**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A logo for a university

Description automatically generated**

**BÁO CÁO BÀI TẬP LỚN**

**MÔN HỌC: LẬP TRÌNH CHO TRÍ TUỆ NHÂN TẠO**

**Nhóm 8:  
Đề tài: Phân tích cảm xúc truyền thông xã hội.**

**Thành viên nhóm**

**Nguyễn Ngọc Hiếu 21011956 K15-KHMT**

**Nguyễn Thế Tường 21011079 K15-KHMT**

**GVHD: Th.S Nguyễn Văn Thiệu**

**28/06/2024 – Hà Nội**

Mục lục

[**I.Tóm tắt**](#_heading=h.1fob9te) **1**

[**II. Giới thiệu**](#_heading=h.3znysh7) **2**

[2.1 Tổng quan môn học](#_heading=h.2et92p0) 2

[2.2 Tổng quan về đề tài phân tích cảm xúc truyền thông xã hội](#_heading=h.tyjcwt) 2

[**III. Kiến thức nền tảng**](#_heading=h.1t3h5sf) **3**

3.1. Mô hình Học máy 6

3.2. Tiền xử lý dữ liệu 7

3.3. Đánh giá và tối ưu mô hình 7

[**IV. Khai phá dữ liệu**](#_heading=h.4d34og8) **7**

[**V. Thiết kế triển khai mô hình**](#_heading=h.2s8eyo1) **11**

[5.1 Các bước xây dụng mô hình](#_heading=h.17dp8vu) 11

[5.2 Huấn luyện mô hình](#_heading=h.3rdcrjn) 13

[5.3 Đánh giá mô hình](#_heading=h.26in1rg) 14

[**VI. Tài liệu tham khảo**](#_heading=h.lnxbz9) **16**

# I.Tóm tắt

Phân tích tình cảm xã hội (social media sentiment analysis) là quá trình xác định và phân loại cảm xúc từ các bài viết và bình luận trên mạng xã hội để hiểu rõ ý kiến của người dùng về một chủ đề, sự kiện, sản phẩm, hoặc thương hiệu. Mục tiêu chính của bài toán này là phân loại cảm xúc thành các nhóm tích cực, tiêu cực, hoặc trung tính, từ đó cung cấp cái nhìn sâu sắc giúp doanh nghiệp điều chỉnh chiến lược marketing, cải thiện dịch vụ khách hàng, và định hướng phát triển sản phẩm. Kết quả đạt được từ quá trình này giúp doanh nghiệp nắm bắt được xu hướng cảm xúc của người dùng, đưa ra quyết định thông minh và kịp thời, nâng cao mối quan hệ với khách hàng và tối ưu hóa hiệu quả kinh doanh.

# II. Giới thiệu

## 2.1 Tổng quan môn học

Lập trình cho trí tuệ nhân tạo (AI) là một lĩnh vực trọng yếu trong khoa học máy tính, tập trung vào việc phát triển các hệ thống và thuật toán có khả năng học hỏi, suy luận, và ra quyết định giống như con người. Môn học này trang bị cho sinh viên các kỹ năng cần thiết để thiết kế, triển khai và tối ưu hóa các mô hình AI.

Các khái niệm về học máy (Machine learning), học sâu (Deep learning), xử lí ngôn ngữ tự nhiên (NLP) là nền tảng quan trọng trong việc phân tích dữ liệu.

Môn học lập trình cho trí tuệ nhân tạo không chỉ cung cấp nền tảng lý thuyết vững chắc mà còn mang lại kỹ năng thực hành quan trọng, giúp người học sẵn sàng tham gia vào các dự án phân tích dữ liệu và ứng dụng AI trong thực tế. Bằng cách hiểu và áp dụng các kỹ thuật này, sinh viên có thể đóng góp vào sự phát triển và đổi mới trong nhiều lĩnh vực, từ kinh doanh, y tế, đến giáo dục và khoa học.

## 2.2 Tổng quan về đề tài phân tích cảm xúc truyền thông xã hội

Bài tập lớn về phân tích cảm xúc truyền thông xã hội nhằm giải quyết vấn đề hiểu và đo lường cảm xúc của người dùng mạng xã hội về các chủ đề, sự kiện, sản phẩm hoặc thương hiệu cụ thể. Trên các nền tảng mạng xã hội như Twitter, Facebook, và Instagram, người dùng thường xuyên chia sẻ quan điểm và cảm xúc của mình. Tuy nhiên, khối lượng thông tin này rất lớn và đa dạng, gây khó khăn cho việc phân tích thủ công. Do đó, cần thiết phải phát triển các hệ thống tự động để phân tích và rút ra những thông tin có giá trị từ dữ liệu này.

**Cụ Thể, Bài Tập Lớn Nghiên Cứu Các Khía Cạnh Sau:**

1. Thu Thập Dữ Liệu Từ Mạng Xã Hội:

* Thu thập dữ liệu từ các bài viết, bình luận, và các phản hồi trên các nền tảng mạng xã hội.
* Sử dụng API hoặc các công cụ web scraping để tự động thu thập dữ liệu liên quan đến chủ đề nghiên cứu.

1. Tiền Xử Lý Dữ Liệu:

* Làm sạch dữ liệu để loại bỏ các thông tin không liên quan, xử lý các từ không có ý nghĩa, và chuẩn hóa văn bản.
* Xử lý các ký tự đặc biệt, biểu tượng cảm xúc và các yếu tố khác ảnh hưởng đến chất lượng dữ liệu.

1. **Phân Tích Cảm Xúc:**

* Áp dụng các thuật toán học máy và sử dụng các từ điển cảm xúc để phân loại cảm xúc trong dữ liệu.
* Sử dụng các thư viện trong python và các mạng nơ-ron để xác định cảm xúc là tích cực, tiêu cực, hoặc trung tính.

1. **Phân Tích Kết Quả:**

* Tổng hợp và phân tích dữ liệu cảm xúc để xác định xu hướng chung của cảm xúc người dùng về chủ đề nghiên cứu.
* Sử dụng các công cụ trực quan hóa dữ liệu như biểu đồ và đồ thị để thể hiện kết quả phân tích.

# III. Kiến thức nền tảng

Phân tích dữ liệu trong lập trình AI là một phần quan trọng của quá trình xây dựng và huấn luyện mô hình máy học và học sâu. Dưới đây là những kiến thức cơ bản, phương pháp và công cụ quan trọng cần phải biết:

### Các khái niệm cơ bản

1. Dữ liệu (Data)**:** Thông tin được sử dụng để huấn luyện mô hình, bao gồm các đặc trưng (features) và nhãn (labels) trong bài toán giám sát.
2. Phân tích dữ liệu (Data Analysis)**:** Quá trình khám phá và hiểu các dữ liệu để chuẩn bị cho việc huấn luyện mô hình.
3. Mô hình học máy (Machine Learning Model): Các thuật toán và công cụ được sử dụng để học từ dữ liệu và thực hiện các dự đoán hoặc phân loại.
4. Mô hình học sâu (Deep Learning Model): Là một phương pháp học máy sử dụng các mạng nơ-ron nhân tạo sâu (deep neural networks) để học và hiểu các biểu diễn dữ liệu phức tạp.

### Các phương pháp quan trọng:

1. Tiền xử lý dữ liệu (Data Preprocessing): Bao gồm chuẩn hóa dữ liệu, xử lý dữ liệu thiếu, mã hóa các biến phân loại (categorical variables), và phân tích đặc trưng (feature analysis).
2. Trực quan hóa dữ liệu (Data Visualization): Sử dụng biểu đồ và đồ thị để hiểu các mối quan hệ và phân bố của dữ liệu.
3. Phân tích thống kê (Statistical Analysis): Đánh giá sự phụ thuộc và tương tác giữa các biến để chọn mô hình phù hợp.
4. Phân tích đặc trưng (Feature Analysis): Đánh giá mức độ quan trọng của các đặc trưng đối với mô hình và tối ưu hóa chọn lọc đặc trưng.

### Các Công Cụ Quan Trọng

1. Scikit-learn: Thư viện Python cung cấp các thuật toán machine learning và công cụ hữu ích cho tiền xử lý dữ liệu và đánh giá mô hình.
2. TensorFlow và PyTorch: Các framework cho deep learning, cung cấp cơ sở hạ tầng để xây dựng, huấn luyện và triển khai mô hình học sâu.
3. Keras: Thư viện cao cấp cho deep learning, giúp đơn giản hóa việc xây dựng và huấn luyện mô hình.
4. Pandas và NumPy: Thư viện cung cấp cấu trúc dữ liệu và công cụ phân tích dữ liệu hiệu quả, làm nền tảng cho việc tiền xử lý dữ liệu.
5. Matplotlib và Seaborn: Thư viện để trực quan hóa dữ liệu, giúp hiểu rõ hơn về các mối quan hệ và phân bố của dữ liệu.
6. NLTK và spacy: Các thư viện xử lý ngôn ngữ tự nhiên, hỗ trợ trong các bài toán liên quan đến văn bản và dữ liệu ngôn ngữ.

Trong Python, có nhiều thư viện và gói công cụ phổ biến được sử dụng để phân tích dữ liệu một cách hiệu quả. Dưới đây là các sơ qua về một số thư viện và gói công cụ quan trọng trong lĩnh vực này:

### Thư viện quan trọng trong phân tích dữ liệu

1. Pandas**:**

**+** Chức năng: Cung cấp cấu trúc dữ liệu và công cụ để xử lý và phân tích dữ liệu.

+ Đặc điểm: Sử dụng DataFrame để tổ chức và xử lý dữ liệu dạng bảng, hỗ trợ các thao tác như chuẩn hóa, ghép nối, lọc dữ liệu, và tính toán thống kê cơ bản.

1. NumPy:

+ Chức năng: Thực hiện tính toán khoa học và toán học trên mảng dữ liệu đa chiều.

+ Đặc điểm: Cung cấp các cấu trúc dữ liệu mảng (arrays) và hàm số để thao tác, tính toán với dữ liệu số học một cách hiệu quả.

1. MatplotlibvàSeaborn:

+ Chức năng: Trực quan hóa dữ liệu bằng biểu đồ và đồ thị.

+ Đặc điểm: Matplotlib là thư viện cơ bản và mạnh mẽ cho việc tạo ra các loại biểu đồ đơn giản và phức tạp. Seaborn là một gói mở rộng trên Matplotlib, giúp tạo ra các biểu đồ thống kê phức tạp và đẹp mắt hơn.

1. Scikit-learn:

+ Chức năng: Cung cấp các thuật toán học máy (machine learning) phổ biến cho phân tích dữ liệu và xây dựng mô hình.

+ Đặc điểm: Bao gồm các công cụ cho tiền xử lý dữ liệu, phân tích và lựa chọn đặc trưng, huấn luyện mô hình, và đánh giá hiệu suất mô hình.

1. NLTK (Natural Language Toolkit**)** và spaCy:

+ Chức năng: Xử lý và phân tích dữ liệu văn bản và ngôn ngữ tự nhiên.

+ Đặc điểm: NLTK cung cấp các công cụ cho xử lý ngôn ngữ tự nhiên như phân tích cú pháp, phân tích tình cảm, và nhiều công cụ khác. spaCy là một thư viện nhanh và hiệu quả cho xử lý ngôn ngữ tự nhiên và phân tích ngữ nghĩa.

1. TensorFlowvàPyTorch:

+ Chức năng: Cung cấp framework cho deep learning và xây dựng mô hình học sâu.

+ Đặc điểm: TensorFlow là một trong những framework phổ biến nhất cho deep learning với khả năng triển khai mô hình trên nhiều nền tảng khác nhau. PyTorch cũng rất phổ biến và được yêu thích với cú pháp đơn giản và sự linh hoạt trong việc xây dựng mô hình.

Nói chung Trí tuệ nhân tạo (Artificial Intelligence - AI) là lĩnh vực nghiên cứu và phát triển các hệ thống máy tính có khả năng thực hiện các công việc mà trước đây đòi hỏi sự thông minh của con người.

**\*Liên hệ kiến thức nền tảng với bài toán phân tích cảm xúc truyền thông**

Ở trong bài toán phân tích cảm xúc truyền thông chúng tôi đã sử dụng các công cụ, thuật toán và mô hình AI sau:

### 1. Mô hình Học máy

* Mạng nơ-ron hồi quy LSTM (Long Short-Term Memory):

+ Mô tả: LSTM là một dạng đặc biệt của mạng nơ-ron hồi quy (RNN) được thiết kế để xử lý dữ liệu có mối quan hệ thời gian hoặc dạng chuỗi, như văn bản.

+ Ứng dụng trong bài toán: LSTM được sử dụng để mô hình hóa dữ liệu văn bản từ các bài đăng trên mạng xã hội, giúp phân loại cảm xúc của các bài đăng này thành các lớp nhất định (positive, negative, neutral).

* Biến thể LSTM: Bidirectional LSTM:

+ Mô tả: Là một biến thể của LSTM cho phép mô hình học từ cả hai hướng của dữ liệu, từ đầu đến cuối và từ cuối về đầu, để cải thiện khả năng biểu diễn và hiệu quả của mô hình.

+ Ứng dụng trong bài toán: Bidirectional LSTM được áp dụng để cải thiện khả năng phân loại cảm xúc bằng cách học hỏi từ cả ngữ cảnh hiện tại và quá khứ của dữ liệu văn bản.

### 2. Tiền xử lý dữ liệu

* Natural Language Toolkit (NLTK)**:**

+ Mô tả: Thư viện Python phổ biến dùng để xử lý và phân tích ngôn ngữ tự nhiên.

+ Công dụng: NLTK được sử dụng để tiền xử lý văn bản bằng cách loại bỏ stopword, chuẩn hóa từ và chuẩn hóa dữ liệu văn bản trước khi đưa vào mô hình LSTM.

### 3. Đánh giá và tối ưu mô hình

* Categorical Crossentropy Loss:

+ Mô tả: Là hàm mất mát thích hợp cho các bài toán phân loại nhiều lớp, đo lường sự sai khác giữa phân phối xác suất dự đoán của mô hình và thực tế.

+ Sử dụng trong bài toán: Được sử dụng để đánh giá hiệu suất của mô hình phân loại cảm xúc trên dữ liệu huấn luyện và kiểm tra.

* Adam Optimizer:

+ Mô tả**:** Là một thuật toán tối ưu hóa gradient dựa trên hợp của hai thuật toán khác là AdaGrad và RMSProp.

+ Sử dụng trong bài toán: Adam được áp dụng để tối ưu hóa mô hình LSTM trong quá trình huấn luyện, giúp cải thiện khả năng hội tụ và tốc độ huấn luyện của mô hình.

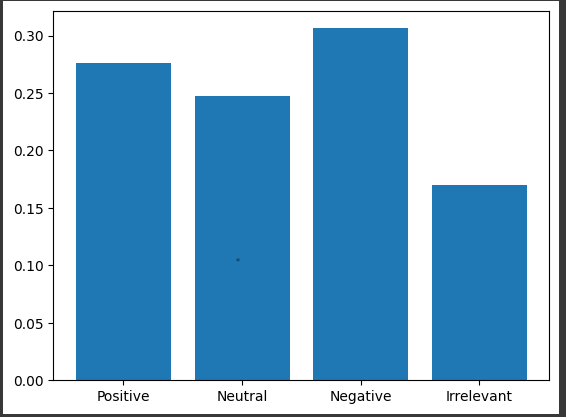
# IV. Khai phá dữ liệu

Điểm quan trọng nhất của dự án của chúng tôi là khai thác dữ liệu để thu thập một lượng lớn dữ liệu từ một số nguồn. Với mục đích này, chúng tôi đã tìm thấy các bộ dữ liệu khác nhau. Sau khi phân tích chúng tôi quyết địch sử dụng bộ dữ liệu “twitter\_sentiment.csv”. Hầu hết các bộ dữ liệu nguồn mở mà chúng tôi tìm thấy trên internet đều được dán nhãn chính xác và có cấu trúc. Dữ liệu do chúng tôi thu thập cần phải được được dán nhãn đúng cách. Sau đó, chúng ta sẽ đi qua làm sạch, tiền xử lý và tách mẫu thử và các bước huấn luyện dữ liệu.

+ Trước hết, chúng ta phải áp dụng tiền xử lý vào tập dữ liệu của chúng tôi để tránh kết quả không mong muốn. Tập traning có 74680 tweet , tập validation có 1000 tweet được trích xuất sử dụng API Twitter. Các tweet được phân loại thành các nhãn sau: “Negative” , “Neutral”, “Positive” và “Irrelevant”



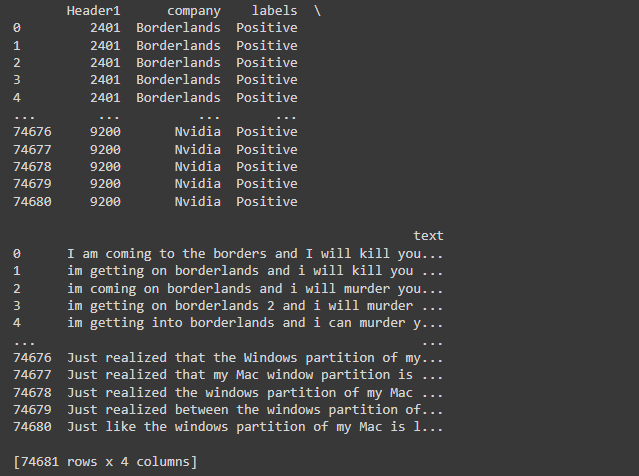
*Ảnh 1:Tập dữ liệu twitter\_sentiment.csv trước khi tiền xử lí*



*Biều đồ 1 :Phân phối dữ liệu của tập dataset*

Từ biều đồ cột chúng ta thấy nhãn “ Negative” có số lượng giá trị nhiều nhất, nhãn “Irrelevant” có số lượng giá trị ít nhất. Sự chênh lệch giữa các nhãn không quá nhiều cho thấy đây là tập dataset chất lượng.

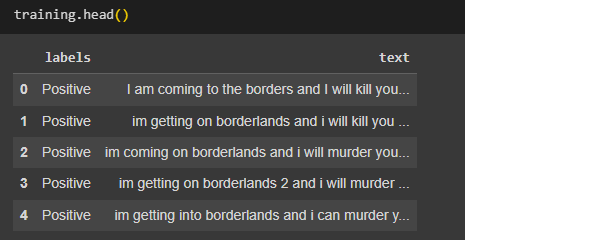
+ Tiếp theo chúng tôi tiến hành thêm tiêu đề cho các cột trong mỗi khung dữ liệu



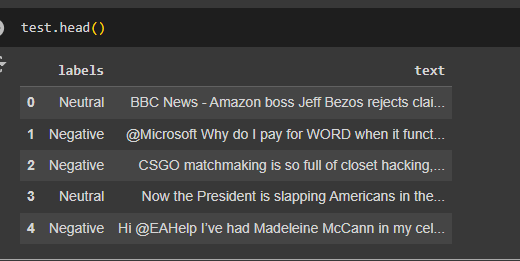
*Ảnh 2: Dataset sau khi thêm tiêu đề cho các cột*

Việc thêm tên các cột này rất hữu ích trong quá trình xử lý dữ liệu khi chúng tôi muốn sử dụng các tên cột rõ ràng và dễ hiểu hơn cho các phân tích và xử lý tiếp theo. Bất kỳ thao tác hoặc phân tích nào sau đó trên các DataFrame này sẽ sử dụng các tên cột mới đã được đặt lại.

+ Tiếp đến chúng tôi loại bỏ các cột không cần thiết từ các DataFrame. Những cột được loại bỏ là những cột không cần thiết để thực hiện các phân tích và các tác vụ khác. Điều này giúp làm sạch dữ liệu và dễ quản lí hơn trong quá trình phân tích

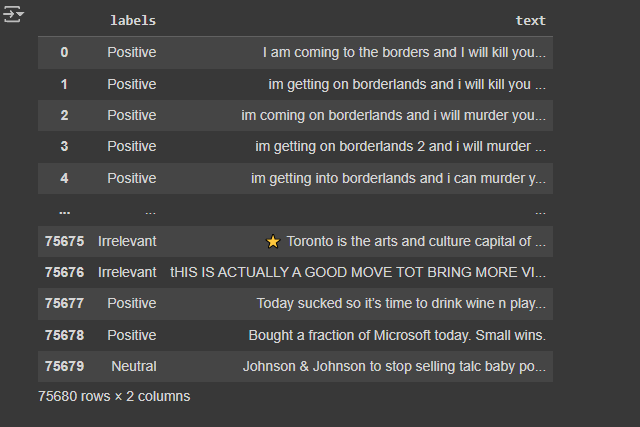


*Ảnh 3:Tập train sau tiền xử lí dữ liệu*

****

*Ảnh 4:Tập test sau tiền xử lí dữ liệu*

+ Sau đó chúng tôi nối 2 DataFrame Traning và test thành 1 DataFrame lớn mới là “ sentiment” bằng hàm “pd.concat”.Việc này giúp tạo ra một tập dữ liệu lớn có thể được sử dụng cho việc huấn luyện mô hình hoặc thực hiện các phân tích dữ liệu phức tạp hơn.



*Ảnh 5: Sentiment*

+ Tiếp theo chúng tôi tiến hành “làm sạch” các đoạn text. Thực hiện việc này bằng cách sử dụng các lệch python để: Loại bỏ khoảng trắng dư thừa**,** loại bỏ ký tự đặc biệt, loại bỏ từ đơn lẻ, loại bỏ ký tự không phải chữ cái, loại bỏ các ký tự không phải là chữ cái từ văn bản, chuyển đổi văn bản thành chữ thường để đồng nhất, Tokenize văn bản thành từng từ, sử dụng WordNetLemmatizer từ NLTK để đưa các từ về dạng gốc, loại bỏ các stop words (những từ phổ biến như "is", "the", "a",...) từ văn bản, loại bỏ các từ có độ dài nhỏ hơn 3 ký tự, loại bỏ các từ trùng lặp. Bước này là bước chuẩn bị dữ liệu cơ bản trong xử lý văn bản để tiếp tục thực hiện các tác vụ phân tích hoặc xây dựng mô hình học máy trong NLP.

+ Sau khi dữ liệu đã dược làm sạch, chúng tôi chia tập dữ liệu thành các tập train và test để chuẩn bị cho việc xây dựng mô hình học máy và huấn luyện.

+ Áp dụng ‘Tokenizer’ để mã hóa các văn bản thành chuỗi số nguyên dựa trên tần suất xuất hiện của từ. Chúng tôi có 24048 số lượng từ khác nhau. Điều này cho thấy dữ liệu của chúng tôi chứa nhiều từ đa dạng và phong phú.

+ Sau đó sử dụng ‘pad\_sequences’ từ ‘tensorflow.keras.preprocessing.sequence’ để đệm pad các chuối số nguyên trong ‘X\_train’ và ‘X\_test’ thành các chuối có độ dài ‘maxlen’. Vì trong quá trình làm việc với các mô hình học sâu (deep learning), đặc biệt là các mô hình xử lý ngôn ngữ tự nhiên (NLP), dữ liệu đầu vào cần phải có kích thước cố định. Tuy nhiên, trong thực tế, các văn bản có độ dài khác nhau. Việc đệm (padding) giúp đảm bảo rằng tất cả các chuỗi đầu vào có cùng độ dài,dễ xử lí dữ liệu và dữ thông tin văn bản quan trọng

Như vậy sao khi tiền xử lí dữ liệu, làm thế nào để dự đoán và phân tích xu hướng tâm lý xã hội từ dữ liệu mạng xã hội?

# V. Thiết kế triển khai mô hình

## 5.1 Các bước xây dụng mô hình

- Xác định các lớp và module cần thiết: Trong bài toán này chúng tôi sử dụng:

+ Sequential: Cho phép bạn xây dựng mô hình lớp này chồng lên lớp khác một cách tuần tự

+ Embedding: Lớp nhúng từ vựng (word embedding layer), chuyển đổi các từ thành vector.

+ LSTM: Lớp LSTM (Long Short-Term Memory), một loại mạng nơ-ron hồi quy (recurrent neural network) phổ biến cho xử lý chuỗi.

+ Dense: Lớp fully connected, mỗi neuron kết nối với tất cả các neuron của lớp trước.

+ GlobalMaxPooling1D: Lớp pooling để giảm chiều không gian đầu vào, giữ lại giá trị tối đa cho mỗi vector.

+ Dropout: Lớp dropout để ngăn ngừa overfitting bằng cách ngẫu nhiên loại bỏ một số neuron trong quá trình huấn luyện.

+ Bidirectional: Lớp LSTM hai chiều để tận dụng thông tin từ cả hai hướng của chuỗi.

+ Optimizers: Để định nghĩa các bộ tối ưu hóa.

- Chọn số chiều của các embedding vectors (D=100)

- Định nghĩa lớp đầu vào của mô hình với chiều dài tối đa ‘maxlen’.

- Thiết lập Learning Rate cho bộ tối ưu Adam:

+ Adam: Bộ tối ưu hóa Adam hường được sử dụng trong học sâu vì nó kết hợp ưu điểm của hai bộ tối ưu hóa phổ biến là AdaGrad và RMSProp.

+ Learning\_Rate: Tốc độ học (learning rate) được đặt là 0.0001. Tốc độ học thấp giúp mô hình học chậm hơn nhưng có khả năng đạt kết quả tốt hơn sau nhiều epoch.

-Xây dựng mô hình:

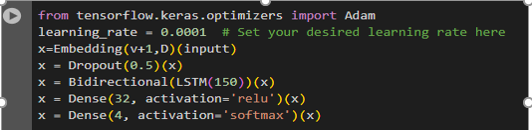
+ Lớp Embedding từ vựng chuyển đổi các từ thành các vector có số chiều là ‘D’, ‘v+1’ là kích thước từ vựng (với v là số lượng từ) cộng thêm một để bao gồm từ điển (indexing).

+ Lớp dropout để ngăn chặn overfitting bằng cách ngẫu nhiên loại bỏ 50% các neuron.

+ Lớp LSTM hai chiều với 150 đơn vị, giúp mô hình có thể học từ cả hai hướng của chuỗi văn bản.

+ Lớp fully connected với 32 neuron và hàm kích hoạt ReLU.

+ Lớp fully connected với 4 neuron và hàm kích hoạt softmax, phù hợp cho bài toán phân loại nhiều lớp (ở đây là 4 lớp).



*Ảnh 6: Các lớp học sâu trong bulid model*

-Định nghĩa và compile mô hình:

+ Định nghĩa bộ tối ưu hóa Adam với tốc độ học đã đặt.

+ Biên dịch mô hình với bộ tối ưu hóa Adam, hàm mất mát categorical\_crossentropy (dùng cho bài toán phân loại nhiều lớp), và độ chính xác (accuracy) là metric đánh giá.

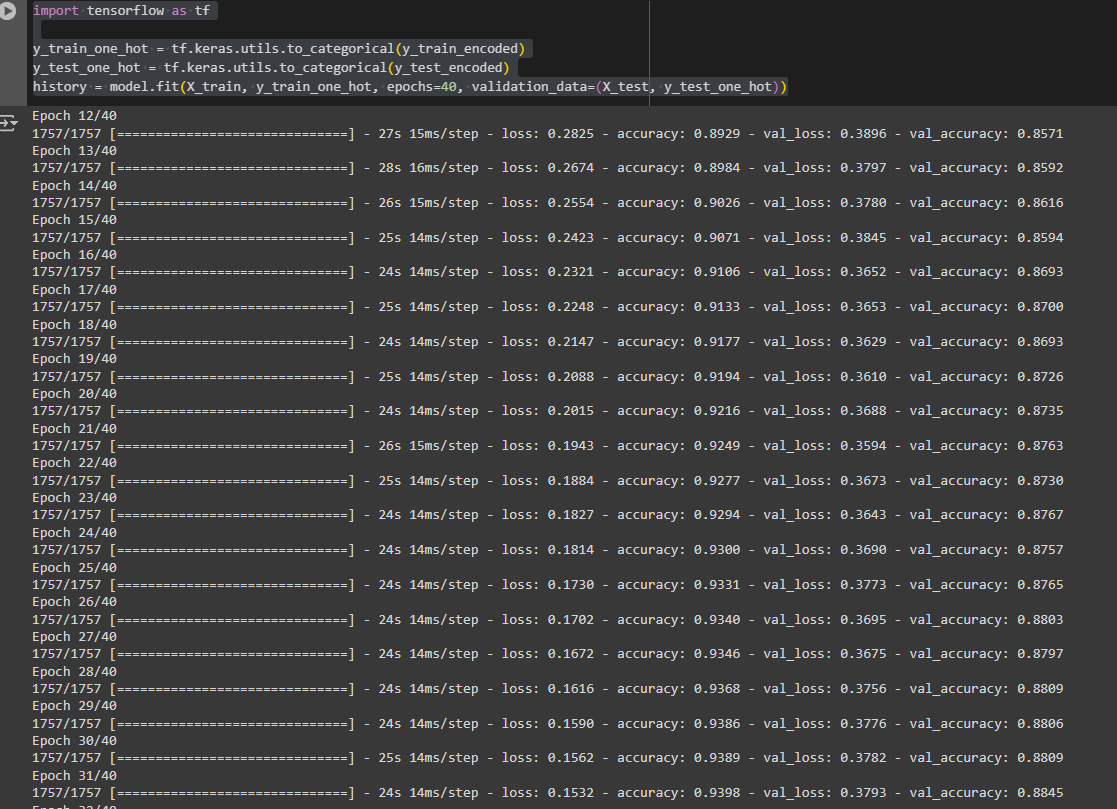
-Sử dụng lớp LabelEncoder từ thư viện scikit-learn dùng để chuyển đổi các nhãn phân loại thành các giá trị số. Ví dụ, các nhãn 'Positive', 'Negative', 'Neutral', và 'Irrelevant' có thể được chuyển thành các giá trị số như 0, 1, 2, và 3 tương ứng.

**-**Dùng phương thức fit\_transformthực hiện hai bước: (1) học ánh xạ từ các nhãn ban đầu đến các giá trị số, và (2) áp dụng ánh xạ đó để chuyển đổi các nhãn. Kết quả là ‘y\_train’ được mã hóa thành các giá trị số và lưu vào ‘y\_train\_encoded’.

## 5.2 Huấn luyện mô hình

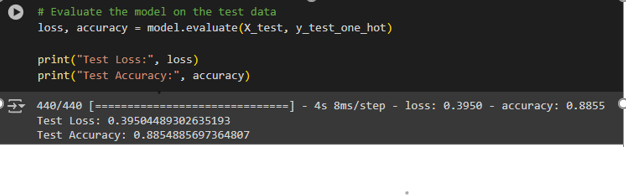
- Chuyển đổi nhãn số thành one-hot vectors: Sử dụng hàm ‘to\_categorical’ để chuyển đổi các nhãn số thành dạng one-hot vectors, phù hợp với yêu cầu của hàm mất mát ‘categorical\_crossentropy’. Mỗi nhãn số sẽ được biểu diễn dưới dạng một vector nhị phân có cùng số lượng phần tử với số lớp (classes).

-Huấn luyện mô hình: Sử dụng phương thức ‘fit’ để huấn luyện mô hình với dữ liệu huấn luyện và kiểm tra mô hình với dữ liệu kiểm tra sau mỗi epoch.



*Ảnh 7: Huấn luyện mô hình*

## 5.3 Đánh giá mô hình

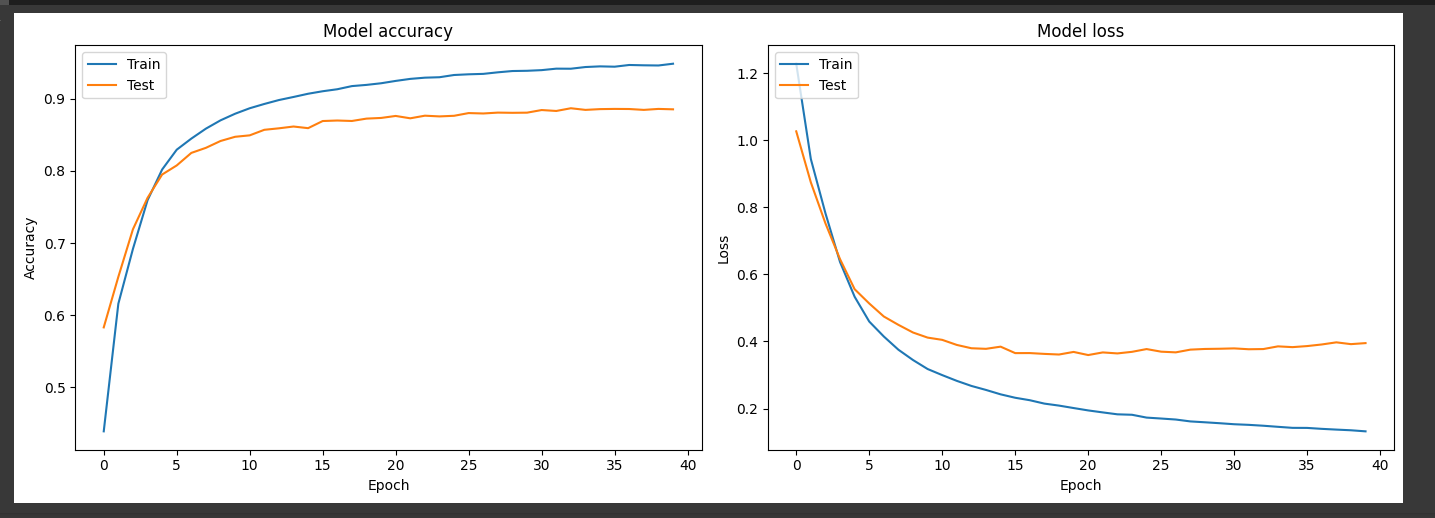


*Ảnh 8: Chỉ số Accuracy và Loss*

Chỉ số accuracy trên tập test là 88,54%

Hàm mất mát là 0,39

Đây là một kết quả khá tốt và cho thấy mô hình có khả năng dự đoán chính xác với mức độ cao trên dữ liệu mới. Tuy nhiên, để đánh giá toàn diện, bạn cần xem xét thêm nhiều yếu tố khác như sự cân bằng giữa các lớp nhãn, khả năng tổng quát hóa của mô hình trên các dữ liệu không quen thuộc, và sự ổn định của các độ đo này qua nhiều lần chạy.



*Biểu đồ 2:Biều đồ biểu thị Accuracy và Loss trên tập train và test*

* Biểu đồ bên trái là biểu đồ thể hiện đô chính xác(accuracy),biểu đồ bên phải là thể hiện hàm mất mát(loss), trên cả tập huấn luyện và tập kiểm tra theo từng epoch.
* Nhận xét:

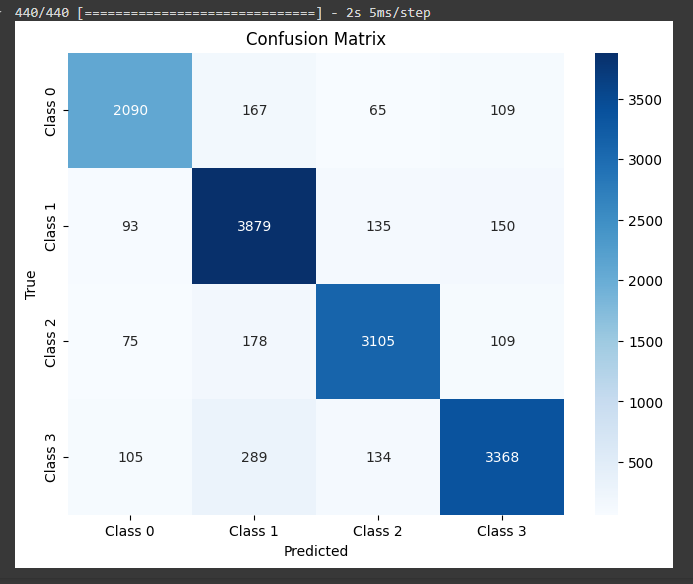
+ Accuracy trên tập train và test đều tăng dần và đạt mức độ ổn định qua từng epochs

+ Accuracy trên tập test cao và gần bằng độ chính xác trên tập train => Mô hình có khả năng tổng quát hóa tốt và không bị overfitting.

+ Hàm mất mát trên tập train và test đều giảm dần và ổn định qua từng epochs

+ Hàm mất mát trên tập test không tăng lên khi độ chính xác trên tập train tăng cao

+ Tổng quát: cả hai biểu đồ đều cho thấy độ chính xác tăng và hàm mất mát giảm dần đồng thời trên cả tập huấn luyện và kiểm tra.



*Ảnh 9: Confusion Matrix*

Dựa trên đường chéo chính (từ trái qua phải) trong Confusion Matrix thể hiện số lượng các điểm dữ liệu mà mô hình đã dự đoán đúng cho từng lớp nhãn. Ta thấy số lượng các điểm mô hình dự đoán đúng là lớn hơn nhiều so với các điểm dữ liệu mà mô hình dự đoán sai.

Kết luận: Mô hình khá tốt và có khả năng dự đoán chính xác trên dữ liệu mới (không bị overfitting)

# VI. Tài liệu tham khảo

https://niithanoi.edu.vn/su-dung-cac-thu-vien-va-framework-de-giai-quyet-cac-van-de-cu-the-trong-python.html

https://www.researchgate.net/publication/268817500\_Sentiment\_Analysis\_for\_Social\_Media

https://viblo.asia/p/recurrent-neural-network-tu-rnn-den-lstm-gGJ597z1ZX2