    #remove link
    text = re.sub(r'\b(?:https?://|www\.|bit\.ly/)\S+\b', '', text)
    text = re.sub(r'\b\S*\.com\S*\b', '', text)

    # remove punctuation
    text = re.sub(r'[^\w\s]+', '', text)

    #remove number
    text = re.sub(r'\d+', '', text)

    # tokenize the text
    tokens = ViTokenizer.tokenize(text)

    # remove stop words
    tokens = [word for word in word_tokenize(tokens) if word not in stop_words]

    # join the tokens back into a single string
    text = " ".join(tokens)
    text = text.replace('_', ' ')
    return text
```

3.1.2.2.1. Tiền xử lí bình luận.

- Các bước tiền xử lí bình luận cũng gần như tương tự với việc tiền xử lí bài viết nhưng ta sẽ bỏ đi việc loại bỏ không cần thiết như loại bỏ từ ‘Xem thêm’.

- Để tiến hành phân tích cảm xúc các bình luận thì em sẽ chọn ra những bài viết có các từ liên quan về bóng đá Messi và Ronaldo và lấy các bình luận của các bài viết đó để tiền xử lí và phân tích cảm xúc các bình luận của bài viết.

```python
import pandas as pd

topics = ['man united', 'bóng đá', 'real madrid', 'juventus','psg' , 'barcelona',
          'leo messi', 'christiano ronaldo', 'manchester united', 'bóng đá',
          'ngoại hạng anh', 'liverpool', 'juventus', 'cầu thủ', 'trận đấu', 'quả bóng vàng',
          'champion league', 'fifa']

# Create an empty dataframe to store the matching posts and comments
topic_cmt = pd.DataFrame(columns=['clean_text', 'comments_full', 'comments_count', 'post_url'])

# Iterate through each post and its comments
for i in range(len(post_df_full_with_reactions)):
    # Check if the value in the 'clean_text' column is not NaN
    if not pd.isna(post_df_full_with_reactions['clean_text'][i]):
        # Check if the post contains any of the keywords
        if any(topic in post_df_full_with_reactions['clean_text'][i] for topic in topics):
            # Append the matching post and its comments to the dataframe
            topic_cmt = topic_cmt.append({'clean_text': post_df_full_with_reactions['clean_text'][i],
                                          'comments_full': post_df_full_with_reactions['comments_full'][i],
                                          'comments_count': post_df_full_with_reactions['comments'][i],
                                          'post_url': post_df_full_with_reactions['post_url'][i]},
                                         ignore_index=True)


# Print the number of matching posts
print(f"Có tất cả {len(topic_cmt)} bài viết chứa từ khóa", ', '.join(topics))
#topic_cmt.head()
```

- Sau khi chạy đoạn code trên thì em thu được 101 bài viết có chứa các từ của list topics mà em đã tạo sẵn liên quan đến bóng đá Messi và Ronaldo.

```
Có tất cả 101 bài viết chứa từ khóa man united, bóng đá,
```

- Tuy nhiên, khi kiểm tra lại thì phát hiện các bài viết càng về sau thì giá trị của trường dữ liệu comments_full là một list rỗng. Và chỉ có 18 bài viết có trường dữ liệu comments_full là cào được về có giá trị.

```
len(topic_cmt[topic_cmt['comments_count'] == 0])
```

```
83
```

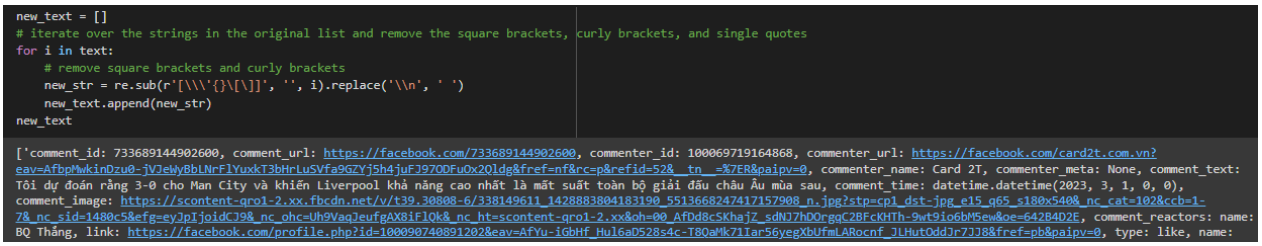
```
sad = topic_cmt['comments_count'] == 0
topic_cmt = topic_cmt.drop(topic_cmt[sad].index)
```

```
topic_cmt.info()
#cause i crawl lots of posts from facebook so i p
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 18 entries, 0 to 47
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   clean_text      18 non-null     object
 1   comments_full   18 non-null     object
 2   comments_count  18 non-null     object
 3   post_url        18 non-null     object
dtypes: object(4)
memory usage: 720.0+ bytes
```

- Chính vì vậy, em sẽ sử dụng các bình luận của 18 bài viết này và phân tích cảm xúc.

- Nhìn sơ qua từng giá trị của trường comments_full thì ta thấy nó có chứa khá nhiều thông tin, nhưng thông tin ta cần chính là các comment_text nên ta sẽ chỉ lấy các comment_text và bỏ hết tất cả những cái còn lại.



```
comment_texts = []
for item in new_text:
    matches = re.findall(r'comment_text:\s*(.*?)(?=, \w+:|$)', item)
    comment_texts.extend(matches)
comment_texts
```

```
['Tôi dự đoán rằng 3-0 cho Man City và khiến Liverpool khả năng cao nhất là mất suất toàn bộ giải đấu châu Âu mùa sau',
 ':v',
 '3-2 cho Liverpool',
 'Thấy bài viết của Theanh28 Entertainment là tương tác nà 😱',
 'Man xanh 4-3',
 '0-3 man xanh thắng',
 'Man xanh 5-1',
 'Man xanh nha 3-1',
 'Man xanh 1-0 liver',
```

```
cmt_noname = []
for sentence in comment_texts:
    words = sentence.split()
    for i, word in enumerate(words):
        if word.islower():
          upper_word = [w for w in words[0:i]]
          if len(upper_word) > 1:
            new_sentence = ' '.join(words[i:])
            break
          else:
            new_sentence = sentence
            break
    else:
        new_sentence = ''
    cmt_noname.append(new_sentence.strip())

cmt_noname
```

```
['Tôi dự đoán rằng 3-0 cho Man City và khiến Liverpool khả năng cao nhất là mất suất toàn bộ giải đấu châu Âu mùa sau',
 ':v',
 '3-2 cho Liverpool',
 'Thấy bài viết của Theanh28 Entertainment là tương tác nà 😱',
 'Man xanh 4-3',
 '0-3 man xanh thắng',
```

- Sau khi đã xử lí và lấy được các comment_text thì ta có được một list chứa các bình luận như ảnh bên trên.

- Tuy nhiên, sau khi xử lí lý dữ liệu bình luận cho các bài viết có chứa các từ liên quan đến chủ đề mà em quan tâm và chuyển nó thành một dataframe, thì em thấy rằng dữ liệu được cào không đầy đủ vì chỉ thu thập được 596 giá trị cho 18 bài viết.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 596 entries, 0 to 595
Data columns (total 1 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Comments  596 non-null    object
dtypes: object(1)
memory usage: 4.8+ KB
```

- Đây là một điều bất khả thi vì chỉ 1 bài viết của trang TheAnh28h Entertainment cũng đã có số lượt bình luận trung bình là từ 500 bình luận trở lên. Chính vì vậy, em cho rằng việc cào dữ liệu lớn với nhiều trường khác nhau sẽ khiến ta không thể cào đầy đủ các thông tin của một trường (nếu trường đó chứa quá nhiều thông tin: ví dụ như các bình

luận của một bài viết) nên em sẽ tiến hành chỉ cào lại dữ liệu của trường dữ liệu comments_full của từng bài viết có các từ liên quan đến chủ đề em quan tâm.

- Và để chỉ cào dữ liệu của những bài viết có chứa những từ liên quan đến những vấn đề mà em quan tâm thì em sẽ cần thêm các url của các bài viết đó, và thông qua url đó em sẽ chỉ cào các bình luận của bài viết nếu url của bài viết em cần cào giống với url của bài viết của page. Phần options sẽ chỉ để comments = True để nó chỉ cào comments còn các allow_extra_requests thì ta để là False để nó bỏ qua các trường dữ liệu khác và tập trung chỉ cào các bình luận.

```
def extract_comments(url: str, max_comments_per_post: int = 600, timeout: int = 10, wait: int = 1) -> List[str]:
    all_comments = []
    for post in get_posts(post_urls=[url], cookies=COOKIE_PATH, options={"comments": True, "allow_extra_requests": False}):
        if 'comments_full' in post:
            comments = [comment['comment_text'] for comment in post['comments_full'][:max_comments_per_post]]
            all_comments.extend(comments)
        else:
            print(f"No comments found for post: {post['post_url']}")
        time.sleep(wait)
        if len(all_comments) >= max_comments_per_post:
            break

    return all_comments
```

- Sau khi hoàn tất việc cào bình luận, thì ta sẽ có được 1 list các bình luận ở từng row tương ứng với bài viết mà ta muốn cào bình luận. Sau đó, ta sẽ thêm tất cả các bình luận của từng bài viết này vào 1 dataframe khác có thêm cmt_all để ta tiến hành đánh giá cảm xúc các bình luận trên.

```
all = []
for i in topic_cmt['all_comments']:
  for j in i:
    all.append(j)

cmt_all = pd.DataFrame(all, columns=['Comment'])
cmt_all.info()
```

- Và kết quả ta thu được là một dataframe chứa 7112 bình luận cho 18 bài viết chứa các từ về chủ đề em quan tâm. Tuy nhiên, ta vẫn thấy nếu tính ra trung bình thì mỗi bài viết sẽ chỉ có tầm 396 bình luận cho mỗi bài nhưng em sẽ tạm chấp nhận kết quả này vì việc cào dữ liệu facebook thông qua cookies đối với việc cào nhiều bài viết sẽ bị tình trạng không thể lấy hết được các thông tin cần cào và kết quả 7112 bình luận cũng tạm đủ cho em để tiến hành phân tích cảm xúc văn bản.

3.1.2.2.1. Tiền xử lí bài viết cho việc xây dựng mô hình gợi ý bài viết cho cá nhân và xây dựng mô hình tóm tắt.

- Việc tiền xử lí bài viết cho việc xây dựng mô hình gợi ý bài viết cho cá nhân và xây dựng mô hình tóm tắt các bài viết được gợi ý khá giống nhau.

```python
def preprocess_vietnamese_text_deeplearning(text):

    text = re.sub(r'(\w+)Xem thêm (\w+)', r'\1 \2', text)
    text = text.replace('/', ' ')
    # lowercase the text
    text = text.lower()

    #replace word with this format 'a.b' with 'ab'
    text = re.sub(r'(\w)\.(\w)', r'\1\2', text)

    # replace the abbreviations
    for abbr, full in abbreviations.items():
        text = re.sub(rf"\b{abbr}\b", full, text)

    text = re.sub(r'\b(cre|theo|nguồn):\s*\S+\s*', '', text)
    text = re.sub(r'\b(cre|theo|nguồn|video|hình ảnh) : \s*\S+.*', '', text)
    # remove accents
    #text = unidecode.unidecode(text)

    # remove emojis
    text = demoji.replace(text, '')
```

```python
# remove hastag
text = re.sub(r'#\w+\b', '', text)

#remove link
text = re.sub(r'\b(?:https?://|www\.|bit\.ly/)\S+\b', '', text)
text = re.sub(r'\b\S*\.com\S*\b', '', text)

# replace newlines with '. '
text = text.replace('\n', '.')
# replace multiple dots with a single dot
text = re.sub(r'\.{2,}', '. ', text)

# remove punctuation
#text = re.sub(r'(\w)[^\w\s]+', r'\1 ', text)
text = re.sub(r'[^\w\s]+', '', text)

#remove number
text = re.sub(r'\d+', '', text)

# tokenize the text
tokens = ViTokenizer.tokenize(text)

# join the tokens back into a single string
text = "".join(tokens)
text = text.replace('_', ' ')
return text
```

- Ta đều sẽ thực hiên các công việc sau:

- Loại bỏ từ ‘Xem thêm’.
- Thay đổi dấu ‘/’ thành dấu cách vì khi có các từ như 9.000/ngày thì sẽ thành ‘9.000 ngày’.
- Chuyển thành lowercase toàn bộ sau đó thay thế các từ viết tắt thành từ chuẩn form của nó.
- Loại bỏ hastag.
- Loại bỏ các câu thường xuất hiện cuối bài viết nhưng không liên quan đến việc phân tích cảm xúc như: theo, nguồn, video, ảnh,…
- Xử lí các dấu xuống dòng ‘\n’.
- Loại bỏ các punctuation và số.

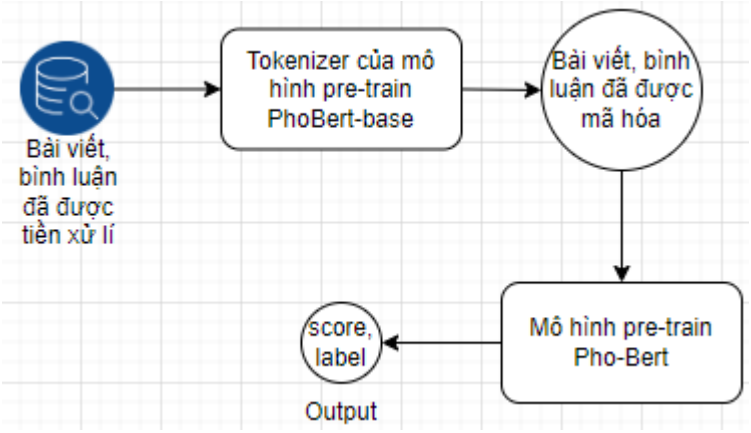
- Tuy nhiên, đối với việc xử lý bài viết cho tóm tắt văn bản thì em sẽ không bỏ các punctuation thì em cho rằng nó không làm cho mô hình bị sai lệch nên việc bỏ nó là không cần thiết.

3.2. ĐỀ XUẤT CÁC MÔ HÌNH TRIỂN KHAI.

- Đối với 2 mô hình PhoBert-base và Bert-base cho đa ngôn ngữ thì em sẽ chọn 2 mô hình đã được huấn luyện sẵn và em chỉ cần đổ dữ liệu 'sạch' vào 2 mô hình và cho ra kết quả, chứ không huấn luyện lại mô hình nữa vì dữ liệu ban đầu cào được là dữ liệu không có nhãn.

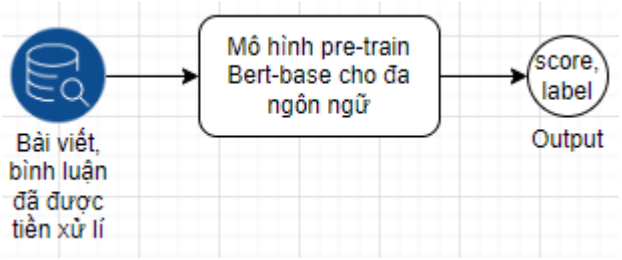
3.2.1. Đề xuất triển khai mô hình PhoBert-base.

- Dữ liệu dược đưa vào mô hình là dữ liệu sau khi đã được tiền xử lí (các bài viết và bình luận đã được tiền xử lí), sau đó sử dụng autotokenizer đã được huấn luyện trước của PhoBERT-base để tokenize các bài viết và các bình luận. Việc tokenizer sẽ chuyển đổi mỗi câu văn thành một chuỗi các mã token, và thêm các mã token đặc biệt để đánh dấu vị trí của từng câu. Sau đó, ta sử dụng mô hình được pre-trained trước là PhoBert-base để tiến hành tính toán score phân tích cảm xúc cho từng bài viết/bình luận sau đó ta sẽ dựa trên score đó để tiến hành chọn nhãn cho nó.



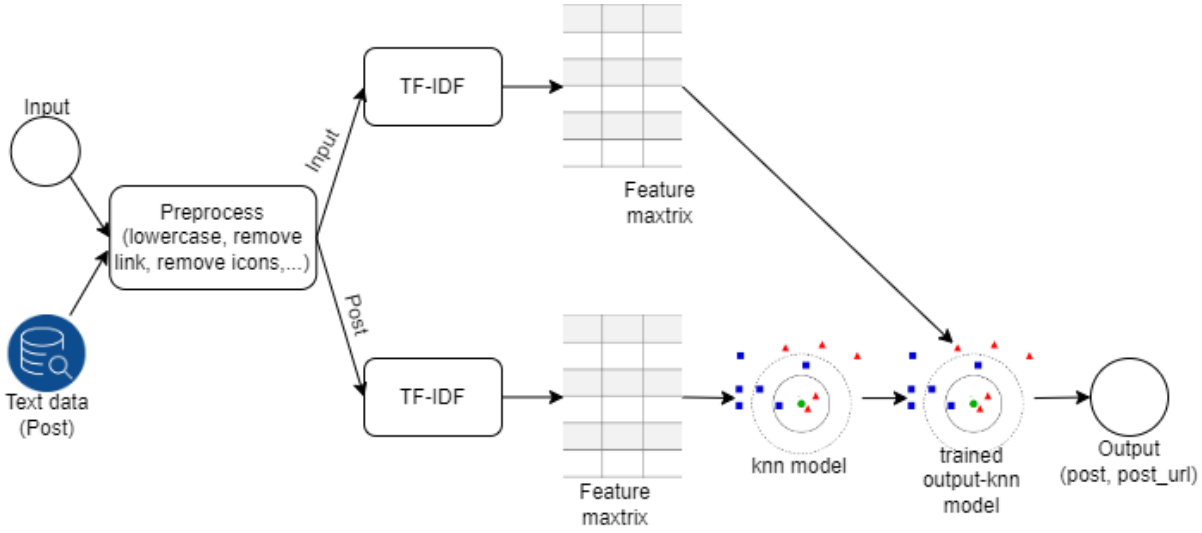
3.2.2. Đề xuất triển khai mô hình Bert-base cho đa ngôn ngữ.

- Đối với mô hình Bert-base cho đa ngôn ngữ thì dữ liệu dược đưa vào mô hình cũng là dữ liệu sau khi đã được tiền xử lí (các bài viết và bình luận đã được tiền xử lí) giống với dữ liệu đã được đưa vào mô hình PhoBert-base, sau đó ta chỉ cần đổ dữ liệu vào mô hình Bert-base cho đa ngôn ngữ để tiến hành tính toán score phân tích cảm xúc cho từng bài viết/bình luận sau đó ta sẽ dựa trên score đó để tiến hành chọn nhãn cho nó.



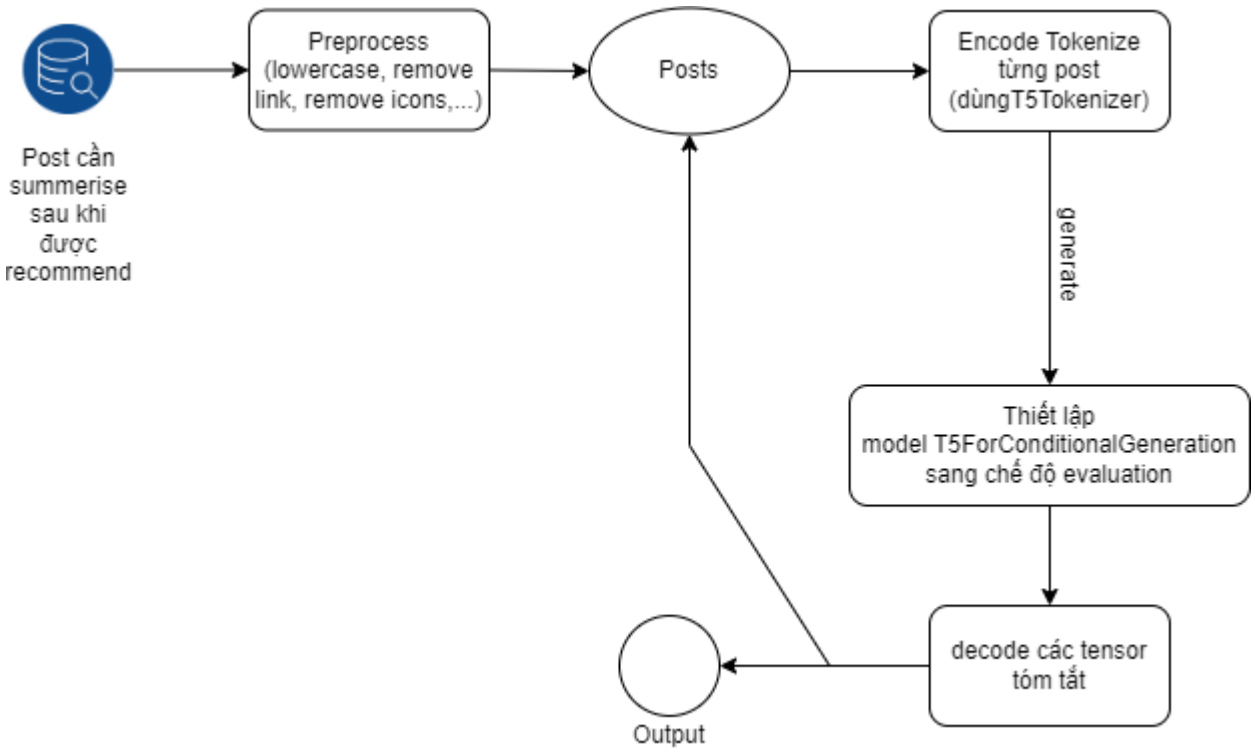
3.2.3. Đề xuất triển khai mô hình personal recommender system.

- Đối với mô hình hệ thống đề xuất cho bản thân thì em cũng sẽ cần phải tiền xử lí các bài viết và input (vấn đề quan tâm của bản thân là gì ?) sau đó ta sẽ tính trọng số từng từ bằng phương pháp TF-IDF và cho nó vào 1 matrix tạo thành 1 feature matrix cho 1 bài viết. Sau đó, ta sẽ dùng thuật toán knn để gom các bài viết có feature matrix tương tự nhau. Tiếp theo, ta sẽ so sánh feature matrix của input với feature matrix của từng cụm xem cụm nào có feature matrix giống với input thì output sẽ là các phần tử của cụm đó.



3.2.4. Đề xuất triển khai mô hình text summarization.

- Để có thể tóm tắt các bài viết sau khi được đề xuất thì em sẽ dùng mô hình pre-train của T5. Trước tiên ta vẫn sẽ cần phải tiền xử lí dữ liệu sao cho hợp lí, và sau đó ta mã hóa các bài viết và cho nó vào mô hình pre-train của T5 để nó chọn ra các mã hóa quan trọng trong từng bài viết. Cuối cùng, sau khi đã có được các đoạn mã hóa quan trọng thì ta sẽ decode nó lại để có được 1 văn bản tóm tắt cho các bài viết.


Post cần summerise sau khi được recommend
Preprocess (lowercase, remove link, remove icons,...)
Posts
Encode Tokenize từng post (dùngT5Tokenizer)
generate
Thiết lập model T5ForConditionalGeneration sang chế độ evaluation
decode các tensor tóm tắt
Output

CHƯƠNG 4. TRIỂN KHAI VÀ ĐÁNH GIÁ

4.1. TRIỂN KHAI CÁC MÔ HÌNH PHÂN TÍCH CẢM XÚC PHOBERT-BASE.

4.1.1. Triển khai mô hình phân tích cảm xúc PhoBert-base.

- Trước khi triển khai mô hình thì ta sẽ tải về một số thư viện cần thiết và khai báo nó.

```
!pip install torch
!pip install transformers

import torch
from transformers import AutoTokenizer, AutoModelForSequenceClassification
```

- Sau đó ta tạo một hàm để dự đoán phân tích cảm xúc. Ta sẽ sử dụng AutoTokenizer đã được huấn luyện trước của PhoBert-base để tokenize các bài viết/bình luận với các thông số như padding = True và truncation = True để đảm bảo mọi bài viết/bình luận đều được mã hóa thành độ dài giống nhau, sau đó đổ dữ liệu đã được mã hóa vào mô hình đã được huấn luyện trước của PhoBert-base để tiến hành tính điểm cho từng bài viết/bình luận.

- Sauk hi đã có điểm cho từng bài viết/bình luận, ta sẽ tiến hành gán nhãn cho nó ‘positive’ nếu nó lớn hơn 0.5 và ‘negative’ nếu nó bé hơn 0.5.

```
def predict_sentiment(texts, model_name = "vinai/phobert-
base", threshold=0.5):
    # Load PhoBERT tokenizer and sentiment analysis model
    tokenizer = AutoTokenizer.from_pretrained(model_name, use_fast=False)
    model = AutoModelForSequenceClassification.from_pretrained(model_name,
 num_labels=2)

    # Tokenize and encode the text data
    encoded_texts = tokenizer(texts, padding=True, truncation=True, return
_tensors="pt")

    # Use the pre-trained model to predict sentiment scores
    outputs = model(**encoded_texts)
    scores = outputs.logits.softmax(dim=1).detach().numpy()

    # Classify the sentiment based on the threshold
    sentiment_labels = np.where(scores[:, 1] > threshold, "positive", "neg
ative")

    # Return the sentiment labels
    return sentiment_labels
```

4.1.2. Triển khai mô hình phân tích cảm xúc Bert-base cho đa ngôn ngữ.

- Trước tiên, ta cần load mô hình BERT và tạo một pipeline để sử dụng cho dự đoán cảm xúc. Sau đó, ta tạo hai cột mới trong DataFrame để lưu trữ điểm số và nhãn dự đoán. Sử dụng vòng lặp để lặp qua từng bài viết/bình luận trong DataFrame và sử dụng pipeline để dự đoán cảm xúc của bình luận đó. Lưu trữ điểm số và nhãn dự đoán vào các cột tương ứng của DataFrame. Trả về DataFrame đã được dự đoán cảm xúc.

```
from transformers import pipeline
import pandas as pd

def predict_sentiment_bert(df, model_name):
    # Load the sentiment analysis pipeline
    sentiment_classifier = pipeline('sentiment-
analysis', model=model_name)

    # Create new columns to store the predicted score and label
    df['score'] = None
    df['label'] = None
    label_mapping = {
        '1 star': 'negative',
        '2 stars': 'negative',
        '3 stars': 'positive',
        '4 stars': 'positive',
        '5 stars': 'positive'
    }
    # Loop through each text in the dataframe and make a prediction
    for i, text in enumerate(df['clean_text']):
        # Predict the score and label using the pipeline
        result = sentiment_classifier(text)[0]
        score = result['score']
        label = result['label']
        mapped_label = label_mapping[label]
        # Add the score and label to the respective columns
        df.at[i, 'score'] = score
        df.at[i, 'label'] = mapped_label

    return df
predict_sentiment_bert(hgft, 'nlptown/bert-base-multilingual-uncased-
sentiment')
```

4.1.3. Triển khai mô hình hệ thống đề xuất bài viết cho cá nhân (Personal Recommender System).

- Bước 1: Sử dụng Tf-idf Vectorizer để rút trích đặc trưng cho từng bài viết trong tập dữ liệu. Đối với mỗi bài viết, ta sẽ tạo ra một vector đặc trưng thể hiện tần suất xuất hiện của từng từ trong văn bản đó.

- Bước 2: Sử dụng KNN để xây dựng mô hình tìm kiếm văn bản tương tự. Trong trường hợp này, mô hình sử dụng metric cosine để tính toán độ tương đồng giữa các bài viết. Điều này có nghĩa là mô hình sẽ tìm kiếm những bài viết có độ tương đồng cao nhất với nhau và gom nó thành các cụm theo khoảng cách cosine.

- Bước 3: Chuẩn bị câu truy vấn bằng cách tiền xử lý văn bản đầu vào bằng hàm preprocess_vietnamese_text_deeplearning.

- Bước 4: Rút trích đặc trưng cho input bằng TfidfVectorizer.

- Bước 5: Sử dụng KNN vừa được huấn luyện ở bước 2 để tìm kiếm các bài viết tương tự với input.

- Bước 6: Hiển thị kết quả tìm kiếm với các bài viết có độ tương đồng cao nhất với input "Post" chứa nội dung bài viết và cột "URL" chứa đường dẫn đến bài viết đó.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import NearestNeighbors


# Extract features using TF-IDF
tfidf = TfidfVectorizer()
features = tfidf.fit_transform(dl['clean_text'])

# Build k-NN model
knn = NearestNeighbors(n_neighbors=5, metric='cosine')
knn.fit(features)

# Define input sentences
input_sentences = ['tôi muốn xem những vụ việc, án mạng, tin nóng mà cộng
đồng mạng quan tâm nhiều.']

# Preprocess input sentences
preprocessed_input_sentences = [preprocess_vietnamese_text_deeplearning(se
ntence) for sentence in input_sentences]

# Extract features for input sentences
input_features = tfidf.transform(preprocessed_input_sentences)

# Find most similar posts to input sentences
```

```
distances, indices = knn.kneighbors(input_features)
similar_posts = dl.iloc[np.ravel(indices)]

# Display similar posts with URLs
similar_posts_with_urls = pd.DataFrame({'Post': similar_posts['post_text']
, 'URL': similar_posts['post_url']})
for i, sentence in enumerate(input_sentences):
    print("Similar posts to input sentence '{}':".format(sentence))
    print(similar_posts_with_urls.iloc[i])
similar_posts_with_urls.head()
```

4.1.4. Triển khai mô hình tóm tắt văn bản (Text Summerization).

- Mô hình T5 được sử dụng ở đây để tóm tắt văn bản, với việc cung cấp một văn bản đầu vào dưới dạng danh sách và một tokenizer được sử dụng để mã hóa các từ trong văn bản. Mã hóa này được thực hiện bằng cách sử dụng phương thức encode() của tokenizer và trả về một tensor kiểu PyTorch. Sau đó, tensor này được chuyển đến mô hình T5 để tạo ra tóm tắt.

- Quá trình tạo tóm tắt được thực hiện bằng cách sử dụng phương thức generate() của mô hình. Phương thức này tạo ra các câu tóm tắt từ các tokens đã mã hóa và trả về một tensor kiểu PyTorch. Các tham số được truyền vào phương thức generate() bao gồm max_length, num_beams, repetition_penalty, length_penalty và early_stopping. Các tham số này có nhiệm vụ điều chỉnh quá trình tạo tóm tắt. Ví dụ: max_length là để xác định độ dài tối đa của câu tóm tắt, num_beams để xác định số lượng "rẽ nhánh" để tạo ra các câu tóm tắt khác nhau, repetition_penalty dể xác định mức độ giảm điểm khi mô hình tạo ra các từ trùng lặp và length_penalty dùng để xác định độ ưu tiên của các câu tóm tắt có độ dài khác nhau.

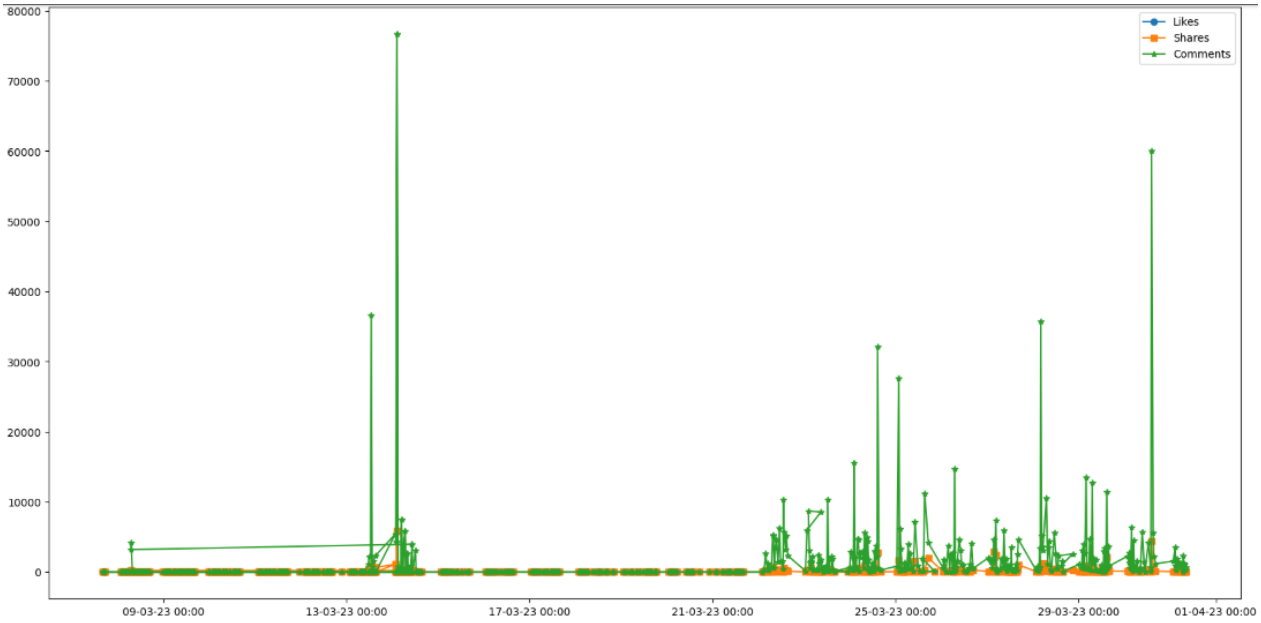
- Cuối cùng, kết quả được giải mã bằng phương thức decode() của tokenizer để tạo ra các câu tóm tắt cuối cùng. Các câu tóm tắt này được lưu trữ trong danh sách summary_post.

```
import torch

from transformers import T5ForConditionalGeneration, T5Tokenizer
import torch
if torch.cuda.is_available():
    device = torch.device("cuda")

    print('There are %d GPU(s) available.' % torch.cuda.device_count())

    print('We will use the GPU:', torch.cuda.get_device_name(0))
else:
    print('No GPU available, using the CPU instead.')
    device = torch.device("cpu")
```

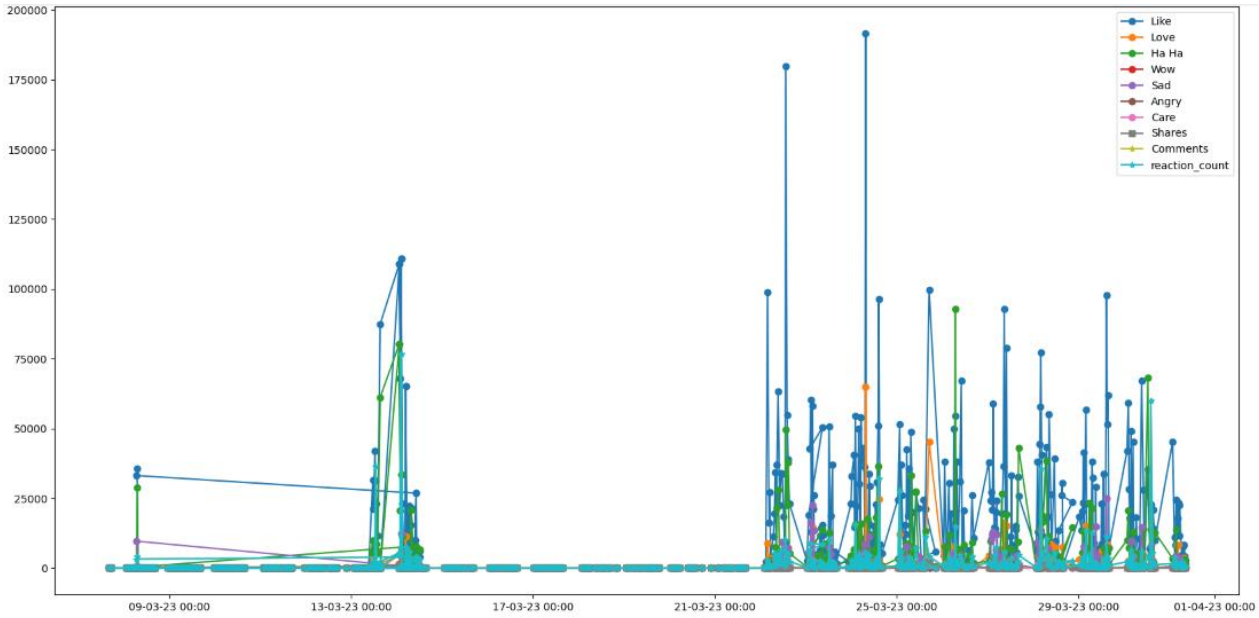
```
model = T5ForConditionalGeneration.from_pretrained("NlpHUST/t5-small-vi-
summarization")
tokenizer = T5Tokenizer.from_pretrained("NlpHUST/t5-small-vi-
summarization")
model.to(device)
src = text_sum['clean_text'].tolist()
summary_post = []
for i in src:
  tokenized_text = tokenizer.encode(i, return_tensors="pt").to(device)
  model.eval()
  summary_ids = model.generate(
                      tokenized_text,
                      max_length=256,
                      num_beams=5,
                      repetition_penalty=2.5,
                      length_penalty=1.0,
                      early_stopping=True
                  )
  output = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
  summary_post.append(output)
summary_post
```

4.2. ĐÁNH GIÁ CÁC MÔ HÌNH.

4.2.1. Đối với việc phân tích các tương tác.



- Nhìn vào biểu đồ, ta thấy đa số mọi người sẽ bình luận và tương tác (thích, yêu thích, haha,…) nhiều hơn rất nhiều so với share các bài viết.



- Like vẫn là tương tác chiếm số lần tương tác cao nhất trong tất cả các tương tác

-> Đối với việc này thì ta có thể cho rằng bởi vì có thể nút like là nút tiện lợi nhất khi ta tương tác, bởi nó luôn hiển thị trên bài viết còn những nút tương tác khác thì ta phải nhấn vào nút like và giữ im thì những nút tương tác khác mới hiện lên. Hoặc cũng có thể do thói quen của mỗi người, họ chỉ thích chọn nút like trong mọi trường hợp…

- Ngoài ra, ta còn có thể thấy được mối quan hệ của từng tương tác so với comments và nút share.

```
print(post_df_full_with_reactions[['thích','yêu thích','haha','wow','buồn','phẫn nộ','thương thương']].corrwith(post_df_full_with_reactions['comments']))
```

```
wow             0.650597
thích           0.521247
buồn            0.407410
haha            0.361121
thương thương   0.224019
yêu thích       0.189940
phẫn nộ         0.092238
```

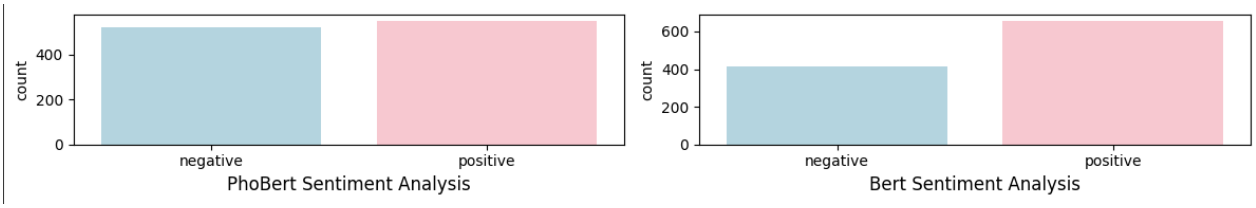
```
print(post_df_full_with_reactions[['thích','yêu thích','haha','wow','buồn','phẫn nộ','thương thương']].corrwith(post_df_full_with_reactions['shares']))
```

```
wow             0.585503
thích           0.548041
buồn            0.415320
haha            0.408576
thương thương   0.360896
yêu thích       0.303127
phẫn nộ         0.065904
```

-> wow và thích vẫn là 2 tương tác có mối quan hệ cao nhất với comments và nút share.

4.2.2. Đối với 2 mô hình phân tích cảm xúc (PhoBert-base và Bert-base cho đa ngôn ngữ).

- Đối với các bài viết sau khi phân tích cảm xúc ta có được biểu đồ thể hiện từng số phân tích cảm xúc của bài viết của 2 mô hình như sau:



-> Ta thấy được, có một sự chênh lệch của 2 nhãn của 2 mô hình với nhau. Nguyên nhân chủ yếu ở đây là vì bên mô hình Bert khi phân tích cảm xúc thì các giá trị từ 3 sao thì em để là positive thay vì là neutral nên các giá trị positive có vẻ vượt trội hơn so với negative của mô hình bert và positive của mô hình PhoBert.

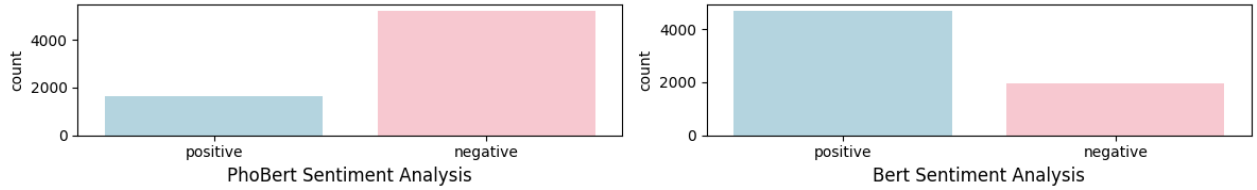
Sau đó, em tiến hành chọn ra 10 bài viết ngẫu nhiên và đưa ra nhãn của cá nhân bản thân để đánh giá 2 mô hình trên.

post_text	clean_text	hobert_lab	bert_label	NHÃN CỦA CÁ NHÂN EM ĐỀ RA
KIẾM GẦN	kiếm triệu	negative	negative	positive
ĐÔNG CO	đông lạc l	positive	negative	negative
LỚP HỌC M	lớp học m	positive	positive	positive
NGHỈ PHẠ	nghỉ phạn	positive	positive	negative
CHA RA TA	ra tay con	negative	negative	negative
Ngoại hạn	ngoại hạn	positive	negative	positive
Mang 5 sổ	sổ đỏ đi cả	negative	negative	negative
NỮ MC TẠ	nữ chươn	negative	positive	positive
HAI T.HI T.	hai thi thể	positive	negative	positive
NÉT ĐẸP N	nét đẹp pl	positive	positive	positive

-> Dựa vào nhãn cá nhân thì em thấy cả 2 mô hình đều hoạt động khá ổn và có số nhãn giống với nhãn em đề ra khá giống nhau.

PhoBert	6
Bert	6

- Đối với các bình luận sau khi phân tích cảm xúc ta có được biểu đồ thể hiện từng số phân tích cảm xúc của bài viết của 2 mô hình như sau:



-> Nhìn vào biểu đồ, ta thấy sự chệnh lệch cực kì lớn giữa các nhãn của 2 biểu đồ với nhau và giữa 2 mô hình với nhau. Vì vậy, em tiến hành kiểm tra lại dữ liệu thì phát hiện đa số các bình luận đều bình luận về tỷ số của trận đấu và tên đội thắng, nên khi phân tích cảm xúc thì của Phobert lại cho đó là negative còn bên Bert thì cho nó là positive:

3	LIVERPOOL 2-1	liverpool	negative	positive
4	Tôi dự đoán rằng 3-0 cho Man City và khiến Liverpool k	dự đoán man city liverpool khả năng suất to	negative	positive
5	Man xanh 2/0	man xanh	negative	positive
6	3-2 cho Liverpool	liverpool	negative	positive
7	Thấy bài viết của Theanh28 Entertainment là tương tá	viết theanh entertainment tương tác nà	negative	positive
8	7.0 cho mc	chương trình	positive	positive
9	Man xanh 2-1	man xanh	negative	positive
10	3-2 cho. Liverpool	liverpool	negative	positive
11	Man xanh 7-2😮	man xanh	negative	positive
12	Man xanh 4-3	man xanh	negative	positive
13	0-3 man xanh thắng	man xanh thắng	negative	positive
14	Man xanh 5-1	man xanh	negative	positive
15	Man xanh 4-0	man xanh	negative	positive
16	Man xanh 5-1	man xanh	negative	positive
17	Man xanh 3-1	man xanh	negative	positive
18	Man xanh 3-1	man xanh	negative	positive
19	Man xanh 3-1	man xanh	negative	positive
20	Man xanh 4-0	man xanh	negative	positive
21	Man xanh nha 3-1	man xanh nha	negative	positive
22	Man xanh 1-0 liver	man xanh liver	negative	positive
23	man 3-0 haaland hatrick	man haaland hatrick	negative	positive
24	Liverpool luôn là khắc tinh của Mc	liverpool khắc tinh chương trình	negative	positive

- Nên để có thể phân tích mô hình nào tốt hơn đối với bình luận thì em sẽ chọn ngẫu nhiên ra 10 comment như sau:

24	Liverpool luôn là khắc tinh của Mc	liverpool khắc tinh chương trình			negative	positive

		phobert	bert		NHÃN CỦA CÁ NHÂN EM ĐỀ RA	
Liverpool luôn là khắc tinh của Mc	liverpool l	negative	positive		negative	
Đam mê bất diệt	đam mê b	negative	positive		positive	
thói quen vấp cỏ bthường thôi mà:)	thói quen	negative	negative		negative	
ỦA tưởng chia tay đội tuyển r	tưởng chi	positive	positive		negative	
vô địch r	vô địch	positive	positive		positive	
Bữa trọng tài VN còn coi phát lại quả ghi bàn trên màn hìn	bữa trọng	negative	positive		negative	
Ở việt nam thì trọng tài cho bên nào win là win. Cùng lắm	việt nam t	positive	positive		negative	
Lại xem a nhà đá nữa rùi	nhà đá	negative	negative		negative	
Bồ ảnh cũng đẹp trai lắm 🤭	bồ ảnh đẹ	negative	positive		negative	
Loại này chỉ đá đít mấy em thôi .chứ đá đấm gì :)))	đá đít mấy	positive	positive		negative	

-> Sau khi kiểm tra thì có 6 nhãn của PhoBert giống với nhãn của em, và 4 nhãn của Bert giống với nhãn của em.

PhoBert	6
Bert	4

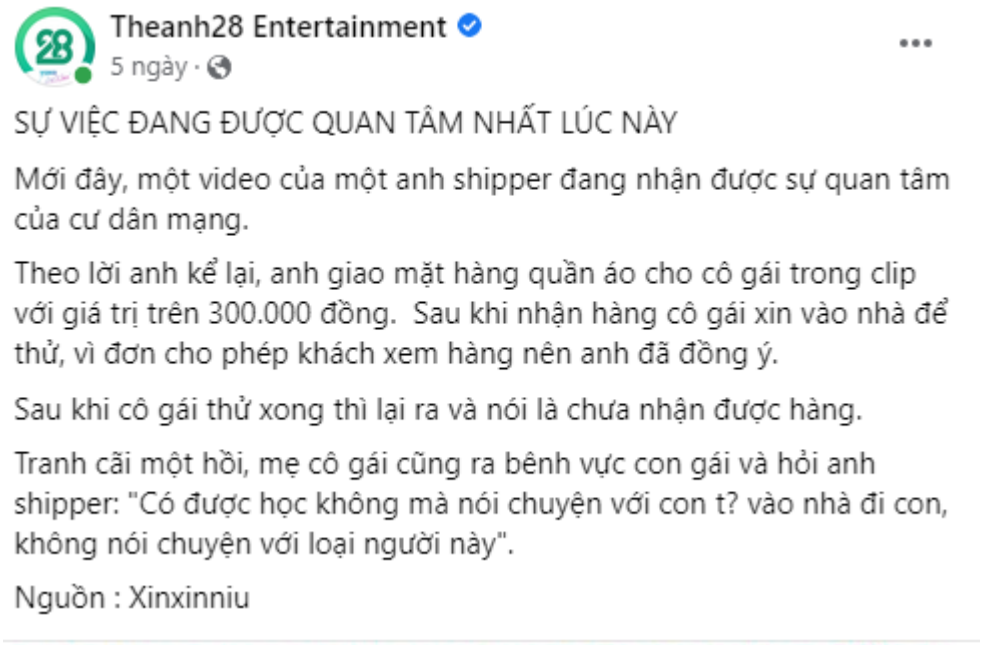
- Chính vì vậy, em có thể cho rằng mô hình PhoBert có thể hoạt động tốt hơn với Bert cho đa ngôn ngữ một chút khi dùng nó cho tiếng việt.

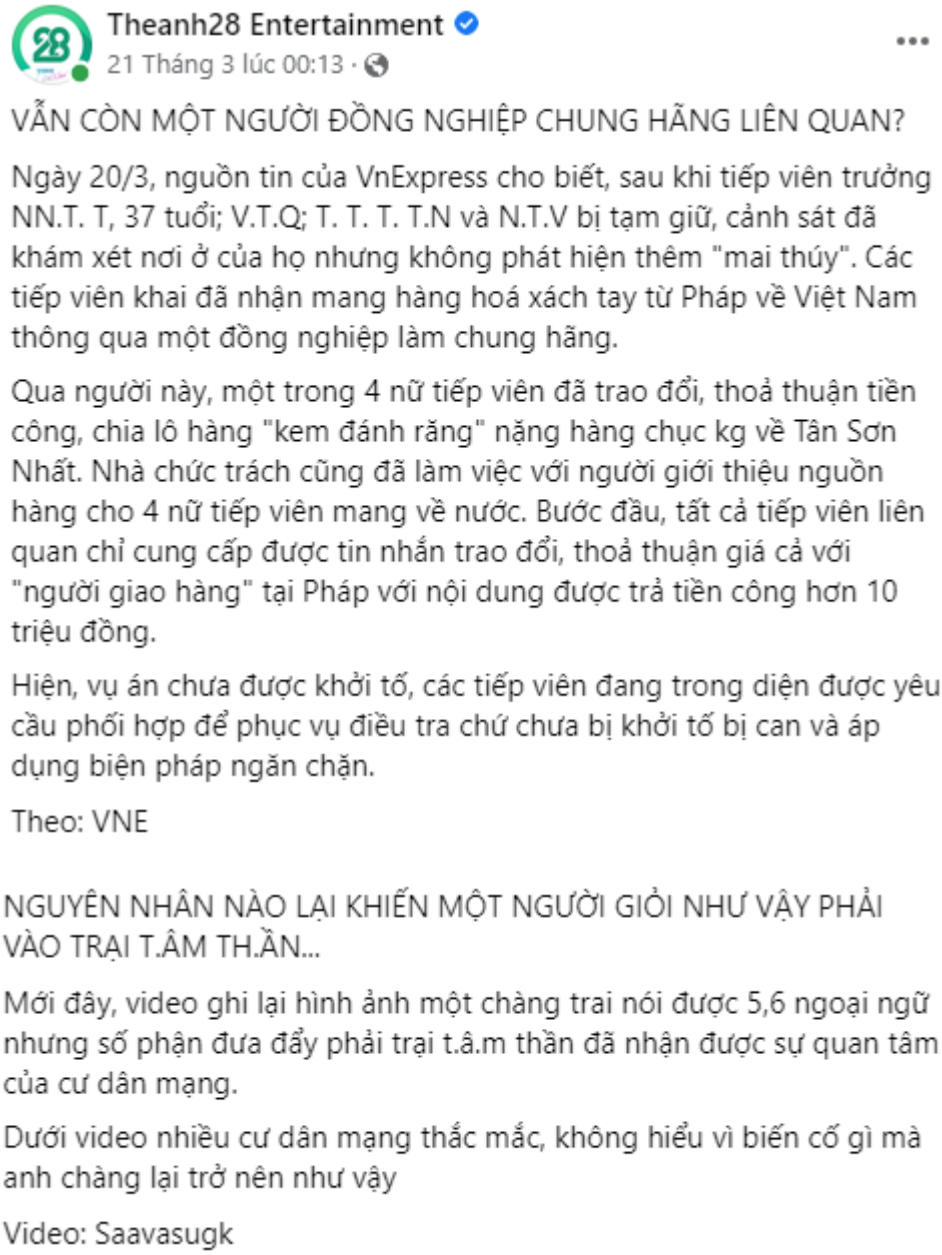
4.2.3. Đối với mô hình hệ thống đề xuất bài viết cho cá nhân (Personal Recommender System).

- Đối với mô hình hệ thống đề xuất bài viết cho cá nhân thì đã có sự thành công, các bài viết liên quan đến câu: ‘tôi muốn xem những vụ việc, án mạng, tin nóng mà cộng đồng quan tâm nhiều.’ đều có liên quan đến câu đó.

	Post	URL
34	SỰ VIỆC ĐANG ĐƯỢC QUAN TÂM NHẤT LÚC NÀY\n\nMới...	https://facebook.com/Theanh28/posts/7337134312...
249	NGUYÊN NHÂN NÀO LẠI KHIẾN MỘT NGƯỜI GIỎI NHƯ V...	https://facebook.com/Theanh28/posts/7312199008...
53	NGƯỜI NGHỆ SỸ CHÂN CHÍNH\n\n"Một đơn vị đã liê...	https://facebook.com/Theanh28/posts/7334738166...
361	KHÔNG PHẢI TÌNH CỜ MÀ PHÁT HIỆN, PHÁ ÁN\n\nTra...	https://facebook.com/Theanh28/posts/7295527993...
299	VẪN CÒN MỘT NGƯỜI ĐỒNG NGHIỆP CHUNG HẴNG LIÊN ...	https://facebook.com/Theanh28/posts/7304136256...

- Output được trả về là các bài viết và link url của các bài viết đó để ta có thể vào xem thử nó có đúng không. Dưới đây là một số ảnh của các bài viết được đề xuất:



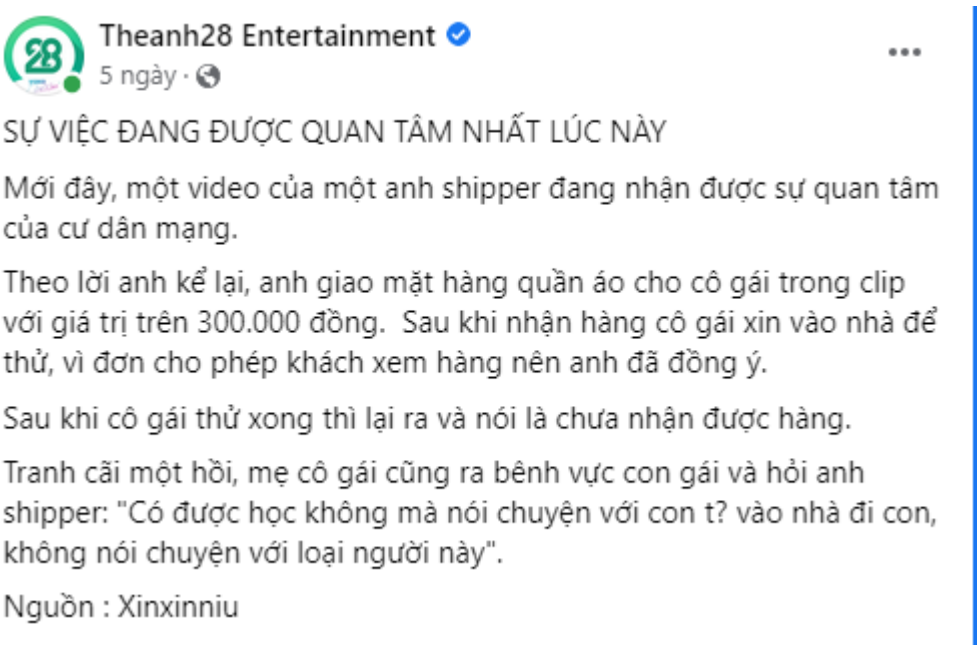


- Ta có thể thấy được, các bài viết được đề xuất đều hợp lí cho input: tôi muốn xem những vụ việc, án mạng, tin nóng mà cộng đồng quan tâm nhiều.' nên em cho rằng mô hình đề xuất bài viết cho cá nhân này là hợp lí và tốt.

4.2.4. Đối với mô hình tóm tắt văn bản (Text Summerization).

- Đối với mô hình tóm tắt bài viết sau khi đã có được các bài viết được đề xuất, đánh giá sơ qua mô hình hoạt động tốt và có hiệu quả, tuy nhiên vẫn có 1 ít ý chính của một số bài không được thêm vào trong bài tóm tắt nhưng không đáng kể nên em vẫn sẽ chấp nhận mô hình này và cho nó là hoạt động tốt.

- Như đối với bài viết này thì ta có được văn bản tóm tắt như bên dưới:



Theanh28 Entertainment
5 ngày ·

SỰ VIỆC ĐANG ĐƯỢC QUAN TÂM NHẤT LÚC NÀY

Mới đây, một video của một anh shipper đang nhận được sự quan tâm của cư dân mạng.

Theo lời anh kể lại, anh giao mặt hàng quần áo cho cô gái trong clip với giá trị trên 300.000 đồng. Sau khi nhận hàng cô gái xin vào nhà để thử, vì đơn cho phép khách xem hàng nên anh đã đồng ý.

Sau khi cô gái thử xong thì lại ra và nói là chưa nhận được hàng.

Tranh cãi một hồi, mẹ cô gái cũng ra bênh vực con gái và hỏi anh shipper: "Có được học không mà nói chuyện với con t? vào nhà đi con, không nói chuyện với loại người này".

Nguồn : Xinxinniu

```
['Một video của một anh shipper đang nhận được sự quan tâm của cư dân mạng. Sau khi nhận hàng cô gái xin vào nhà để thử xong thì lại ra và nói là chưa nhận được hàng, vì đơn cho phép khách xem hàng nên anh đã đồng ý.',
```

- Sau khi tóm tắt, thì ta thấy bài viết chủ yếu nói về vụ việc anh shipper giao hàng cho khách và khách xin vào nhà để thử hàng nhưng khi ra lại bảo không nhận được hàng, vì đơn hàng cho phép khách xem nên shipper đã đồng ý cho thử.

➔ Văn bản tóm tắt khá ổn so với toàn bộ nội dung bài viết, tuy nhiên vẫn còn thiếu ý: mẹ của khách hàng còn ra bênh vực người con (theo em đây là một ý chính để bổ sung thêm cho vụ lùm xùm trên).

CHƯƠNG 5. KẾT LUẬN

5.1. CÁC KẾT QUẢ ĐẠT ĐƯỢC.

- Đã hoàn thành được việc cào dữ liệu từ Facebook, và phân tích cảm xúc cho từng bài viết/bình luận.

- Đã so sánh được kết quả phân tích cảm xúc của 2 mô hình PhoBert-Base và Bert-base cho đa ngôn ngữ.

- Các bài viết sau khi được mô hình đề xuất bài viết cho cá nhân hiển thị đều có liên quan đến yêu cầu ban đầu và hợp lí đối với cá nhân bản thân em.

- Việc tóm tắt các bài viết cá nhân khi có được các bài viết được đề xuất khá ổn khi thể hiện được ý chính của bài viết.

5.2. NHỮNG HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN.

- Những hạn chế:

- Đối với việc cào dữ liệu từ Facebook còn bị hạn chế bởi khi chính sách bảo mật an toàn thông tin của Facebook khi sử dụng phương pháp cào dữ liệu Facebook bằng cookies với mong muốn cào được nguồn dữ liệu lớn.
- Đối việc tiền xử lí thì có quá nhiều từ viết tắt khiến cho mô hình phân tích cảm xúc không hiểu nên em chỉ có thể viết 1 số từ viết tắt sang từ hoàn chỉnh của nó.
- Đối với các bình luận thì đa số toàn là những bình luận về kết quả và tên đội thắng.
- Việc chỉnh các thông số của mô hình còn chưa hợp lí nên có bị lệch về 1 bên nhãn 1 chút.
- Mô hình hệ thống đề xuất bài viết cho cá nhân hoạt động khá hiệu quả nhưng vẫn cần thêm hiệu chỉnh để có thể đưa ra các bài viết chính xác hơn với yêu cầu ban đầu.
- Mô hình tóm tắt văn bản thì có 1 ít ý chính không được đưa thêm vào văn bản được tóm tắt.

- Hướng phát triển:

- Ta có thể thử việc cào dữ liệu Facebook bằng API để kiểm tra việc cào dữ liệu lớn tren Facebook bằng API của chính nó có bị hạn chế số lượng dữ liệu được cào hay không.
- Ngoải ra, ta có thể tìm thêm những cách khác để có thể cào dữ liệu từ Facebook.
- Ta nên huấn luyện lại 2 mô hình phân tích cảm xúc theo bộ dữ liệu của ta cào được từ Facebook để nó có thể đạt hiệu quả tốt hơn trong việc phân tích cảm xúc.

- Tùy chỉnh các tham số của 2 mô hình phân tích cảm xúc để cải thiện hiệu năng của nó. Cũng như có thể set thêm cho nó phân tích thành 3 nhãn để tránh bị lệch về 1 bên quá nhiều.
- Đối với các từ viết tắt thì ta có thể tìm kiếm các giải pháp khác như: tìm kiếm một file có sẵn các từ viết tắt và thể hoàn chỉnh của nó, hoặc đối với những từ viết tắt thì ta phải loại bỏ nó đi, hoặc xử lí nó sao cho hợp lí.
- Với hệ thống đề xuất bài viết thì ta có thể sử dụng các phương pháp tính khoàng cách khác ngoài cosine như euclidean, khoảng cách mahalanobis,… để kiểm tra và chọn ra khi áp dụng phương pháp tính khoảng cách nào thì sẽ cho ra các bài viết đáp ứng được yêu cầu đầu vào hợp lí và chính xác nhất có thể.
- Đối với việc tóm tắt văn bản còn bị mất đi 1 ít ý chính thì ta có thể kéo dài khoảng cách của văn bản tóm tắt ra thêm 1 tí để nó có thể thêm ý chính đó vào. Hoặc ta có thể sử dụng dữ liệu đầu vào đa dạng hơn, và điều chỉnh các tham số của mô hình để nó hoạt động hiệu quả và tốt nhất.

TÀI LIỆU THAM KHẢO

- Code cào dữ liệu từ facebook.
- Chat GPT hỗ trợ việc code các mô hình và lý thuyết.