Phân tích cảm xúc và nhận diện chủ đề cho văn bản

*Tháng 6/2024

1st Nguyen Le Trong Nhan
MSSV: 20521698
University of Information Technology
HCM City, VietNam
20521698@gm.uit.edu.vn

2nd Nguyễn Phước An Vũ
MSSV: 20522165
University of Information Technology
HCM City, VietNam
20522165@gm.uit.edu.vn

3rd Đặng Quang Trung
MSSV: 20522067
University of Information Technology
HCM City, VietNam
20522067@gm.uit.edu.vn

4th Phạm Quang Khải
MSSV: 20520566
University of Information Technology
HCM City, VietNam
20520566@gm.uit.edu.vn

***Tóm tắt nội dung*—Ngày nay, với sự phát triển nhanh chóng của công nghệ thông tin, con người càng dễ dàng hơn trong việc tiếp cận được nhiều nguồn thông tin từ nhiều luồng khác nhau có thể qua: mạng xã hội, phương tiện thông tin đại chúng, truyền miệng ... Để góp phần tối ưu hóa quá trình tiếp cận thông tin của người dùng được chính xác hơn với các nhu cầu về thông tin, phương pháp phân loại nội dung văn bản (hay còn gọi là phân loại văn bản) là một lĩnh vực quan trọng trong xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP). Đây là quá trình tự động gán nhãn hoặc phân loại các văn bản thành các nhóm hoặc danh mục xác định trước dựa trên nội dung của chúng. Trong bài viết này, chúng tôi đã vận dụng phương pháp trên vào việc phân tích cảm xúc và nhận diện chủ đề cho văn bản. Với đầu vào của bài toán là một đoạn văn bản chứa các thông tin có thể liên quan đến một chủ đề cụ thể nào đó với cảm xúc nhất định; sau khi đi qua các bước xử lý về mặt dữ liệu và qua các mô hình được áp dụng để phân tích sẽ cho ra kết quả là đoạn văn bản đó với các nhãn chủ đề mà đoạn văn đó thể hiện cũng như nhãn cảm xúc của đoạn văn đó. Cụ thể, các mô hình đó là gì, phương pháp được sử dụng để phân tích cảm xúc và nhận diện chủ đề là gì đã được trình bày trong bài viết của chúng tôi.**

Từ khóa: ***Index Terms*—Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), Logistic Regresstion (LR), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Natural Language Processing (NLP), Sentiment Analysis, Topic Modeling**

I. Giới thiệu

Phương pháp phân loại nội dung văn bản () là một lĩnh vực quan trọng trong xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP). Đây là quá trình tự động gán nhãn hoặc phân loại các văn bản thành các nhóm hoặc danh mục xác định trước dựa trên nội dung của chúng. Trong bài viết này chúng tôi dùng hai kỹ thuật quan trọng trong xử lý ngôn ngữ tự nhiên là phân tích cảm xúc và nhận diện chủ đề (Sentiment Analysis) và nhận diện chủ đề (Topic Modeling). Phân tích cảm xúc là quá trình xác định và phân loại các ý kiến thể hiện trong văn bản thành các loại cảm xúc khác nhau như tích cực, tiêu cực, trung tính, hoặc các trạng thái cảm xúc cụ thể như vui vẻ, buồn bã, giận dữ, lo lắng, v.v. Nhận diện chủ đề là quá trình khám phá các chủ đề ẩn (ẩn ngữ) trong một tập hợp lớn các văn bản. Mỗi chủ đề là một tập hợp các từ thường xuất hiện cùng nhau và có xu hướng đại diện cho một khái niệm hoặc ý tưởng nhất định. Hai kỹ thuật này thường được sử dụng một cách riêng biệt đáp ứng từng bài toán cụ thể về đánh giá sản phẩm, giám sát khách hàng đối với phân tích cảm xúc và phân loại tài liệu, tìm kiếm thông tin đối với nhận diện chủ đề. Trong bài viết này, chúng tôi đã thực hiện kết hợp hai kỹ thuật trên để giải quyết bài toán tối ưu hóa việc tiếp cận thông tin của người dùng, làm thông tin tiếp cận chính xác hơn với nhu cầu của từng người dùng.

Identify applicable funding agency here. If none, delete this.

II. Các công cụ liên quan

A. Công cụ Pyspark

Một giao diện lập trình ứng dụng (API) Python cho Apache Spark, một công cụ mạnh mẽ giúp thực hiện các tác vụ xử lý big data và thời gian thực trong một môi trường phân tán, sử dụng ngôn ngữ lập trình Python.

B. Công cụ Google Colab

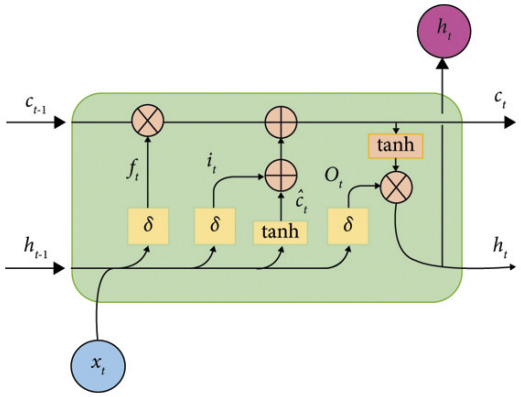
Một sản phẩm từ Google Research, nó cho phép chạy các dòng code python thông qua trình duyệt, đặc biệt phù hợp với Data analysis, machine learning và giáo dục. Colab không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, bạn có thể sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được cung cấp cho bạn.

III. Các model

A. Model Long Short-Term Memory

Long Short-Term Memory (LSTM) là một loại mạng nơ-ron hồi quy (Recurrent Neural Network - RNN) được thiết kế để xử lý và dự đoán dữ liệu tuần tự, chẳng hạn như chuỗi thời

gian và dữ liệu ngôn ngữ tự nhiên. LSTM đã được giới thiệu bởi Hochreiter và Schmidhuber vào năm 1997, và nó đã trở nên phổ biến trong nhiều ứng dụng như nhận diện giọng nói, xử lý ngôn ngữ tự nhiên, và dự báo chuỗi thời gian.



Hình 1. Long Short-Term Memory Model.

$$\text{Forget gate: } f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{1}$$

- f_t: vector đầu ra của Forget Gate tại thời điểm t.
- W_f: ma trận trọng số cho Forget Gate.
- $[h_{t-1}, x_t]$: vector kết hợp giữa trạng thái ẩn tại thời điểm $t-1$ và đầu vào hiện tại x_t.
- b_f: vector bias cho Forget Gate.
- σ: hàm kích hoạt sigmoid, giá trị từ 0 đến 1.

Tạo ra một vector để quyết định những giá trị nào sẽ được cập nhật:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{1}$$

- i_t: vector đầu ra của Input Gate tại thời điểm t.
- W_i: ma trận trọng số cho Input Gate.
- $[h_{t-1}, x_t]$: vector kết hợp giữa trạng thái ẩn tại thời điểm $t-1$ và đầu vào hiện tại x_t.
- b_i: vector bias cho Input Gate.

Tạo ra một vector các giá trị mới có thể được thêm vào trạng thái ô nhớ:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2}$$

- $\tilde{C}_t$: vector giá trị ô nhớ mới tại thời điểm t.
- W_C: ma trận trọng số cho ô nhớ.
- b_C: vector bias cho ô nhớ.
- $\tanh$: hàm kích hoạt tanh, giá trị từ -1 đến 1.

Trạng thái ô nhớ mới C_t được cập nhật bằng cách kết hợp trạng thái cũ đã được quên một phần và thông tin mới từ Input Gate:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3}$$

- C_t: trạng thái ô nhớ tại thời điểm t.
- f_t: đầu ra của Forget Gate.
- C_{t-1}: trạng thái ô nhớ tại thời điểm $t-1$.
- i_t: đầu ra của Input Gate.
- $\tilde{C}_t$: giá trị ô nhớ mới.

Output Gate quyết định phần nào của trạng thái ô nhớ sẽ được xuất ra ngoài và trở thành trạng thái ẩn tiếp theo h_t:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{4}$$

- o_t: vector đầu ra của Output Gate tại thời điểm t.
- W_o: ma trận trọng số cho Output Gate.
- $[h_{t-1}, x_t]$: vector kết hợp giữa trạng thái ẩn tại thời điểm $t-1$ và đầu vào hiện tại x_t.
- b_o: vector bias cho Output Gate.
- σ: hàm kích hoạt sigmoid, giá trị từ 0 đến 1.

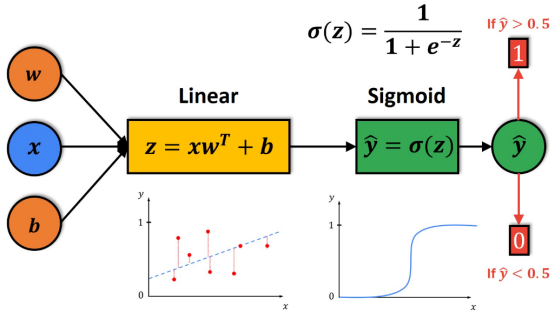
Cuối cùng, trạng thái ẩn mới h_t được xác định bằng cách kết hợp Output Gate và trạng thái ô nhớ mới đã qua hàm tanh:

$$h_t = o_t * \tanh(C_t) \tag{5}$$

- h_t: trạng thái ẩn tại thời điểm t.
- o_t: đầu ra của Output Gate.
- C_t: trạng thái ô nhớ mới.
- $\tanh$: hàm kích hoạt tanh, giá trị từ -1 đến 1.

B. Model Logistic Regression

Hồi quy logistic là một phương pháp thống kê được sử dụng rộng rãi trong các ngành khoa học xã hội, y học, và kinh doanh để dự đoán xác suất xảy ra của một sự kiện nhị phân. Được phát triển từ mô hình hồi quy tuyến tính, hồi quy logistic cung cấp một cách tiếp cận linh hoạt và hiệu quả để giải quyết các vấn đề phân loại.



Hình 2. Logistic Regression Model.

Hồi quy logistic dựa trên ý tưởng rằng xác suất xảy ra của một sự kiện có thể được mô tả bởi một hàm logistic (hay còn gọi là hàm sigmoid). Hàm này được định nghĩa như sau:

$$P(Y = 1|X) = \frac{1}{1 + e(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)} \tag{2}$$

Trong đó:

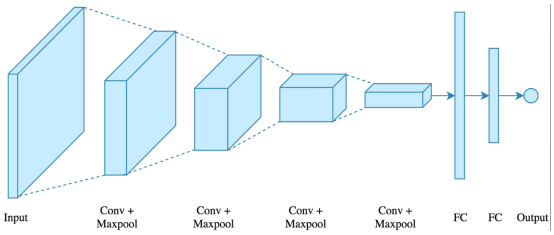
- $P(Y = 1|X)$ là xác suất xảy ra sự kiện Y (ví dụ như bệnh có/không, mua hàng/không mua hàng).

- $\beta_0, \beta_1, ...\beta_n$ là các tham số của mô hình.
- $X_1, X_2, ...X_n$ là các biến độc lập.

Mô hình logistic biến đổi đầu ra của hồi quy tuyến tính thành một xác suất trong khoảng từ 0 đến 1, thông qua hàm logistic, dễ hiểu và dễ triển khai. Tuy nhiên mô hình này có thể không phản ánh chính xác quan hệ thực tế cũng như sẽ gặp khó khăn khi dữ liệu mất cân bằng.

C. Model CNN (Convolutional Neuro Network)

Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) là một loại mô hình học sâu đặc biệt mạnh mẽ trong các tác vụ xử lý hình ảnh và thị giác máy tính. Được phát triển dựa trên cấu trúc và chức năng của hệ thống thị giác sinh học, CNN đã đạt được nhiều thành tựu đáng kể trong các lĩnh vực như nhận dạng hình ảnh, phân loại ảnh, phát hiện đối tượng và nhiều ứng dụng khác.



Hình 3. Convolutional Neuro Network Model.

CNN bao gồm nhiều lớp khác nhau, mỗi lớp đảm nhiệm một vai trò cụ thể trong quá trình phân tích và nhận dạng hình ảnh. Các lớp chính trong CNN bao gồm:

*1) **Lớp Tích Chập (Convolutional Layer)**:* Lớp tích chập là lớp quan trọng nhất trong CNN. Nó sử dụng các bộ lọc (filter) để trích xuất các đặc trưng từ hình ảnh đầu vào. Mỗi bộ lọc được convolve qua toàn bộ hình ảnh, tạo ra một bản đồ đặc trưng (feature map).

$$Z_{ij} = (X * W)_{ij} + b \tag{3}$$

Trong đó:

- Z_{ij} là bản đồ đặc trưng tại vị trí (i, j).
- X là đầu vào (thường là một ma trận biểu diễn hình ảnh) .
- W là bộ lọc.
- b là hệ số điều chỉnh (bias).
- * là phép tích chập (convolution).

Tích Chập (Convolution): Bộ lọc W được trượt (slide) qua đầu vào X, và tại mỗi vị trí, tính tổng tích của các phần tử tương ứng từ W và X. Quá trình này có thể được viết chi tiết hơn như sau:

$$Z_{ij} = \sum_{M=0}^{M-1} \sum_{N=0}^{N-1} X_{i+m,j+n} W_{mn} + b \tag{4}$$

- **M** và **N** là kích thước của bộ lọc WWW.
- **i** và **j** là chỉ số hàng và cột của vị trí hiện tại của bộ lọc trên đầu vào X.
- $X_{i+m,j+n}$ là giá trị của đầu vào tại vị trí (i+m, j+n).
- W_{mn} là giá trị của bộ lọc tại vị trí (m, n).
- b hoặc **Bias** là một hằng số được thêm vào kết quả của phép tích chập để điều chỉnh giá trị đầu ra.

2) Lớp Kích Hoạt (Activation Layer): Lớp kích hoạt thường sử dụng hàm ReLU (Rectified Linear Unit) để giới hạn các giá trị đầu ra và tạo ra tính phi tuyến cho mô hình.

$$f(x) = max(0, x) \tag{5}$$

Hàm này đơn giản thay thế tất cả các giá trị âm bằng 0, trong khi giữ nguyên các giá trị dương. Điều này giúp tạo ra tính phi tuyến cho mô hình, làm cho mô hình có khả năng học các đặc trưng phức tạp hơn.

*3) **Lớp Gộp (Pooling Layer)**:* Lớp gộp được sử dụng để giảm kích thước không gian của bản đồ đặc trưng, giúp giảm số lượng tham số và tính toán trong mô hình. Phương pháp gộp phổ biến là Max Pooling, trong đó giá trị lớn nhất trong mỗi vùng nhỏ của bản đồ đặc trưng được chọn làm đầu ra.

$$Z_{pool} = Max(Z_{region}) \tag{6}$$

Chia bản đồ đặc trưng thành các vùng nhỏ không chồng chéo nhau (thường là 2x2), và chọn giá trị lớn nhất trong mỗi vùng làm giá trị đầu ra cho vùng đó. Ví dụ, với một vùng 2x2 như sau: **[1 3] [2 4]** Giá trị đầu ra của vùng này sau Max Pooling sẽ là 4 (giá trị lớn nhất trong vùng). **Average Pooling**: Một phương pháp gộp khác là Average Pooling, trong đó giá trị trung bình của mỗi vùng được chọn làm đầu ra.

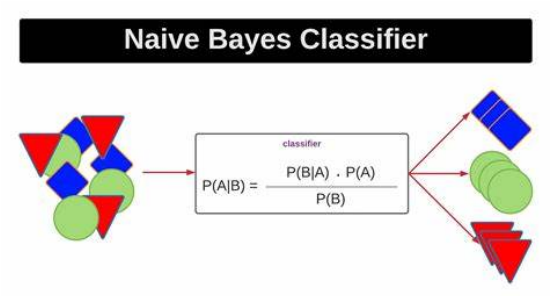
*4) **Lớp Kết Nối Đầy Đủ (Fully Connected Layer)**:* Lớp này kết nối tất cả các nơ-ron từ lớp trước với mỗi nơ-ron của lớp hiện tại, tạo ra một mạng nơ-ron truyền thống. Đây là lớp cuối cùng trong CNN, thường được sử dụng để thực hiện các tác vụ phân loại.

$$y = WZ + b \tag{7}$$

- **Kết Nối Đầy Đủ**: Tất cả các nơ-ron từ lớp trước được kết nối với mỗi nơ-ron của lớp hiện tại. Điều này tương tự như một mạng nơ-ron truyền thống (fully connected neural network).
- **Ma Trận Trọng Số (Weight Matrix)**: Mỗi kết nối giữa hai nơ-ron có một trọng số tương ứng, và tất cả các trọng số này được biểu diễn dưới dạng ma trận W.
- **Hệ Số Điều Chỉnh (Bias)**: b là một vector bias được thêm vào kết quả của phép nhân ma trận để điều chỉnh giá trị đầu ra.

D. Model Multinomial Naive Bayes

Multinomial Naive Bayes (MNB) là một thuật toán máy học xác suất được sử dụng chủ yếu cho phân loại văn bản. Nó thuộc họ các bộ phân loại Naive Bayes, dựa trên Định lý Bayes với giả định "ngây thơ"về tính độc lập có điều kiện giữa mỗi cặp đặc trưng cho trước nhãn lớp. Biến thể Naive Bayes Đa Thức đặc biệt phù hợp cho phân loại với các đặc trưng rời rạc, như số lần xuất hiện từ trong các vấn đề phân loại văn bản.



Hình 4. Multinomial Naive Bayes Model.

1) Định lý Bayes: là nền tảng của các bộ phân loại Naive Bayes. Nó mô tả xác suất của một sự kiện dựa trên kiến thức trước đó về các điều kiện có thể liên quan đến sự kiện đó. Định lý được phát biểu bằng toán học như sau:

$$P(C|X) = \frac{P(X|C) * P(C)}{P(X)} \tag{8}$$

Trong đó:

- $P(C|X)$ là xác xuất hậu nghiệm của lớp C cho trước dự đoán X.
- $P(X|C)$ là khả năng của dự đoán X cho trước lớp C.
- $P(C)$ là xác xuất tiên nghiệm của lớp C.
- $P(X)$ là xác xuất tiên nghiệm của dự đoán X.

2) Phân phối đa thức: Trong bối cảnh của Multinomial Naive Bayes, thuật ngữ 'multinomial' đề cập đến phân phối đa thức, được sử dụng để mô hình hóa phân phối số lượng từ trong một tài liệu. Mỗi tài liệu được biểu diễn như một vector của tần suất từ.

3) Giả định: Các đặc trưng (ví dụ: tần suất từ) là độc lập có điều kiện cho trước lớp và khả năng của các đặc trưng tuân theo phân phối đa thức.

4) Huấn luyện: Trong quá trình huấn luyện, thuật toán tính toán xác suất của mỗi lớp và xác suất có điều kiện của mỗi đặc trưng cho trước lớp.

- **Xác xuất tiên nghiệm** $P(C_i)$: Xác xuất của mỗi lớp C_i trong dữ liệu huấn luyện.
- **Khả năng** $P(x_j|C_i)$: Xác xuất của đặc trưng x_j cho trước lớp C_i, thường được ước lượng bằng tần suất của x_j trong các tài liệu của lớp C_i.

5) Làm mịn Laplace: Để xử lý các xác suất bằng không, làm mịn Laplace được áp dụng. Điều này bao gồm việc thêm 1 vào số lần xuất hiện của mỗi từ trong từ vựng để đảm bảo không có xác suất nào bằng không.

$$P(x_j|C_i) = \frac{N_{ij} + \alpha}{N_i + \alpha.n} \tag{9}$$

Trong đó:

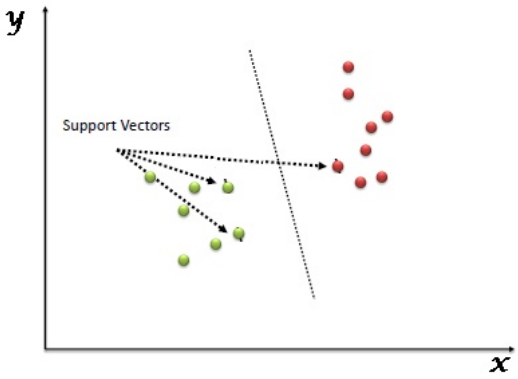
- N_{ij} là số lần xuất hiện của đặc trưng x_j trong lớp C_i.
- N_i là tổng số lần xuất hiện của tất cả các đặc trưng trong lớp C_i.
- α là tham số làm mịn (thường được đặt là 1).
- n là số lượng đặc trưng duy nhất.

6) Dự đoán: Để phân loại một trường hợp mới, thuật toán tính toán xác suất hậu nghiệm cho mỗi lớp bằng công thức xuất phát từ định lý Bayes. Lớp có xác suất hậu nghiệm cao nhất được chọn là lớp dự đoán.

$$\widehat{y} = argmax P(C_i) \prod_{j=1}^{n} P(x_j|C_i)^{x_j} \tag{10}$$

E. Model Support Vector Machine

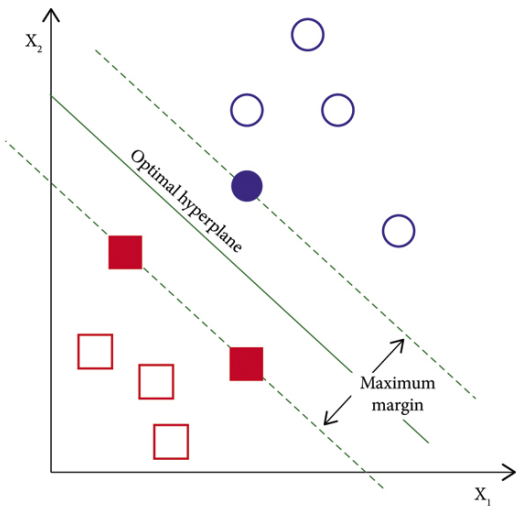
Support Vector Machine (SVM) là là một thuật toán học máy giám sát, có thể được sử dụng cho cả phân loại và hồi quy. Tuy nhiên, nó chủ yếu được sử dụng cho các bài toán phân loại. SVM hoạt động bằng cách tìm kiếm một siêu phẳng (hyperplane) tốt nhất để phân tách các mẫu dữ liệu thuộc các lớp khác nhau trong không gian đặc trưng nhiều chiều. Hyperplane được hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.



Hình 5. Support Vector Machine Model.

Support Vectors là các đối tượng trên đồ thị tọa độ quan sát, Support Vector Machine là một biên giới để chia hai lớp tốt nhất.

1) Biên giữa (Margin): là khoảng cách giữa siêu phẳng và các điểm dữ liệu gần nhất từ mỗi lớp. SVM tìm kiếm siêu phẳng tối đa hóa biên giữa này. Các điểm dữ liệu nằm trên biên hoặc gần biên được gọi là các vector hỗ trợ (support vectors).



Hình 6. Biểu đồ Margin trong SVM.

2) Kernel Trick: Để xử lý các dữ liệu không thể phân tách tuyến tính trong không gian ban đầu, SVM sử dụng kernel trick để ánh xạ dữ liệu vào một không gian đặc trưng cao hơn mà tại đó dữ liệu có thể phân tách tuyến tính. Một số kernel phổ biến bao gồm:

- **Polynomial kernel**: $K(x, x') = (x.x' + 1)^d$
- **Gaussian (RBF) kernel**: $K(x, x') = exp(-\frac{||x-x'||^2}{2\sigma^2})$
- **Sigmoid kernel**: $K(x, x') = tanh(\alpha x.x' + c)$

IV. Kết quả thực nghiệm

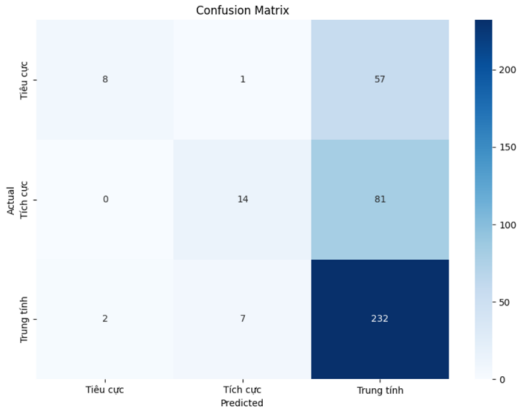
Đầu tiên, chúng tôi thực hiện phân tích bộ dữ liệu dưới khía cạnh áp dụng kỹ thuật phân tích cảm xúc với cột có tên là "Feeling", được kết quả như sau:

Bảng I
Bảng kết quả hiệu suất của các mô hình sử dụng kỹ thuật phân tích cảm xúc

Models	Test Accuracy (%)	Precision	Recall	F1-score
Multinomial Naive Bayes (MNB)	60.95	0.74	0.35	0.29
Support Vector Machine (SVM)	63.18	0.69	0.41	0.40
Long Short-Term Memory (LSTM)	59.95	0.20	0.33	0.25
Logistic Regression (LR)	61.69	0.59	0.38	0.35
Convolutional Neural Network (CNN)	62.00	0.56	0.43	0.42

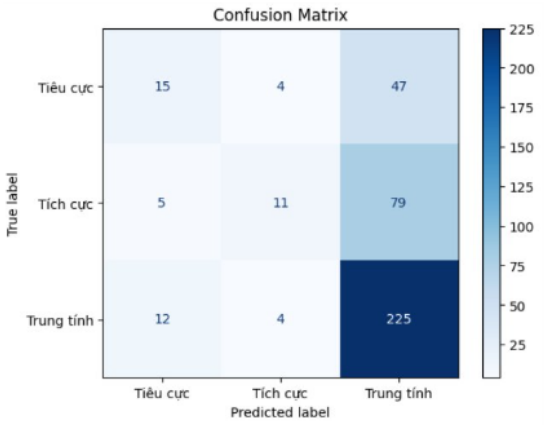
Từ kết quả trên có thể thấy rằng, mô hình SVM và CNN có hiệu suất tổng thể tốt nhất cho thấy độ tương thích với bộ dữ liệu, khả năng phân loại chính xác các mẫu dữ liệu dương tính của hai mô hình này.

Ma trận nhầm lẫn của mô hình SVM:



Hình 7. Ma trận nhầm lẫn của SVM.

Ma trận nhầm lẫn của mô hình CNN:



Hình 8. Ma trận nhầm lẫn của CNN.

Các mô hình còn lại có kết quả hiệu suất chưa cao là do một số nguyên nhân về đặc trưng của bộ dữ liệu không tương thích cao với mô hình, cấu hình và tham số của từng mô hình, kỹ thuật tiền xử lý và trích xuất đặc trưng.
Tiếp theo, chúng tôi thực hiện phân tích kết hợp hai kỹ thuật phân tích cảm xúc và nhận diện chủ đề cho bộ dữ liệu, kết quả được đạt như sau:

Bảng II
BẢNG KẾT QUẢ HIỆU SUẤT CỦA CÁC MÔ HÌNH SỬ DỤNG HAI KỸ THUẬT KẾT HỢP

Models	Test Accuracy (%)	Precision	Recall	F1-score
Multinomial Naive Bayes (MNB)	20.38	0.19	0.18	0.14
Support Vector Machine (SVM)	14.93	0.06	0.06	0.05
Long Short-Term Memory (LSTM)	18.35	0.03	0.06	0.03
Logistic Regression (LR)	21.06	0.27	0.15	0.16
Convolutional Neural Network (CNN)	13.2	0.18	0.15	0.16

Từ kết quả trên có thể thấy mô hình MNB cho kết quả hiệu suất cao nhất nhưng kết quả vẫn thấp hơn rất nhiều so với kỳ vọng của chúng tôi, có thể thấy kết quả Precision, Recall, F1-score rất kém. Nguyên nhân lớn nhất dẫn đến điều này là do bộ dữ liệu kém chất lượng (dữ liệu này lấy không rõ nguồn gốc), độ chuẩn của bộ dữ liệu không rõ dẫn đến ảnh hưởng sâu sắc đến kết quả đánh giá của bài toán.

V. KẾT LUẬN

Việc sử dụng kết hợp hai phương pháp phân tích cảm xúc và nhận diện chủ đề đối với đoạn văn bản đã cho ra một số kết quả. Nhìn chung, việc phân tích cũng đã cho ra kết quả như mục tiêu ban đầu đề ra đó là với đoạn văn bản đã cho, sau khi trải qua quá trình phân tích, đánh giá đã cho ra chủ đề cũng như cảm xúc của đoạn văn bản đó. Về mặt thông số kỹ thuật, kết quả đạt được của việc phân tích kết hợp khá là thấp so với việc sử dụng một phương pháp, điều này không được như kỳ vọng ban đầu. Một số nguyên nhân có thể gây ra kết quả thấp như sau: bộ dữ liệu được sử dụng để train cho các mô hình chưa được đạt chuẩn, mô hình hóa dữ liệu không được tốt, giới hạn về số lượng các nhãn, sự chênh lệch số lượng giữa các đoạn văn có chủ đề khác nhau gây ảnh hưởng đến hiệu suất của các mô hình. Với những nguyên nhân đó, để góp phần cải thiện kết quả phân tích chúng tôi đưa ra một số giải pháp như sau: sử dụng bộ dữ liệu đa dạng hơn, được chuẩn hóa tốt hơn; với các mô hình được sử dụng phân tích cần phải tối ưu hơn, sử dụng các mô hình có độ phức tạp cao.

TÀI LIỆU

[1] **"Pattern Recognition and Machine Learning"** by Christopher M. Bishop.
[2] **"Introduction to Machine Learning"** by Ethem Alpaydin.
[3] **"The Elements of Statistical Learning"** by Trevor Hastie, Robert Tibshirani, and Jerome Friedman.
[4] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory."in Neural Computation.
[5] Christopher Olah's blog on understanding LSTM networks.
[6] Deep Learning Book by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
[7] Andrew Ng's Deep Learning Specialization on Coursera.
[8] **"Regularized Text Logistic Regression: Key Word Detection and Sentiment Classification for Online Reviews"** by Ying Chen, Peng Liu, and Chung Piaw Teo.
[9] **"Categorizing Text Documents Using Naïve Bayes, SVM and Logistic Regression"** by Shubham Kumar et al.
[10] **"Understanding Convolutional Neural Networks for Text Classification"** Alon Jacovi and Oren Sar Shalom Computer Science Department, Bar Ilan Univesity, Israel, IBM Research, Haifa, Israel,Intuit, Hod HaSharon, Israel, Allen Institute for Artificial Intelligence
[11] **"Convolutional Neural Networks for Sentence Classification"** Yoon Kim, New York University