**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**A blue and yellow logo

Description automatically generated with low confidence**

**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI:**

**Giảng viên hướng dẫn: ThS. TRẦN PHONG NHÃ**

**Sinh viên thực hiện:** BÙI NHẬT HUY

**Lớp:** CQ.61.CNTT

**Khoá:** 61

TP. Hồ Chí Minh, năm 2024

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** Độc lập – Tự do – Hạnh phúc

# NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

**BỘ MÔN: CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

**Mã sinh viên:** 6151071052  **Họ tên SV:** BÙI NHẬT HUY **Khóa:** 61 **Lớp:** CQ. 61.CNTT

**1. Tên đề tài**

**2. Mục tiêu**

Xây dựng mô hình Phân Loại Cảm Xúc Văn Bản để nhận diện bình luận tiêu cực. Từ đó ứng dụng tích hợp vào xây dựng trang web Mạng Xã Hội.

**3. Nội dung thực hiện**

- Tìm hiểu sơ bộ về AI, Machine Learning, Deep Learning, NLP

- Nghiên cứu về mô hình mạng neural Convolutional Neural Network (CNN), Bidirectional Long Short-Term Memory (BiLSTM)

- Nghiên cứu các bài toán về Sentiment Analysis

- Áp dụng kiến thức: ứng dụng các mô hình mạng neural vào phân tích và đưa ra dự đoán về Cảm Xúc Văn Bản

**4. Công nghệ, công cụ và ngôn ngữ lập trình**

- Công nghệ sử dụng: Tensorflow

- Công cụ lập trình: Google Colab, Visual Studio Code

- Ngôn ngữ lập trình: Python, JavaScript

**5. Các kết quả chính dự kiến**

- Phân loại được các bình luận tiêu cực

- Tích hợp được vào trang web Mạng Xã Hội

**6. Kế hoạch thực hiện**

- Tuần 1-2: Tìm và chọn đề tài

- Tuần 3: Xác định đề tài

- Tuần 4-5: Tìm hiểu về AI, Machine Learning, Deep Learning, NLP

- Tuần 6 đến 10: Nghiên cứu về các mô hình mạng neural, các kiến thức liên quan và áp dụng kiến thức vào bài toán

- Tuần 11-12: Viết báo cáo và slide

- Tuần 13: Nộp báo cáo và chờ duyệt

**7. Giảng viên và cán bộ hướng dẫn**

Họ tên: ThS. TRẦN PHONG NHÃ

Đơn vị công tác: Trường Đại học Giao thông Vận tải Phân hiệu tại TP. Hồ Chí Minh

Điện thoại: 0906 761 014 Email: [tpnha@utc2.edu.vn](mailto:tpnha@utc2.edu.vn)

|  |  |
| --- | --- |
| **Ngày…..tháng…..năm 2024**  **Trưởng BM Công nghệ thông tin** | **Đã giao nhiệm vụ TKTN**  **Giảng viên hướng dẫn** |
| **Trần Phong Nhã** | **Trần Phong Nhã** |

# LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc đến Ban Giám Hiệu, các thầy cô giáo trong **Bộ môn Công Nghệ Thông Tin - Trường Đại học Giao thông Vận tải phân hiệu tại TP. Hồ Chí Mình** đã tạo điều kiện thuận lợi và cung cấp cho em những kiến thức, kỹ năng cần thiết trong suốt quá trình học tập và thực hiện đồ án tốt nghiệp.

Đặc biệt em xin chân thành cảm ơn thầy **Trần Phong Nhã**, người đã tận tình hướng dẫn, đóng góp ý kiến quý báu và giúp đỡ em trong suốt quá trình thực hiện và hoàn thiện đồ án này. Sự nhiệt tình, tâm huyết và những kinh nghiệm quý báu của thầy đã giúp em,vượt qua những khó khăn và hoàn thành đồ án một cách tốt nhất.

Em cũng xin gửi lời cảm ơn đến tất cả các bạn bè đã luôn động viên, hỗ trợ và cùng em chia sẻ những kinh nghiệm, ý tưởng trong suốt quá trình thực hiện đồ án.

Vì thời gian làm đề tài có hạn cũng như hiểu biết còn hạn chế, em cũng đã nỗ lực hết sức để hoàn thành đề tài một cách tốt nhất, nhưng chắc chắn vẫn sẽ có những thiếu sót không thể tránh khỏi. Em kính mong nhận được sự thông cảm và những ý kiến đóng góp chân thành từ quý thầy cô.

Sau cùng, em xin kính chúc Quý Thầy Cô trong Bộ môn Công nghệ thông tin hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Em xin chân thành cảm ơn!

**TP. Hồ Chí Minh, ngày ….… tháng ….… năm 2024**

**Sinh viên thực hiện**

**Bùi Nhật Huy**

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Tp. Hồ Chí Minh, ngày … tháng …. năm 2024**

**Giảng viên hướng dẫn**

**Trần Phong Nhã**

# MỤC LỤC

[NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP i](#_Toc168442314)

[LỜI CẢM ƠN iii](#_Toc168442315)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN iv](#_Toc168442316)

[MỤC LỤC v](#_Toc168442317)

[DANH MỤC HÌNH ẢNH vii](#_Toc168442318)

[DANH MỤC BẢNG viii](#_Toc168442319)

[DANH MỤC BẢNG VIẾT TẮT ix](#_Toc168442320)

[TỔNG QUAN 1](#_Toc168442321)

[1. Lý do chọn đề tài 1](#_Toc168442322)

[2. Mục tiêu và nhiệm vụ của đồ án 2](#_Toc168442323)

[3. Bố cục đồ án 2](#_Toc168442324)

[CHƯƠNG I. CƠ SỞ LÝ THUYẾT 3](#_Toc168442325)

[1. Tổng quan về AI – Machine Learning 3](#_Toc168442326)

[a. AI là gì? 3](#_Toc168442327)

[b. Machine Learning là gì? 4](#_Toc168442328)

[2. Xử lý ngôn ngữ tự nhiên 5](#_Toc168442329)

[a. Định nghĩa 5](#_Toc168442330)

[b. Các bài toán và ứng dụng 6](#_Toc168442331)

[3. Sentiment Analysis 7](#_Toc168442332)

[a. Định nghĩa 7](#_Toc168442333)

[b. Các ứng dụng của Sentiment Analysis 8](#_Toc168442334)

[c. Sentiment Analysis hoạt động như thế nào? 9](#_Toc168442335)

[4. Deep Learning 12](#_Toc168442336)

[a. Deep Learning là gì? 12](#_Toc168442337)

[b. Mạng Neural nhân tạo 13](#_Toc168442338)

[c. CNN 15](#_Toc168442339)

[5. BiLSTM 16](#_Toc168442340)

[a. LSTM 16](#_Toc168442341)

[b. BiLSTM 19](#_Toc168442342)

[CHƯƠNG II. XÂY DỰNG MÔ HÌNH 21](#_Toc168442343)

[1. Chuẩn bị tập dữ liệu, môi trường và các thư viện cần thiết 21](#_Toc168442344)

[a. Môi trường triển khai 21](#_Toc168442345)

[b. Tập dữ liệu 21](#_Toc168442346)

[c. Các thư viện. 22](#_Toc168442347)

[2. Tiền xử lý dữ liệu 24](#_Toc168442348)

[3. Xây dựng mô hình CNN và BiLSTM 26](#_Toc168442349)

[4. Huấn luyện mô hình và thử nghiệm trên tập Test 26](#_Toc168442350)

[5. Dự đoán với mô hình đã huấn luyện 27](#_Toc168442351)

[CHƯƠNG III. THỰC NGHIỆM MÔ HÌNH VÀ ÁP DỤNG VÀO TRANG WEB MẠNG XÃ HỘI 28](#_Toc168442352)

[1. 28](#_Toc168442353)

[2. 28](#_Toc168442354)

[3. 28](#_Toc168442355)

[4. 28](#_Toc168442356)

[5. 28](#_Toc168442357)

[KẾT LUẬN VÀ KIẾN NGHỊ 28](#_Toc168442358)

[I. Kết quả đạt được 28](#_Toc168442359)

[II. Những vấn đề còn tồn đọng 28](#_Toc168442360)

[III. Hướng phát triển tiếp theo của đề tài 28](#_Toc168442361)

[TÀI LIỆU THAM KHẢO 29](#_Toc168442362)

# DANH MỤC HÌNH ẢNH

# DANH MỤC BẢNG

# DANH MỤC BẢNG VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Viết tắt | Diễn giải | Ý nghĩa |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# TỔNG QUAN

## 1. Lý do chọn đề tài

Cùng với sự phát triển mạnh mẽ của công nghệ thông tin và Internet, các diễn đàn mạng xã hội phổ biến ngày nay như: Facebook, Instagram, Linkedin, Twitter, Zalo, Tiktok, Snopchat, … đều là các nền tảng rất thông dụng với người dùng và chúng hoạt động tiếp diễn hằng ngày. Thông qua các công cụ ứng dụng chúng như một phương tiện hữu ích giúp người dùng bày tỏ cảm xúc thông qua các trang mạng một cách dễ dàng hơn, cũng như chia sẻ những trải nghiệm, bình luận và đánh giá trong quá trình trải nghiệm.

Dễ thấy rằng việc thể hiện cảm xúc là nhu cầu cơ bản của con người và chúng ta sử dụng ngôn ngữ không chỉ để truyền đạt suy nghĩ mà còn có cả cảm xúc của chúng ta. Cảm xúc quyết định chất lượng cuộc sống của chúng ta, và chúng ta tổ chức cuộc sống của mình để tối đa hóa trải nghiệm của cảm xúc tích cực và giảm thiểu trải nghiệm của cảm xúc tiêu cực.

Do đó, nhà tâm lý học Paul Ekman đã đề xuất sáu cảm xúc cơ bản của con người bao gồm cả sự thích thú, buồn bã, tức giận, ngạc nhiên, sợ hãi và ghê tởm qua nét mặt. Tuy nhiên, ngoài biểu hiện trên khuôn mặt, nhiều nguồn thông tin khác nhau có thể được sử dụng để phân tích cảm xúc vì nhận biết cảm xúc đã nổi lên như một lĩnh vực nghiên cứu quan trọng. Và trong những năm gần đây, nhận dạng cảm xúc trong văn bản đã trở nên phổ biến hơn do những ứng dụng tiềm năng to lớn của nó trong tiếp thị, bảo mật, tâm lý học, tương tác giữa con người với máy tính, trí tuệ nhân tạo, ...v.v.

Trong nghiên cứu này, em tập trung vào vấn đề nhận biết cảm xúc đối với bình luận của người Việt trên mạng xã hội. Cụ thể hơn, đầu vào của bài toán là một bình luận bằng tiếng Việt từ mạng xã hội và đầu ra là cảm xúc dự đoán của bình luận đó được gán nhãn một trong những thứ sau: thích thú, buồn bã, tức giận, ngạc nhiên, sợ hãi, ghê tởm và cảm xúc khác.

Chính vì vậy để có thể thấu hiểu cảm xúc người dùng thông qua ý kiến tích cực hay tiêu cực qua quá trình trải nghiệm là một trong những vấn đề quan trọng. Giải pháp cho vấn đề này, nghiên cứu đề xuất phương pháp khai thác ý kiến và phân tích cảm xúc người dùng thông qua việc thu thập tập dữ liệu là ý kiến bình luận của khách hàng trên các website trực tuyến. Sau đó, tiến hành thực nghiệm bằng phương pháp học máy để khai phá ý kiến từ bình luận dạng văn bản của người dùng và trực quan hóa kết quả hỗ trợ ra quyết định. Kết quả thực nghiệm cho thấy độ chính xác hơn 67% của phương pháp đề xuất và kết quả khai thác được tập thông tin, tri thức tiềm ẩn có giá trị từ tập ngữ liệu nhằm giúp việc nghiên cứu trở nên trực quan và dễ dàng cải thiện chiến lược.

Hiện nay, bài toán phân tích ý kiến cảm xúc theo người dùng được quan tâm ở rất nhiều lĩnh vực khác nhau, từ giáo dục đến khảo sát ý kiến xã hội và hơn hết là lĩnh vực dịch vụ/kinh doanh. Hầu hết các bộ ngữ liệu cũng như các thuật toán được xây dựng và thử nghiệm trên nhiều ngôn ngữ khác nhau như tiếng Anh, tiếng Trung Quốc...v.v. Tuy nhiên đối với tiếng Việt, nhằm trong phạm vi của bài báo cáo, giới hạn của em sẽ chỉ nằm ở mỗi khía cạnh cấp độ câu văn.

## 2. Mục tiêu và nhiệm vụ của đồ án

Xây dựng mô hình Phân Loại Cảm Xúc Văn Bản để nhận diện bình luận tiêu cực. Từ đó ứng dụng tích hợp vào xây dựng trang web Mạng Xã Hội.

## 3. Bố cục đồ án

TỔNG QUAN

CHƯƠNG 1: CƠ SỔ LÝ THUYẾT

CHƯƠNG 2: XÂY DỰNG MÔ HÌNH

CHƯƠNG 3: THỰC NGHIỆM MÔ HÌNH VÀ ÁP DỤNG VÀO TRANG WEB MẠNG XÃ HỘI

KẾT LUẬN VÀ KIẾN NGHỊ

# CHƯƠNG I. CƠ SỞ LÝ THUYẾT

## 1. Tổng quan về AI – Machine Learning

### a. AI là gì?

Trí tuệ nhân tạo (AI) trong khoa học máy tính là trí thông minh được thể hiện bằng máy móc, trái ngược với trí thông minh tự nhiên của con người. AI thường được dùng để mô tả máy móc có khả năng bắt chước các chức năng nhận thức của con người như học tập và giải quyết vấn đề. Khi máy móc phát triển, các nhiệm vụ đòi hỏi trí thông minh thường bị loại khỏi định nghĩa AI, hiện tượng này gọi là hiệu ứng AI.

Ban đầu, AI phát triển với các thuật toán mô phỏng lý luận từng bước của con người. Vào cuối thập kỷ 1980 và 1990, AI tiến triển với phương pháp xử lý thông tin không chắc chắn, sử dụng xác suất và kinh tế. Các vấn đề khó yêu cầu phần cứng mạnh để tránh "vụ nổ tổ hợp". AI sau này cũng tiếp cận phương pháp "biểu tượng phụ" và mạng thần kinh, mô phỏng cấu trúc não bộ và khả năng của con người.

Các lĩnh vực điển hình áp dụng Trí tuệ nhân tạo:

* Học Máy (Machine Learning)
* Xử lý Ngôn ngữ Tự nhiên (Natural Language Processing - NLP)
* Thị giác Máy Tính (Computer Vision)
* Hệ Thống Chuyên Gia (Expert Systems)
* Robot Học (Robotics)
* Mạng Neural Nhân Tạo (Artificial Neural Networks)

### b. Machine Learning là gì?

Tính đến thời điểm hiện tại, có rất nhiều định nghĩa về machine learning. Machine learning (ML) hay máy học là một nhánh của trí tuệ nhân tạo (AI), nó là một lĩnh vực nghiên cứu cho phép máy tính có khả năng cải thiện chính bản thân chúng dựa trên dữ liệu mẫu (training data) hoặc dựa vào kinh nghiệm (những gì đã được học). Machine learning có thể tự dự đoán hoặc đưa ra quyết định mà không cần được lập trình cụ thể.

Bài toán machine learning thường được chia làm hai loại là dự đoán (prediction) và phân loại (classification). Các bài toán dự đoán như dự đoán giá nhà, giá xe… Các bài toán phân loại như nhận diện chữ viết tay, nhận diện đồ vật…

Phân loại Machine Learning:

**Supervised Learning (Học có giám sát):** Supervised Learning (Học có giám sát) là một nhóm thuật toán sử dụng dữ liệu được gán nhãn nhằm mô hình hóa mối quan hệ giữa biến đầu vào (x) và biến đầu ra (y). Hai nhóm bài toán cơ bản trong học có giám sát là classification (phân loại) và regression (hồi quy), trong đó biến đầu ra của bài toán phân loại có các giá trị rời rạc trong khi biến đầu ra của bài toán hồi quy có các giá trị liên tục. Với Supervised Learning, bên cạnh xây dựng các mô hình mạnh, việc thu thập và gán nhãn dữ liệu tốt và hợp lý cũng đóng vai trò then chốt để giải quyết các bài toán trong thực tế.

Một số thuật toán Supervised Learning phổ biến :

* *Linear Regression (Hồi quy tuyến tính)*
* *Logistic Regression (Hồi quy logistic)*
* *Classification (Phân loại)*
* *Naive Bayes Classifier (Phân loại Naïve Bayes)*
* *K-Nearest Neighbors*
* *Decision Tree (Cây quyết định)*
* *Support Vector Machine*

**Unsupervised Learning (Học không có giám sát):** Ngược lại với Supervised Learning, Unsupervised Learning (Học không giám sát) là một nhóm thuật toán sử dụng dữ liệu không có nhãn. Các thuật toán theo cách tiếp cận này hướng đến việc mô hình hóa được cấu trúc hay thông tin ẩn trong dữ liệu. Hay nói cách khác, sử dụng các phương pháp này thiên về việc mô tả tính chất hay đặc tính của dữ liệu. Thông thường, các thuật toán này dựa trên những thông tin sau:

* Mối quan hệ tương tự (similarity) giữa các ví dụ (được gọi là instance) trong dữ liệu như trong các thuật toán clustering (phân cụm)
* Xác suất đồng xuất hiện của các đối tượng như trong Association mining
* Các phép biến đổi ma trận để trích xuất các đặc trưng như PCA, SVD.

Một số thuật toán Unsupervised Learning phổ biến:

* *K-Means (Phân cụm K-Means)*
* *DBSCAN (Phân cụm dựa trên mật độ)*
* *Spectral Clustering (Phân vùng quang phổ)*
* *Hierarchical Clustering (Phân cụm phân cấp)*
* *Apriori*
* *SVD (Phân tích suy biến)*

**Reinforcement Learning (Học tăng cường):** Học tăng cường (RL) là kỹ thuật máy học (ML) giúp đào tạo phần mềm đưa ra quyết định nhằm thu về kết quả tối ưu nhất. Kỹ thuật này bắt chước quy trình học thử và sai mà con người sử dụng để đạt được mục tiêu đã đặt ra. RL giúp phần mềm tăng cường các hành động hướng tới mục tiêu, đồng thời bỏ qua các hành động làm xao lãng mục tiêu.

Một số thuật toán Reinforcement Learning phổ biến:

* *Q-Learning*
* *Hybrid (Mô hình tương tác kết hợp)*
* *PPO (Proximal Policy Optimization)*
* *Multi-agent System (Hệ thống Đa tác nhân)*

## 2. Xử lý ngôn ngữ tự nhiên

### a. Định nghĩa

Xử lý ngôn ngữ tự nhiên (NLP) là một nhánh của Trí tuệ nhân tạo tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói hoặc văn bản. Mục tiêu của NLP là giúp máy tính hiểu và thực hiện hiệu quả các nhiệm vụ liên quan đến ngôn ngữ con người, như tương tác người-máy, cải thiện giao tiếp giữa con người, và nâng cao hiệu quả xử lý văn bản và lời nói. NLP ra đời từ những năm 1940 và đã trải qua nhiều giai đoạn phát triển với các phương pháp khác nhau: ô-tô-mát và mô hình xác suất (những năm 1950), phương pháp dựa trên ký hiệu và ngẫu nhiên (những năm 1970), học máy truyền thống (đầu thế kỷ 21), và học sâu trong thập kỷ vừa qua.

Xử lý ngôn ngữ tự nhiên (NLP) được chia thành hai nhánh lớn: xử lý tiếng nói và xử lý văn bản:

- Xử lý tiếng nói tập trung vào các thuật toán xử lý ngôn ngữ ở dạng âm thanh, bao gồm nhận dạng tiếng nói (chuyển từ tiếng nói sang văn bản) và tổng hợp tiếng nói (chuyển từ văn bản sang tiếng nói).

- Xử lý văn bản tập trung vào phân tích dữ liệu văn bản với các ứng dụng như tìm kiếm thông tin, dịch máy, tóm tắt văn bản tự động và kiểm lỗi chính tả. Xử lý văn bản có thể chia thành hiểu văn bản (phân tích) và sinh văn bản (tạo văn bản mới).

### b. Các bài toán và ứng dụng

**Nhận dạng tiếng nói (Automatic Speech Recognition – ASR, hoặc Speech To Text – STT)** chuyển đổi ngôn ngữ từ dạng tiếng nói sang dạng văn bản, thường được ứng dụng trong các chương trình điều khiển qua giọng nói.

**Tổng hợp tiếng nói (Speech synthesis hoặc Text to Speech – TTS)** chuyển đổi ngôn ngữ từ dạng văn bản sang tiếng nói, thường được dùng trong đọc văn bản tự động.

**Truy xuất thông tin (Information Retrieval – IR)** có nhiệm vụ tìm các tài liệu dưới dạng không có cấu trúc (thường là văn bản) đáp ứng nhu cầu về thông tin từ những nguồn tổng hợp lớn. Những hệ thống truy xuất thông tin phổ biến nhất bao gồm các công cụ tìm kiếm như Google, Yahoo, hoặc Bing search. Những công cụ này cho phép tiếp nhận một câu truy vấn dưới dạng ngôn ngữ tự nhiên làm đầu vào và cho ra một danh sách các tài liệu được sắp xếp theo mức độ phù hợp.

**Phân tích cảm xúc (Sentiment Analysis)** sử dụng kỹ thuật xử lý ngôn ngữ tự nhiên và học máy để xác định cảm xúc tích cực, tiêu cực hoặc trung tính từ văn bản. Ứng dụng bao gồm quản lý quan hệ khách hàng, phân tích thị trường, và theo dõi cảm xúc trên mạng xã hội. Công cụ phổ biến gồm NLTK, TextBlob, và BERT.

**Trả lời câu hỏi (Question Answering – QA)** có khả năng tự động trả lời câu hỏi của con người ở dạng ngôn ngữ tự nhiên bằng cách truy xuất thông tin từ một tập hợp tài liệu. Một hệ thống QA đặc trưng thường bao gồm ba mô đun: Mô đun xử lý truy vấn (Query Processing Module) – tiến hành phân loại câu hỏi và mở rộng truy vấn; Mô đun xử lý tài liệu (Document Processing Module) – tiến hành truy xuất thông tin để tìm ra tài liệu thích hợp; và Mô hình xử lý câu trả lời (Answer Processing Module) – trích chọn câu trả lời từ tài liệu đã được truy xuất.

**Tóm tắt văn bản tự động (Automatic Text Summarization)** là bài toán thu gọn văn bản đầu vào để cho ra một bản tóm tắt ngắn gọn với những nội dung quan trọng nhất của văn bản gốc. Có hai phương pháp chính trong tóm tắt, là phương pháp trích xuất (extractive) và phương pháp tóm lược ý (abstractive). Những bản tóm tắt trích xuất được hình thành bằng cách ghép một số câu được lấy y nguyên từ văn bản cần thu gọn. Những bản tóm lược ý thường truyền đạt những thông tin chính của đầu vào và có thể sử dụng lại những cụm từ hay mệnh đề trong đó, nhưng nhìn chung được thể hiện ở ngôn ngữ của người tóm tắt.

**Chatbot** là việc chương trình máy tính có khả năng trò chuyện (chat), hỏi đáp với con người qua hình thức hội thoại dưới dạng văn bản (text). Chatbot thường được sử dụng trong ứng dụng hỗ trợ khách hàng, giúp người dùng tìm kiếm thông tin sản phẩm, hoặc giải đáp thắc mắc.

**Dịch máy (Machine Translation – MT)** là việc sử dụng máy tính để tự động hóa một phần hoặc toàn bộ quá trình dịch từ ngôn ngữ này sang ngôn ngữ khác. Các phương pháp dịch máy phổ biến bao gồm dịch máy dựa trên ví dụ (example-based machine translation – EBMT), dịch máy dựa trên luật (rule-based machine translation – RBMT), dịch máy thống kê (statistical machine translation – SMT), và dịch máy sử dụng mạng nơ-ron (neural machine translation).

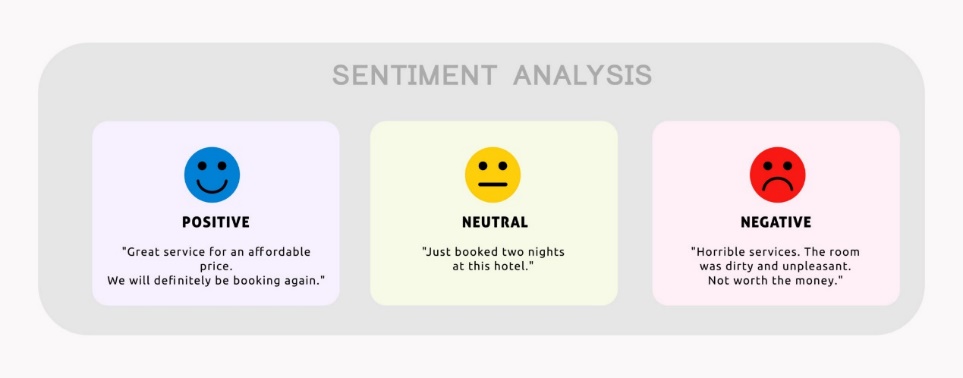
**Kiểm lỗi chính tả tự động** là việc sử dụng máy tính để tự động phát hiện các lỗi chính tả trong văn bản (lỗi từ vựng, lỗi ngữ pháp, lỗi ngữ nghĩa) và đưa ra gợi ý cách chỉnh sửa lỗi.

## 3. Sentiment Analysis

### a. Định nghĩa

Phân tích cảm nghĩ hay Sentiment Analysis là một tập hợp con của xử lý ngôn ngữ tự nhiên (NLP) sử dụng học máy để phân tích và phân loại giọng điệu cảm xúc của dữ liệu văn bản. Các mô hình cơ bản chủ yếu tập trung vào việc phân loại các trạng thái tích cực, tiêu cực và trung lập nhưng cũng có thể xét đến những cảm xúc cơ bản của người nói (vui mừng, tức giận, phẫn nộ), cũng như các ý định mua hàng.

Bối cảnh sẽ thêm sự phức tạp vào quá trình Phân tích cảm nghĩ. Ví dụ: câu cảm thán “không có gì!” có ý nghĩa khác nhau đáng kể tùy thuộc vào việc người nói đang bình luận về những gì người đó thích hay không thích về một sản phẩm. Để hiểu cụm từ “Tôi thích nó”, hệ thống phải có khả năng gỡ rối ngữ cảnh để hiểu “nó” ám chỉ điều gì. Ý nghĩ “mỉa mai” cũng rất phổ biến vì người nói có thể đang nói điều gì đó tích cực nhưng lại có ý ngược lại.



Hình . Sentiment Analysis

Nguồn: dev.to

### b. Các ứng dụng của Sentiment Analysis

Một số ví dụ về ứng dụng của Sentiment Analysis bao gồm:

* Các nhà thiết kế sản phẩm sử dụng Sentiment Analysis để xác định tính năng nào đang gây được tiếng vang với khách hàng và do đó xứng đáng được đầu tư và tập trung hơn. Ngược lại, họ có thể tìm hiểu khi nào một sản phẩm hoặc tính năng không còn giá trị và điều chỉnh để ngăn chặn tồn kho tăng cao.
* Các công ty tiếp thị chủ yếu dựa vào Phân tích cảm nghĩ để giúp họ điều chỉnh thông điệp, khám phá những người có ảnh hưởng trên mạng (KoLs) và xây dựng chiến dịch marketing truyền miệng tích cực.
* Các tổ chức bán lẻ khai thác Phân tích cảm nghĩ để xác định sản phẩm nào có khả năng bán chạy và điều chỉnh hàng tồn kho cũng như khuyến mãi cho phù hợp.
* Các nhà đầu tư có thể xác định các xu hướng mới nổi lên trong các cuộc trò chuyện trực tuyến, dự báo trước các cơ hội thị trường.
* Các chính trị gia sử dụng nó để lấy mẫu thái độ của cử tri về các vấn đề quan trọng.

### c. Sentiment Analysis hoạt động như thế nào?

* **Feature Engineering trong học máy**

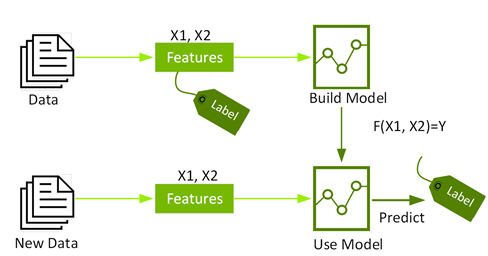
Feature Engineering là quá trình chuyển đổi dữ liệu thô thành đầu vào cho thuật toán học máy. Để được sử dụng trong các thuật toán học máy, các đặc điểm (feature) phải được đưa vào các vectors đặc trưng, ​​là các vectors số đại diện cho giá trị của từng đặc điểm. Để phân tích cảm nghĩ, dữ liệu văn bản phải được đưa vào các vectors từ ngữ, là vectors của các số biểu thị giá trị cho mỗi ký tự. Văn bản đầu vào có thể được mã hóa thành các vectors ký tự bằng cách sử dụng các kỹ thuật đếm như Bag of Words (BoW), bag-of-ngrams hoặc Term Frequency/Inverse Document Frequency (TF-IDF).



Hình . Feature Engineering

* **Sử dụng học máy có giám sát (Supervised Learning)**

Sau khi văn bản đầu vào đã được chuyển đổi thành vectors từ ngữ, thuật toán học máy phân loại có thể được sử dụng để phân loại cảm xúc. Phân loại là một nhóm các thuật toán học máy được giám sát để xác định chủng loại nào mà một item thuộc về (chẳng hạn như văn bản là tiêu cực hay tích cực) dựa trên dữ liệu được gắn nhãn (chẳng hạn như văn bản được gắn nhãn là tích cực hay tiêu cực).



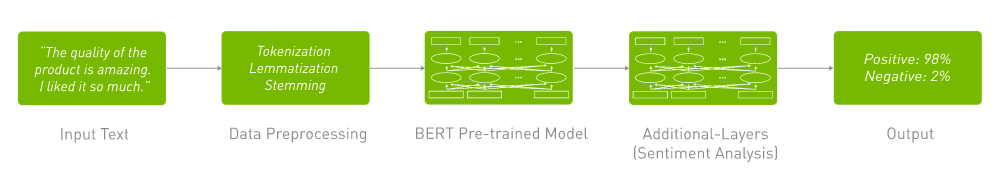
Hình . Sử dụng Supervised Learning

Nguồn: nvidia.com

Các thuật toán học máy phân loại có thể được sử dụng để phân tích cảm nghĩ bao gồm:

* Naïve Bayes là một bộ các thuật toán xác suất xác định xác suất có điều kiện của lớp dữ liệu đầu vào.
* Support Vector Machines tìm thấy một “hyperplane” trong không gian N chiều (N là số lượng đối tượng địa lý) phân loại rõ ràng các điểm dữ liệu.
* Logistic regression sử dụng hàm logistic để mô hình hóa xác suất của một cấp nhất định.
* **Phân tích cảm nghĩ bằng cách sử dụng Deep Learning**

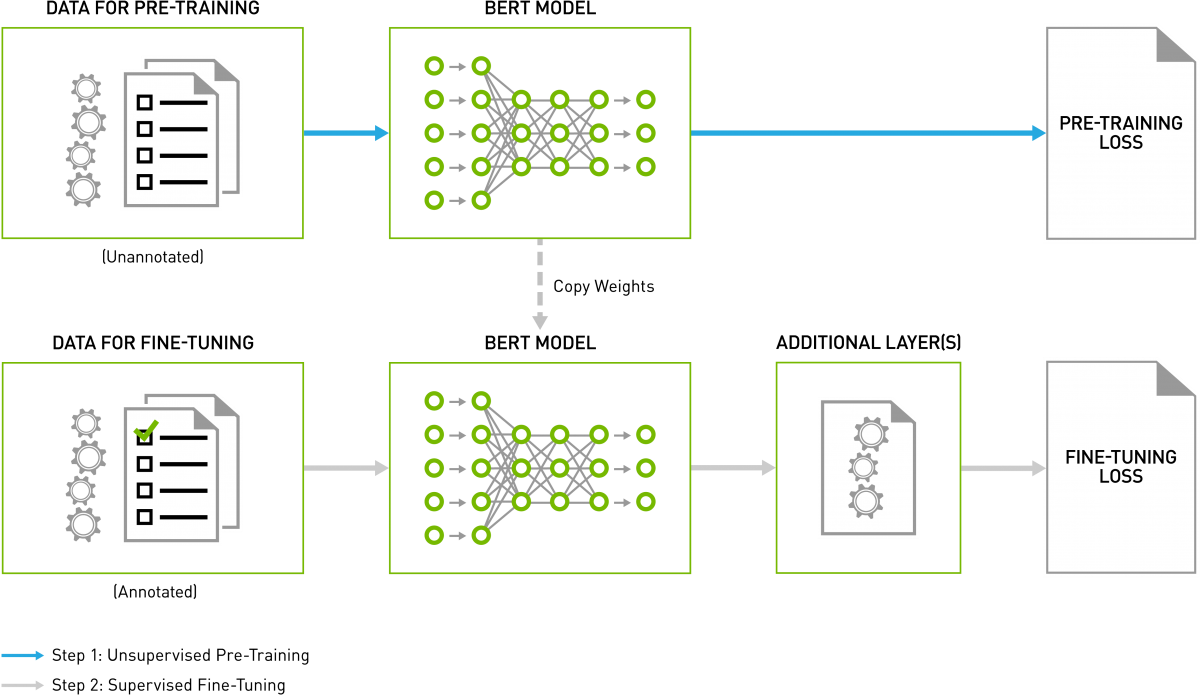
Học sâu (DL) là một tập hợp con của học máy (ML) sử dụng các neural networks để mang lại độ chính xác cao nhất trong các tác vụ như NLP và các tác vụ khác. Các kỹ thuật nhúng từ DL chẳng hạn như Word2Vec mã hóa các ký tự theo những cách có ý nghĩa bằng cách học các liên kết ký tự, ý nghĩa, ngữ nghĩa và cú pháp . Các thuật toán DL cũng cho phép đào tạo từ đầu đến cuối các mô hình NLP mà không cần phải thiết kế thủ công các tính năng từ dữ liệu thô đầu vào.



Hình . Sử dụng Deep Learning

Nguồn: nvidia.com

Có nhiều biến thể của thuật toán học sâu. Recurrent neural networks (RNNs) là công cụ toán học để phân tích các mẫu ngôn ngữ và dữ liệu theo trình tự, được dùng trong xử lý ngôn ngữ tự nhiên như thính giác và lời nói cho Alexa, dịch thuật ngôn ngữ, dự đoán chứng khoán và giao dịch thuật toán. BERT (Đại diện bộ mã hóa hai chiều từ Transformer) là một giải pháp thay thế cho RNNs, áp dụng kỹ thuật phân tích cú pháp câu bằng cách tập trung vào các ký tự liên quan trước và sau nó. BERT đã cách mạng hóa NLP bằng cách cung cấp độ chính xác cao, có thể so sánh với con người trong việc nhận biết ý định, phân tích tình cảm, và giữ ngữ cảnh tốt hơn. Một thách thức chính với các mô hình ngôn ngữ là thiếu dữ liệu được dán nhãn, nhưng BERT được đào tạo trên các nhiệm vụ không giám sát và sử dụng bộ dữ liệu phi cấu trúc từ kho sách, Wikipedia tiếng Anh, v.v.



Hình . Sử dụng BERT Model

Nguồn: nvidia.com

## 4. Deep Learning

### a. Deep Learning là gì?

Deep Learning hay Học sâu là một phương thức trong lĩnh vực trí tuệ nhân tạo (AI) cũng có thể được coi là một lĩnh vực thuộc Machine Learning ,được sử dụng để dạy máy tính xử lý dữ liệu theo cách được lấy cảm hứng từ bộ não con người. Mô hình học sâu có thể nhận diện nhiều hình mẫu phức tạp trong hình ảnh, văn bản, âm thanh và các dữ liệu khác để tạo ra thông tin chuyên sâu và dự đoán chính xác. Ta có thể sử dụng các phương pháp học sâu để tự động hóa các tác vụ thường đòi hỏi trí tuệ con người, chẳng hạn như mô tả hình ảnh hoặc chép lời một tập tin âm thanh. Chủ yếu hoạt động với mạng thần kinh nhân tạo để bắt chước khả năng tư duy và suy nghĩ của bộ não con người. Thực ra các khái niệm liên quan đến mạng nơ-ron nhân tạo và Deep Learning đã được phát triển những năm 1960. Nhưng nó bị giới hạn bởi lượng dữ liệu và khả năng tính toán tại thời điểm đó.

Mô hình học sâu có một số trường hợp sử dụng trong lĩnh vực ô tô, hàng không vũ trụ, sản xuất, điện tử, nghiên cứu y học và nhiều lĩnh vực khác. Sau đây là một vài ví dụ về học sâu:

* Xe tự lái sử dụng các mô hình học sâu để tự động phát hiện biển báo giao thông và người đi bộ.
* Hệ thống quốc phòng sử dụng mô hình học sâu để tự động gắn cờ các khu vực được quan tâm trong ảnh vệ tinh.
* Phân tích hình ảnh y khoa sử dụng học sâu để tự động phát hiện các tế bào ung thư trong chẩn đoán y tế.
* Các nhà máy sử dụng ứng dụng học sâu để tự động phát hiện con người hoặc vật thể khi những đối tượng này đang nằm trong khoảng cách không an toàn của máy móc.

Có thể nhóm các trường hợp sử dụng mô hình học sâu khác nhau thành 4 loại chính: thị giác máy tính, nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên (NLP) và công cụ đề xuất

**- Thị giác máy tính:** là khả năng của máy tính thực hiện trích xuất dữ liệu cũng như thông tin chuyên sâu từ hình ảnh và video. Máy tính có thể sử dụng các kỹ thuật học sâu để hiểu hình ảnh theo cách giống như con người.

**- Nhận dạng giọng nói:** Các mô hình học sâu có thể phân tích giọng nói con người, bất kể mẫu giọng, cao độ, tông, ngôn ngữ và giọng vùng miền khác nhau.

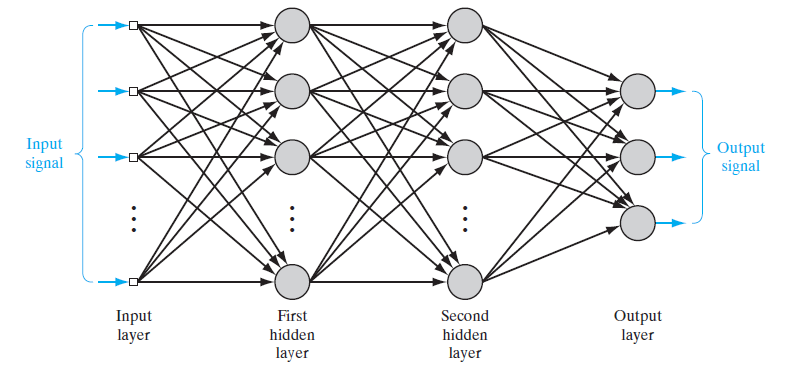
**- Kỹ thuật xử lý ngôn ngữ tự nhiên:** Máy tính sử dụng các thuật toán học sâu để thu thập thông tin chuyên sâu và ý nghĩa từ dữ liệu văn bản và tài liệu. Khả năng xử lý văn bản tự nhiên, do con người tạo ra này có một số trường hợp sử dụng như: tổng đài viên ảo và chatbot tự động, tự động tóm tắt tài liệu hoặc bài viết, lập chỉ mục các cụm từ quan trọng thể hiện cảm xúc, chẳng hạn như những bình luận tích cực và tiêu cực trên mạng xã hội.

**- Công cụ đề xuất:** Các phương pháp này có thể phân tích hành vi của nhiều người dùng khác nhau và giúp họ khám phá các sản phẩm hoặc dịch vụ mới. Ví dụ: nhiều công ty truyền thông và giải trí, chẳng hạn như Netflix, Fox và Peacock, sử dụng mô hình học sâu để đưa ra các video đề xuất cá nhân hóa.

### b. Mạng Neural nhân tạo

Mạng neural nhân tạo (hay còn gọi là mạng thần kinh nhân tạo) Artificial Neural Network (ANN) là một hệ thống các chương trình và cấu trúc dữ liệu mô phỏng cách xử lý thông tin của mạng neural sinh học. Nó được tạo nên từ một số lượng lớn các phần tử neural kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng neural nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu, ...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các neural.

Cấu tạo của neural nhân tạo:



Hình . Cấu tạo của mạng Neural

Nguồn: itnavi.com.vn

- **Tầng Input Layer (tầng vào):** Tầng này nằm bên trái cùng của mạng, thể hiện cho các đầu vào của mạng

- **Tầng Output Layer (tầng ra):** là tầng nằm bên phải cùng và nó thể hiện cho những đầu ra của mạng

- **Tầng First và Second hidden layer (tầng ẩn):** tầng này nằm giữa tầng vào và tầng ra, nó thể hiện cho quá trình suy luận và logic của mạng

Một số loại mạng Neural nhân tạo:

* **Mạng Perceptron:** do Frank Rosenblatt tạo ra năm 1958, là mạng thần kinh đơn giản nhất với một lớp neural
* **Mạng neural truyền thẳng (Feedforward):** hay còn gọi là mạng perceptron nhiều lớp (MLP - Multi-layer perceptrons) gồm một lớp đầu vào, nhiều lớp ẩn và một lớp đầu ra, xử lý dữ liệu một chiều
* **Mạng neural tích chập (CNN):** thường dùng cho nhận dạng hình ảnh, khai thác phép nhân ma trận
* **Mạng thần kinh hồi quy (RNN):** có bộ nhớ lưu thông tin từ bước tính toán trước, dùng trong phân tích dữ liệu chuỗi thời gian như dự đoán thị trường chứng khoán

### c. CNN

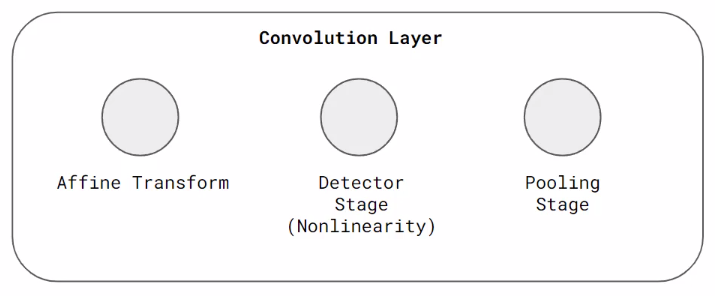
CNN là từ viết tắt của cụm Convolutional Neural Network hay là mạng nơ ron tích chập. Đây là mô hình vô cùng tiên tiến được áp dụng nhiều trong lĩnh vực học sâu Deep learning. Mạng nơ ron này cho phép người dùng xây dựng những hệ thống phân loại và dự đoán với độ chính xác cực cao. Hiện nay, mạng CNN có thể được ứng dụng nhiều trong xử lý ảnh, ngoài ra còn có thể xử lý được tín hiệu âm thanh, ngôn ngữ tự nhiên, video hoặc ảnh y tế 3D.



Hình . Mô hình mạng CNN

Những lớp cơ bản của mạng CNN:

* Convolutional Layer:



Hình . Lớp Layer của mạng CNN

* Một lớp Convolutional Layer bao gồm 3 giai đoạn:

1. Layer thực hiện song song các phép biến đổi tuyến tính

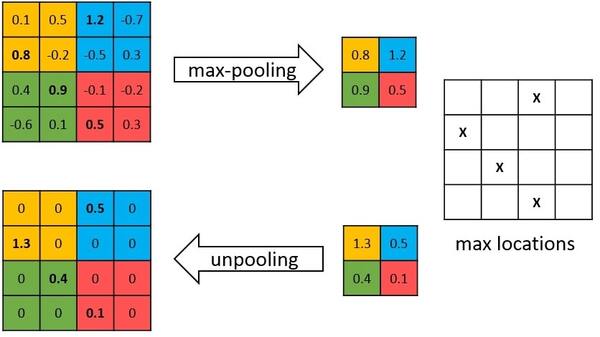
2. Đầu ra của các phép biến đổi này được đưa qua hàm phi tuyến tính

3. Giai đoạn cuối cùng tổng hợp lại thông tin

* Sử dụng các bộ lọc (filters) để quét ma trận văn bản đầu vào. Các bộ lọc này có kích thước (window size) xác định, ví dụ

***h*** x ***d***, trong đó ***h*** là số từ trong một cửa sổ

* Mỗi bộ lọc sẽ thực hiện phép nhân chập (convolution) và tạo ra một bản đồ đặc trưng (feature map). Kích thước của bản đồ đặc trưng phụ thuộc vào kích thước của bộ lọc và các tham số như stride và padding.
* Pooling Layer: Khi ma trận ảnh đầu vào có kích thước quá lớn, các lớp Pooling layer sẽ được đặt vào giữa những lớp Convolutional để làm giảm những parameters. Có thể áp dụng Max pooling hoặc Average pooling để giảm kích thước của bản đồ đặc trưng (feature map). Pooling giúp giảm số lượng tham số và tính toán trong mạng, đồng thời giúp trích xuất các đặc trưng quan trọng nhất.



Hình . Lớp Pooling của mạng CNN

* Fully Connected Layer: Một hoặc nhiều lớp fully connected (dense) được sử dụng để kết hợp các đặc trưng (feature) và thực hiện dự đoán cuối cùng.

## 5. BiLSTM

### a. LSTM

*- Trước khi hiểu về BiLSTM, ta cần hiểu về LSTM:*

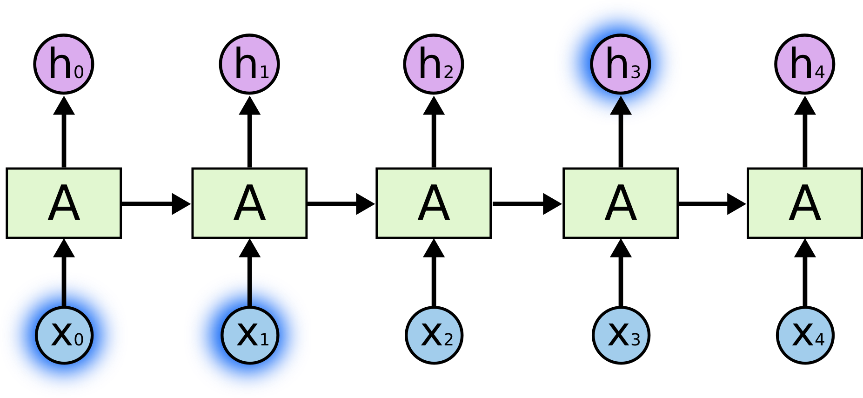
LSTM là một loại mạng neural hồi quy (RNN) được thiết kế để giải quyết vấn đề về gradient biến mất và gradient bùng nổ trong RNN truyền thống. LSTM có cấu trúc gồm các cổng (gates) như input gate, forget gate, và output gate giúp nó kiểm soát luồng thông tin một cách hiệu quả hơn.

Cấu trúc LSTM cho phép nó học và ghi nhớ các phụ thuộc dài hạn trong dữ liệu chuỗi, giúp nó phù hợp cho các tác vụ như dự đoán chuỗi, phân tích ngôn ngữ, và nhận dạng mẫu.

*- Vấn đề phụ thuộc quá dài (Long-Term Dependencies):*

Một điểm nổi bật của RNN chính là ý tưởng kết nối các thông tin phía trước để dự đoán cho hiện tại. Việc này tương tự như ta sử dụng các cảnh trước của bộ phim để hiểu được cảnh hiện thời. Nếu mà RNN có thể làm được việc đó thì chúng sẽ cực kì hữu dụng, tuy nhiên liệu chúng có thể làm được không? Câu trả lời là còn tùy.

Đôi lúc ta chỉ cần xem lại thông tin vừa có thôi là đủ để biết được tình huống hiện tại. **Ví dụ**, ta có câu: “cá thì sống ở dưới nước” thì ta chỉ cần đọc tới “cá thì sống ở dưới” là đủ biết được chữ tiếp theo là “nước” rồi. Trong tình huống này, khoảng cách tới thông tin có được cần để dự đoán là nhỏ, nên RNN hoàn toàn có thể học được.

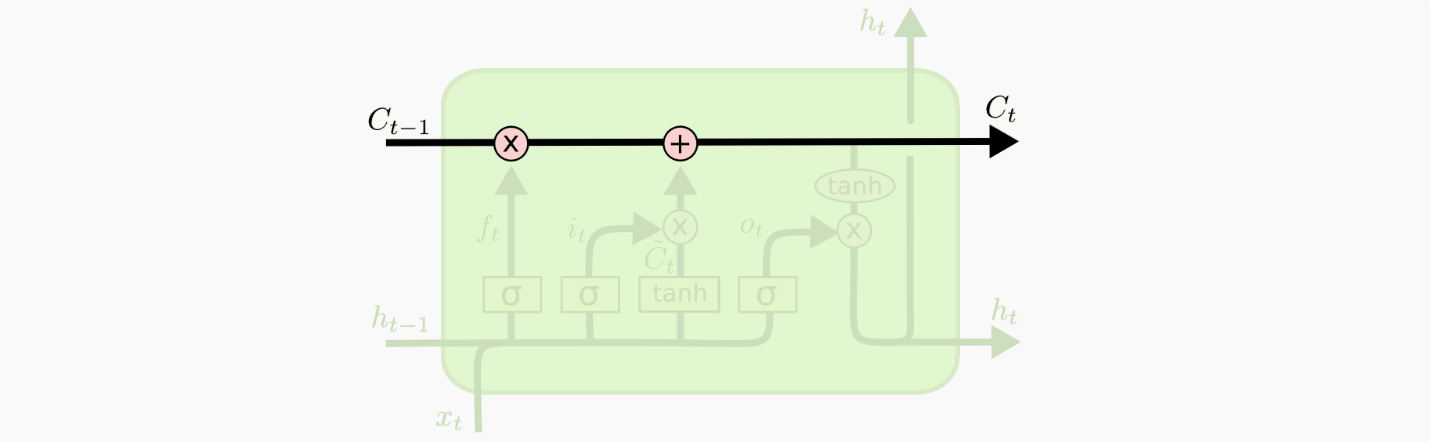


Hình . Vấn đề phụ thuộc quá dài của mạng LSTM

*- Ý tưởng của LSTM:*

Chìa khóa của LSTM là trạng thái tế bào (cell state) - chính đường chạy thông ngang phía trên của sơ đồ hình vẽ.

Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi.



Hình . Trạng thái băng truyền LSTM

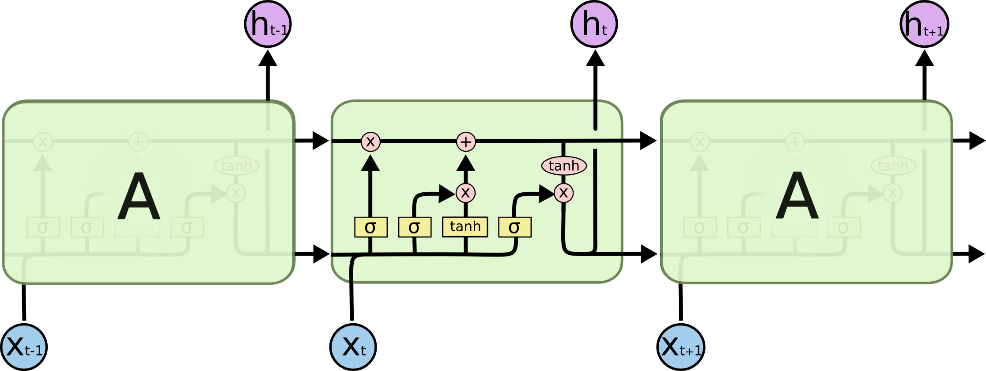
LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate).

Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân. Một LSTM gồm có 3 cổng như vậy để duy trì và điều hành trạng thái của tế bào.



Hình . Cổng nơi để LSTM băng qua

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.



Hình . Mô-đun lặp lại trong một LSTM chứa bốn lớp tương tác

### b. BiLSTM

BiLSTM là một mở rộng của LSTM, cho phép xử lý chuỗi dữ liệu theo cả hai hướng thời gian. Điều này giúp nắm bắt ngữ cảnh tốt hơn và cải thiện hiệu suất của các tác vụ xử lý ngôn ngữ tự nhiên và các tác vụ liên quan đến chuỗi dữ liệu khác.

- BiLSTM mở rộng LSTM bằng cách thêm một chiều xử lý ngược lại. Cụ thể:

* Forward LSTM: Xử lý chuỗi dữ liệu từ đầu đến cuối. Ví dụ, với một câu, forward LSTM sẽ đọc từ từ đầu tiên đến từ cuối cùng.
* Backward LSTM: Xử lý chuỗi dữ liệu từ cuối đến đầu. Ví dụ, backward LSTM sẽ đọc từ từ cuối cùng đến từ đầu tiên.

- Kiến trúc của BiLSTM:

* Forward Pass: Một LSTM xử lý chuỗi dữ liệu theo chiều từ đầu đến cuối.
* Backward Pass: Một LSTM khác xử lý chuỗi dữ liệu theo chiều từ cuối đến đầu.
* Kết Hợp Kết Quả: Tại mỗi thời điểm, kết quả từ forward LSTM và backward LSTM được kết hợp lại (thường bằng cách nối hoặc cộng) để tạo ra một biểu diễn duy nhất.

- Ưu điểm của BiLSTM:

* Nắm Bắt Ngữ Cảnh Toàn Diện: Bằng cách xử lý dữ liệu theo cả hai hướng, BiLSTM có thể nắm bắt thông tin ngữ cảnh từ cả trước và sau mỗi đơn vị trong chuỗi. Điều này đặc biệt quan trọng trong NLP, nơi ngữ cảnh xung quanh một từ hoặc cụm từ có thể thay đổi ý nghĩa của nó.
* Hiệu Quả Cao Trong Nhiều Tác Vụ: BiLSTM đã chứng minh hiệu quả cao trong nhiều tác vụ NLP như gán nhãn từ loại (POS tagging), nhận dạng thực thể có tên (NER), phân tích cảm xúc (sentiment analysis), và dịch máy (machine translation).

- Ví dụ minh hoạ:

Giả sử chúng ta có một câu: "The movie was not bad at all."

* Forward LSTM: Đọc câu từ trái sang phải và có thể học rằng "The movie was not" dẫn đến một dự đoán nào đó.
* Backward LSTM: Đọc câu từ phải sang trái và có thể học rằng "bad at all" mang ý nghĩa tích cực khi nhìn từ ngữ cảnh tổng thể.
* Kết Hợp: Khi kết hợp cả hai chiều, BiLSTM có thể nhận ra rằng toàn bộ câu "The movie was not bad at all" mang ý nghĩa tích cực.

# CHƯƠNG II. XÂY DỰNG MÔ HÌNH

## 1. Chuẩn bị tập dữ liệu, môi trường và các thư viện cần thiết

### a. Môi trường triển khai

Google Colab (Colaboratory) là một công cụ cung cấp môi trường thực thi mã lệnh tương tác miễn phí trên nền tảng đám mây, cho phép ta viết và chạy Python trực tiếp trong trình duyệt. Colab đặc biệt hữu ích cho các tác vụ học máy, khoa học dữ liệu và phân tích dữ liệu. Nó có nhiều tính năng mạnh mẽ, như hỗ trợ GPU và TPU, mà ta có thể sử dụng mà không cần phải cấu hình thêm gì nhiều.

Các Đặc Điểm Chính của Google Colab:

**- Miễn phí và Dễ Sử Dụng:** Ta chỉ cần có tài khoản Google là có thể sử dụng Colab. Môi trường này không yêu cầu cài đặt phần mềm hoặc cấu hình phức tạp.

**- Hỗ trợ GPU và TPU:** Colab cho phép ta sử dụng GPU và TPU miễn phí, giúp tăng tốc các tác vụ đòi hỏi hiệu suất cao như huấn luyện mô hình học sâu.

**- Tích hợp với Google Drive:** Ta có thể dễ dàng lưu trữ và chia sẻ các notebook của mình qua Google Drive.

**- Môi Trường Cài Đặt Sẵn:** Colab đi kèm với nhiều thư viện Python phổ biến đã được cài đặt sẵn, như TensorFlow, Keras, PyTorch, OpenCV, và nhiều thư viện khác.

**- Tích hợp với GitHub:** Ta có thể mở các notebook từ GitHub trực tiếp trong Colab, và ngược lại, dễ dàng chia sẻ và cộng tác với người khác.

**- Giao diện Thân thiện với Người Dùng:** Giao diện tương tự như Jupyter Notebook, dễ dàng sử dụng cho cả người mới bắt đầu và các nhà nghiên cứu chuyên sâu.

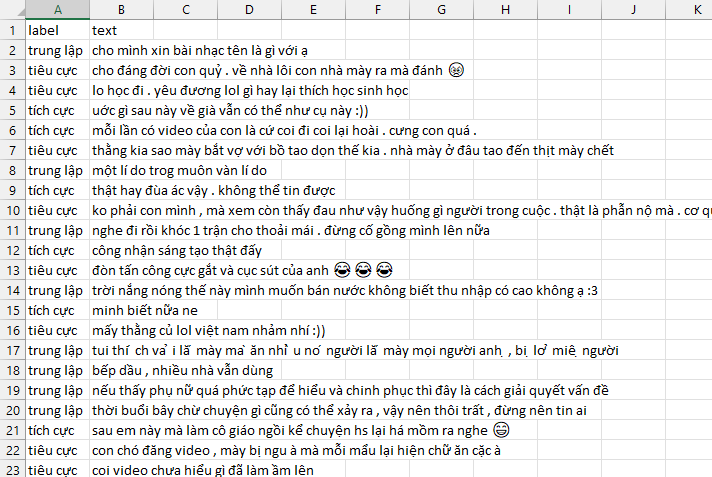
### b. Tập dữ liệu

* **Vietnamese Social Media Emotion – [UIT-VSMEC (version 1.0)]**

Nhận biết cảm xúc là một cách tiếp cận cao hơn hoặc trường hợp đặc biệt của phân tích tình cảm. Trong nhiệm vụ này, kết quả được tạo ra theo 3 nhãn: tích cực tiêu cực hoặc trung lập. Nhận biết cảm xúc đóng một vai trò quan trọng trong việc đo lường giá trị thương hiệu của một sản phẩm bằng cách ghi nhận những cảm xúc cụ thể trong nhận xét của khách hàng.

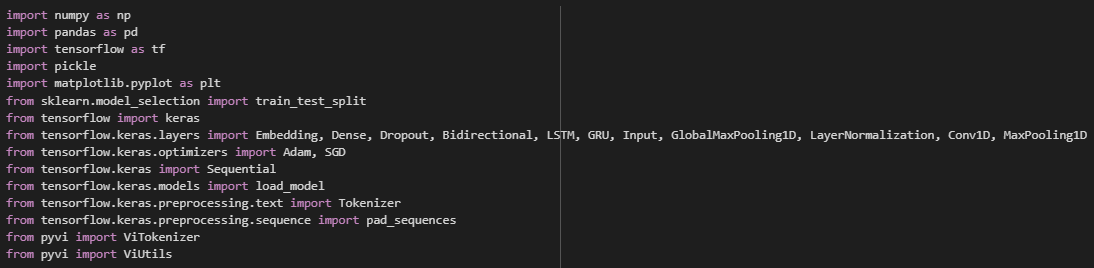
Trong bộ dữ liệu này, Vietnamese Social Media Emotion – (UIT-VSMEC) với khoảng hơn 6.900 câu có chú thích của con người với 3 nhãn, góp phần vào nghiên cứu nhận dạng cảm xúc trong tiếng Việt, vốn là ngôn ngữ sử dụng ít tài nguyên trong Xử lý ngôn ngữ tự nhiên (NLP). Với dữ liệu này giúp chúng ta huấn luyện cho mạng neural của máy.

Mô tả file **\*.CSV** dữ liệu:



Hình . Tập dữ liệu

### c. Các thư viện.



Hình . Các thư viện cần để xây dựng mô hình

- Numpy(Numerical Python): là một thư viện cơ bản cho tính toán khoa học với Python. Nó cung cấp hỗ trợ cho mảng (array), ma trận (matrix) và nhiều hàm toán học khác.

- Pandas: là một thư viện mạnh mẽ cho thao tác và phân tích dữ liệu trong Python. Nó cho phép ta làm việc với các tệp file dữ liệu như CSV hay excel.

- Maplotlib: là một thư viện toàn diện để hiển thị đồ thị, hình ảnh mình hoạ cho dữ liệu của bài toán.

- Pickle: là một module trong Python và được sử dụng để lưu các cấu trúc dữ liệu phức tạp vào file hoặc cơ sở dữ liệu.

- TensorFlow: là một framework mã nguồn mở được phát triển bởi Google dành cho học máy (Machine Learning) và học sâu (Deep Learning). TensorFlow cung cấp các công cụ hỗ trợ cho việc xây dựng, huấn luyện và triển khai các mô hình học máy một cách linh hoạt và hiệu quả trên nhiều nền tảng khác nhau. TensorFlow sở hữu nhiều ưu điểm vượt trội như sau:

+ Hỗ trợ nhiều nền tảng

+ Hiệu suất cao

+ Được sử dụng rộng rãi

+ Hỗ trợ mạnh mẽ cho các mô hình học sâu

+ API linh hoạt

- Keras: là một API cấp cao cho mạng neural và được tích hợp trong TensorFlow, nó có thể giúp xây dựng các mô hình học máy và học sâu một cách hiệu quả bằng việc cung cấp các lớp (Layers) và các API như Sequential API và Preprocessing API

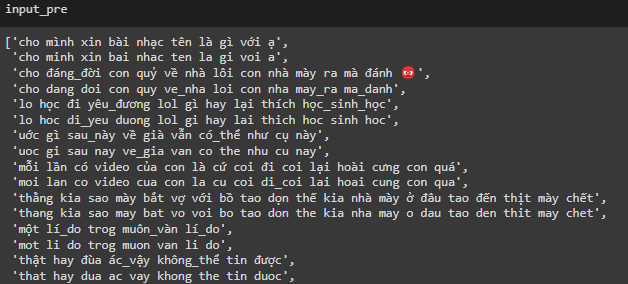
+ Layers:

* Core Layers: các khối xây dựng cơ bản như lớp Dense, lớp Activation, lớp Dropout,v.v
* Convolutional Layers: được sử dụng cho những dữ liệu liên quan tới mô hình Convolutional Neural Network
* Recurrent Layers: được sử dụng cho dữ liệu tuần tự, ví dụ: LSTM, GRU
* Normalization Layers: được sử dụng để chuẩn hoá đầu vào, ví dụ như LayerNormalization

+ Preprocessing API: là một API có chức năng tiền xử lý dữ liệu trước khi đưa vào mạng neural như là Tokenization, padding chuỗi và các tiện ích khác cho dữ liệu văn bản

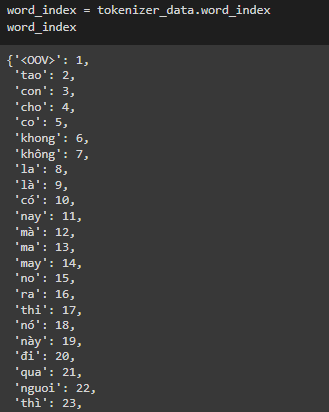
## 2. Tiền xử lý dữ liệu

Trong các bài toán Xử lý ngôn ngữ tự nhiên (NLP) khi xử lý dữ liệu ta thường có một bước quan trọng gọi là Tokenization. Quá trinh này giúp phân chia văn bản thành các đơn vị nhỏ hơn, gọi là token. Token có thể là từ đơn, cụm từ, câu hoặc thậm chí là các ký tự riêng lẻ, tuỳ thuộc vào ứng dụng cụ thể của NLP.

Tokenization thường được sử dụng nhất là Tokenization từ (Word Tokenization): phân chia văn bản thành các từ riêng lẻ. Ví dụ, câu “Bầu trời hôm nay đẹp nhỉ” sẽ được token hoá thành [“Bầu”, “trời”, “hôm”, “nay”, “đẹp”, “nhỉ”]. Tiếng Việt khác tiếng Anh ở chỗ là đối với tiếng Việt một token có thể là 2 từ kết nối với nhau chứ không phải 1 từ như tiếng Anh. Do đó đối với bài toán này trước khi chúng ta thực hiện Tokenization bằng các thư viện của Tensorflow thì chúng ta cần phải Tokenization bằng các thư viện hỗ trợ xử lý ngôn ngữ tiếng Việt như pyvi. Ta sẽ tiến hành nhân đôi 1 câu thành 2 câu, 1 câu sẽ giữ nguyên dấu và 1 câu sẽ loại bỏ dấu và sử dụng cả 2 câu này để tiến hành huấn luyện nhằm đưa ra kết quả tốt nhất.

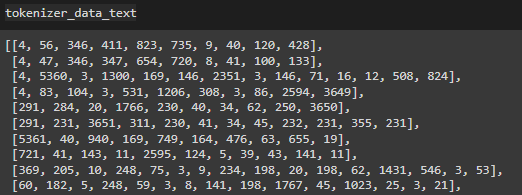
Hình . Tokenization bằng thư viện pyvi

Sau khi thực hiện Tokenization, một bước quan trọng tiếp theo là xây dựng từ điển (Vocabulary Construction). Đây là bước tạo ra một danh sách các từ duy nhất (token) từ toàn bộ tập dữ liệu văn bản. Từ điển này sẽ chứa ít nhất tất cả các token xuất hiện trong tập dữ liệu và thường được sử dụng trong các bước tiếp theo của xử lý ngôn ngữ tự nhiên.



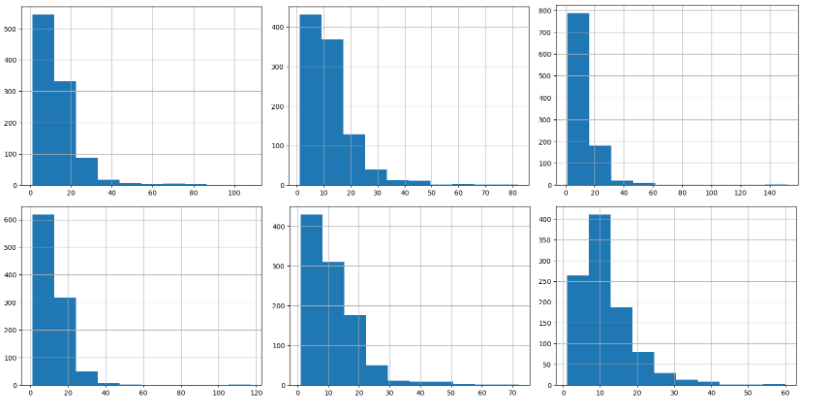
Hình . Bộ từ điển được mapping

Tuy nhiên tập dữ liệu không có độ dài giống nhau nên sau khi xây dựng bộ từ điển chúng ta cần thực hiện padding cho mỗi câu trong tập dữ liệu.

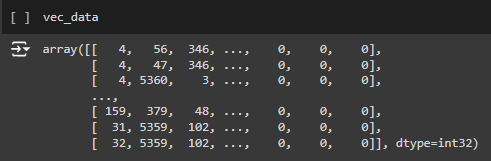


Hình . Độ dài mỗi câu trước khi padding

Padding có nghĩa là thêm các giá trị đặc biệt (thường là 0) vào các chuỗi token để đảm bảo rằng tất cả chuỗi có cùng độ dài. Để padding phù hợp ta cần xác định được maxlength thông qua việc visualization tập dữ liệu. Sau khi visualize ta thấy tập dữ liệu trung bình ở khoảng 500 nên ta lấy maxlen = 512 để padding.

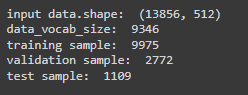


Hình . Visualization tập dữ liệu



Hình . Độ dài mỗi câu sau khi padding

Cuối cùng ta sẽ chia tập dữ liệu thành các tập train, test và validation để tiến hành huấn luyện mô hình.

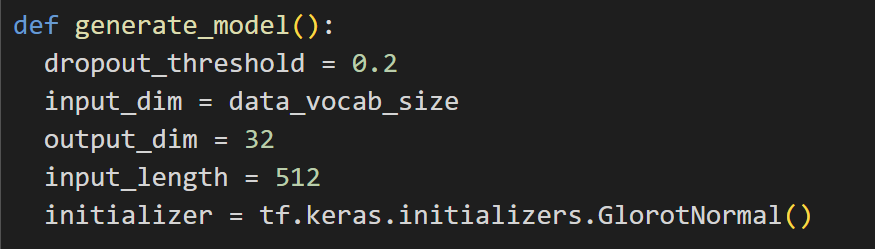


Hình . Chia tập dữ liệu

## 3. Xây dựng mô hình CNN và BiLSTM

Sau khi chuẩn bị tập dữ liệu và tiền xử lý dữ liệu, em sẽ tiến hành xây dựng mô hình để dự đoán. Ở đây em sử dụng mô hình Convolutional Neural Network kết hợp với Bidirectional-LSTM.

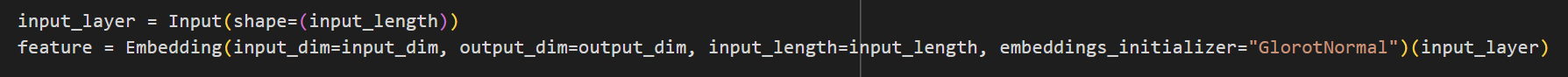
Trước khi xây dựng các lớp model em sẽ định nghĩa các siêu tham số và bộ khởi tạo như sau:



Hình . Siêu tham số và bộ khởi tạo

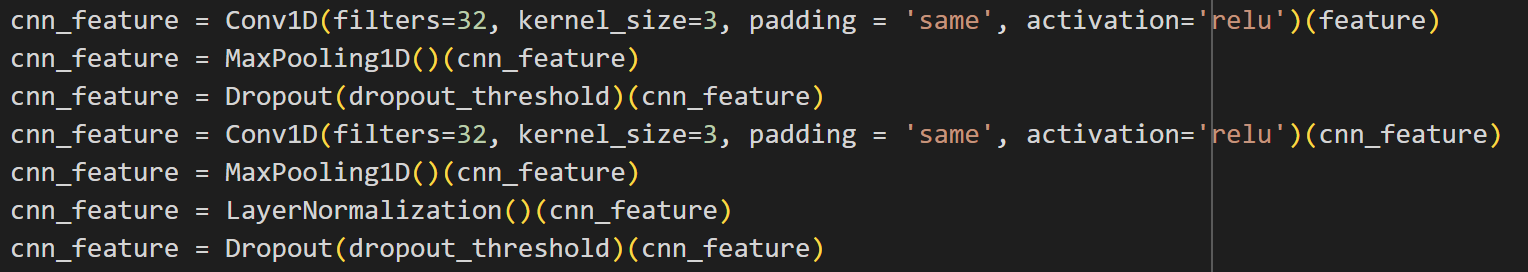
* **Dropout\_threshold**: tỷ lệ dropout để tránh overfitting.
* **Input\_dim** là kích thước từ vựng của dữ liệu đầu vào, em sẽ để nó bằng với data\_vocab\_size để đảm bảo rằng lớp Embedding có thể ánh xạ tất cả các từ trong tập từ vựng một cách chính xác và hiệu quả.
* **Input\_length** sẽ tương đương với max\_length trong quá trình padding tạo ra các véc tơ.
* **Initializer** là bộ khởi tạo trọng số, ở đây em sử dụng Glorot Normal (hay còn gọi là Xavier Normal). Đây là một phương pháp khởi tạo trọng số hiệu quả, giúp duy trì sự ổn định của gradient và cải thiện quá trình học của mạng nơ rơn.

Tiếp theo em tạo ra feature là một lớp Embedding, chuyển đổi các giá trị số nguyên đầu vào (tương ứng với các từ trong từ vựng) thành các vector nhúng có kích thước cố định. Lớp Embedding này giúp biểu diễn từ trong không gian vector có ý nghĩa ngữ nghĩa.



Hình . Khởi tạo Embedding

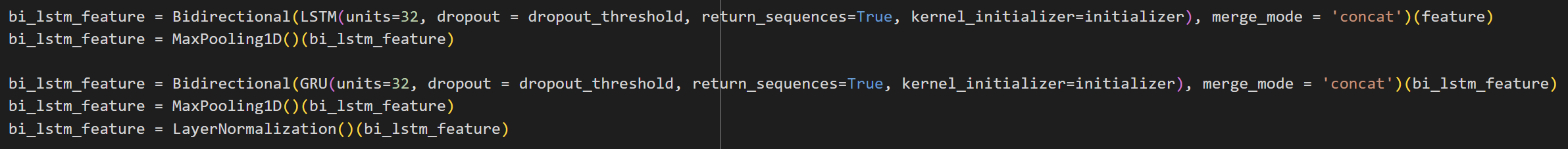
Sau khi có được lớp feature Embedding ở trên em sẽ tiến hành xây dựng các feature từ CNN và Bi-LSTM.



Hình . Tạo các lớp mạng CNN

Ở đây em sẽ sử dụng các lớp tích chậm 1D như Con1D và MaxPooling1D vì đây là bài toán về xử lý các chuỗi văn bản, được biểu diễn dưới dạng các chuỗi các chỉ số nguyên (sau khi nhúng vào không gian véc tơ bởi lớp Embedding). So với các lớp tích chập 2D thì các lớp tích chập 1D có ít tham số hơn (chỉ cần một chiều không gian để áp dụng kernel) và đòi hỏi ít tài nguyên tính toán hơn, làm cho chúng phù hợp cho việc xử lý chuỗi dài.

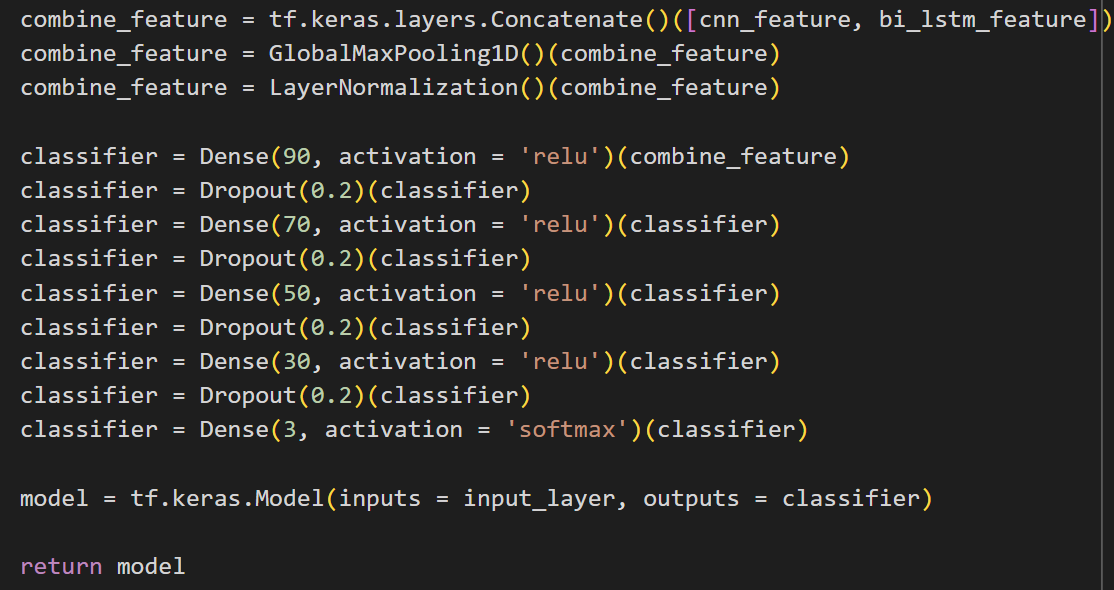
Tiếp theo em tạo thêm các lớp LSTM và GRU có tính chất song song hai chiều (Bidirectional) cùng với các lớp MaxPooling1D và LayerNormalization để xử lý và chuẩn hoá đầu ra



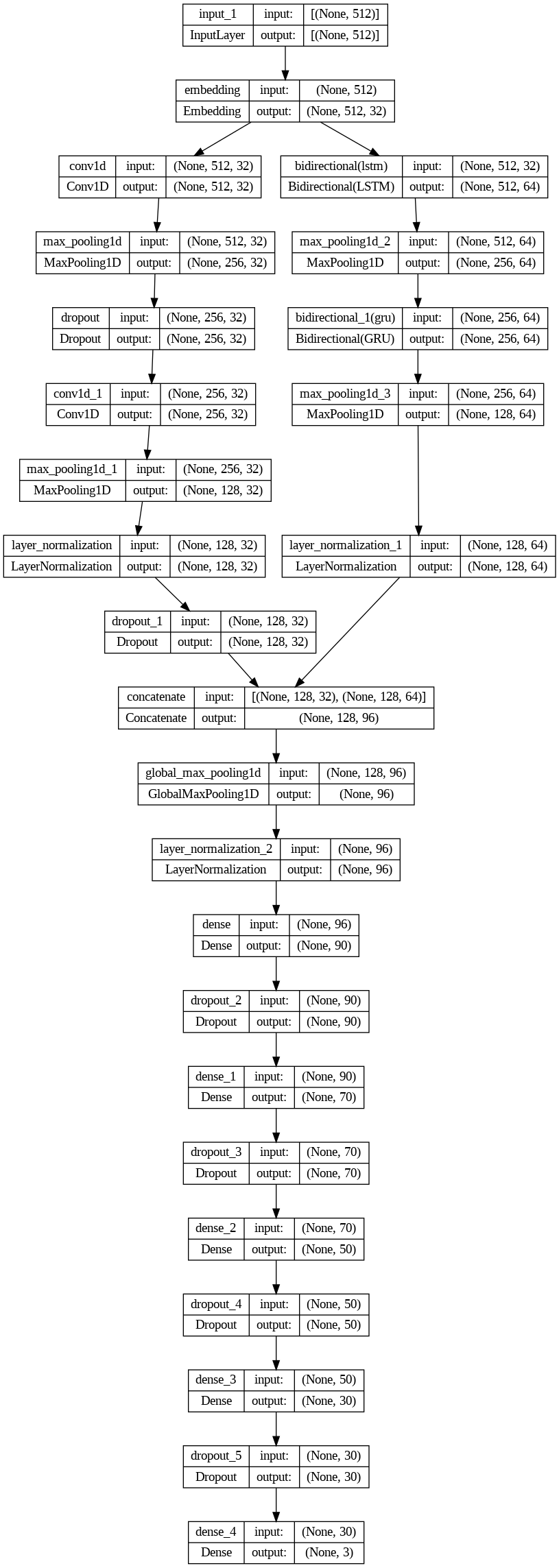
Hình . Tạo lớp mạng BiLSTM

Em sẽ áp dụng các lớp Bi-LSTM để học các đặc trưng từ chuỗi đầu vào theo cả hai hướng từ trái qua phải và từ phải qua trái. Sau đó sử dụng MaxPooling1D để giảm độ dài của chuỗi và giữ lại các đặc trưng quan trọng. Áp dụng thêm lớp GRU Bidirectional để học các đặc trưng từ đầu ra của LSTM và lặp lại MaxPooling1D để tiếp tục giảm độ dài của chuỗi. Cuối cùng là chuẩn hoá đầu ra bằng LayerNormalization để ổn định quá trình huấn luyện.

Sau khi có được hai feature CNN và Bi-LSTM em sẽ tiến hành kết hợp các feature và tạo lớp phân loại.



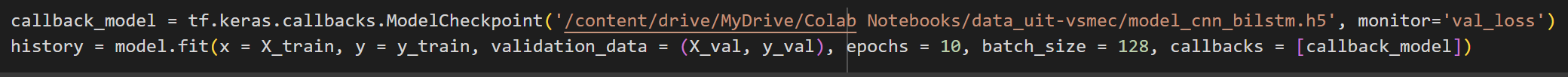
Hình . Nối các lớp đặc trưng



Hình . Kiến trúc mô hình

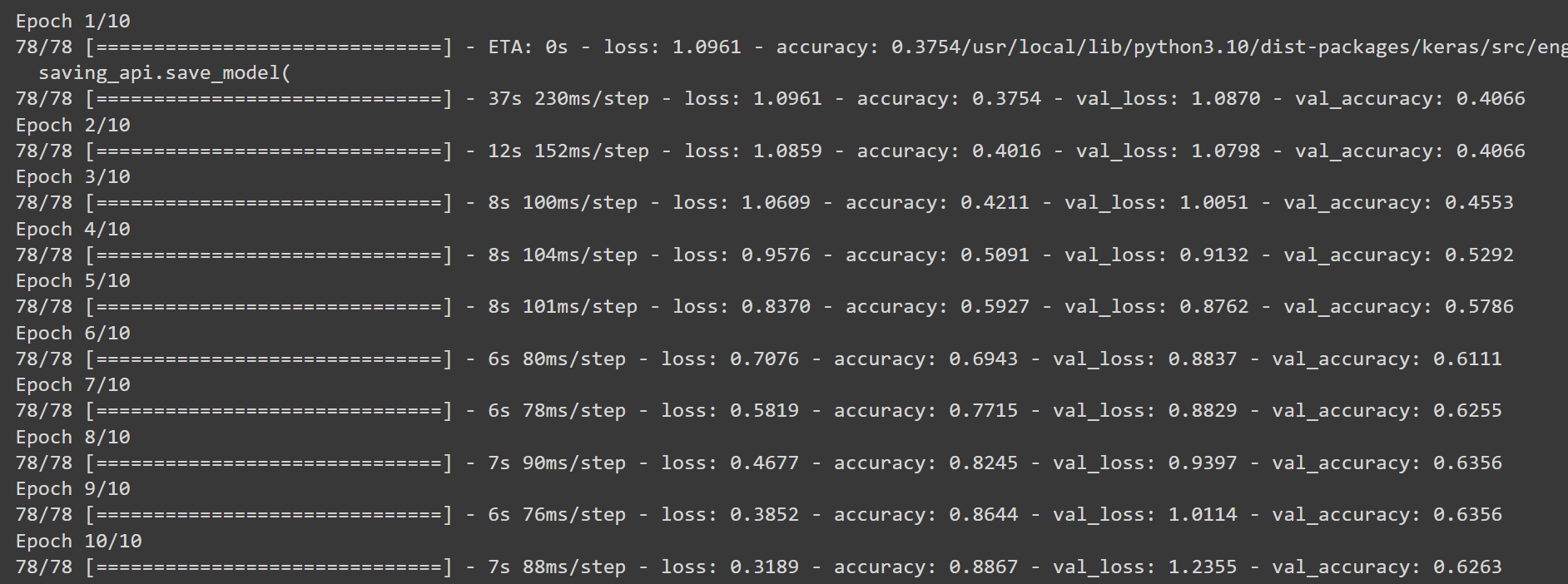
## 4. Huấn luyện và đánh giá mô hình

Ta sẽ tiến hành huấn luyện mô hình, ở đây em train 10 epochs, batch\_size=128. Train xong em sẽ tiến hành lưu model thành file \*.h5 để deploy lên trang web Mạng xã hội.



Hình . Huấn luyện mô hình

Kết quả sau khi train xong 10 epochs



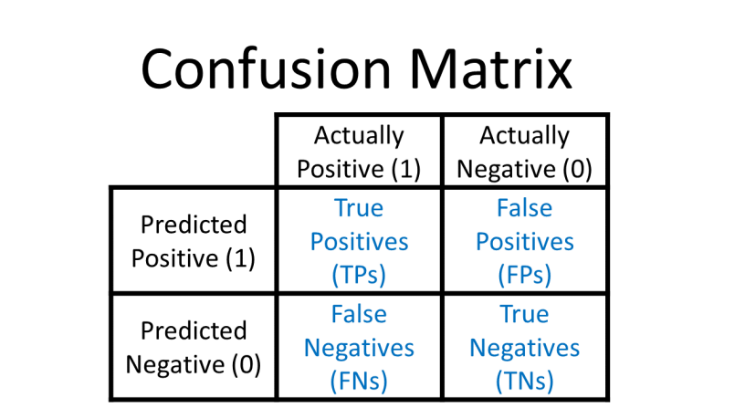
Hình . Kết quả huấn luyện

Tiếp theo sẽ đánh giá mô hình thông qua phép đo Presicion, Recall, F1\_Score, đây là các phép đo thường được sử dụng cho bài toán phân loại.

Khi thực hiện bài toán phân loại, có 4 trường hợp của dự đoán có thể xảy ra:

* **True Positive (TP):** đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Positive (dự đoán đúng)
* **True Negative (TN):** đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Negative (dự đoán đúng)
* **False Positive (FP):** đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Positive (dự đoán sai) – Type I Error
* **False Negative (FN):** đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Negative (dự đoán sai) – Type II Error

Bốn trường hợp trên thường được biểu diễn dưới dạng ma trận hỗn loạn (confusion matrix). Chúng ta có thể tạo ra ma trận này sau khi dự đoán xong trên tập dữ liệu thử nghiệm và rồi phân loại các dự đoán vào một trong bốn trường hợp.



Hình , Confusion Matrix

Ba độ đo chủ yếu để đánh giá một mô hình phân loại là Accuracy, Presicion và Recall

* Accuracy được định nghĩa là tỷ lệ phần trăm dự đoán đúng cho dữ liệu thử nghiệm. Nó có thể được tính toán dễ dàng bằng cách chia số lần dự đoán đúng cho tổng số lần dự đoán

Accurary =

* Precision kiểm tra xem có bao nhiêu kết quả thật là kết quả tích cực trong tổng số các kết quả được dự đoán tích cực

Precision =

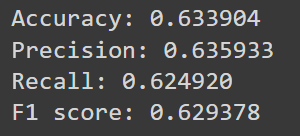
* Recall: kiểm tra các kết quả dự đoán tích cực chính xác trong số các kết quả tích cực

Recall =

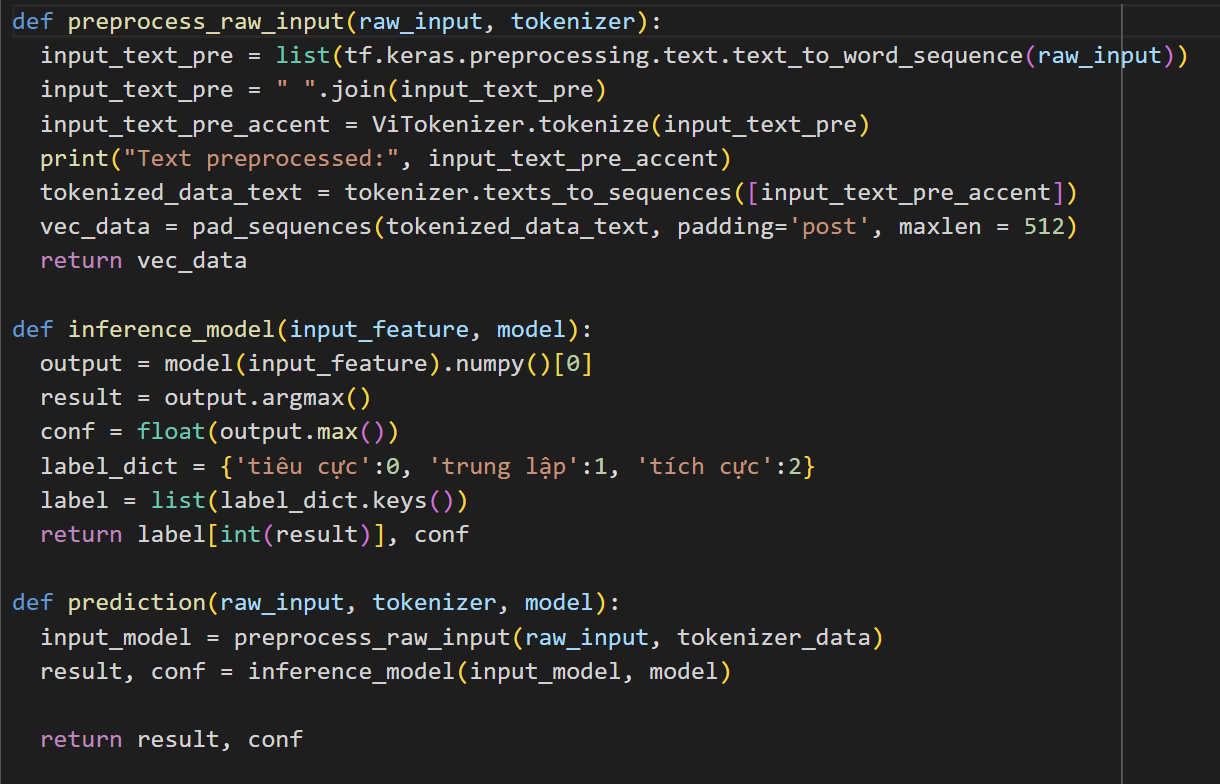
* F1\_score: là trung bình hài hòa của Accuracy và recall, thể hiện sự đóng góp của cả hai. Sự đóng góp phụ thuộc vào giá trị beta, nếu sự đóng của cả 2 là như nhau thì ta có:

=

Đánh giá mô hình qua các độ đo trên ta được tỉ lệ khoảng 63%

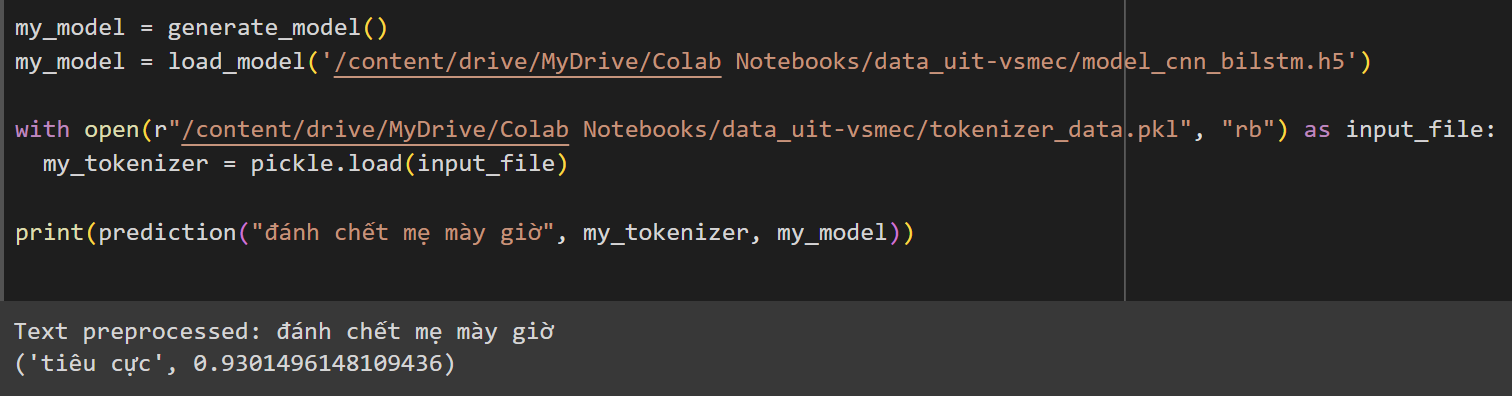


## 5. Dự đoán với mô hình đã huấn luyện



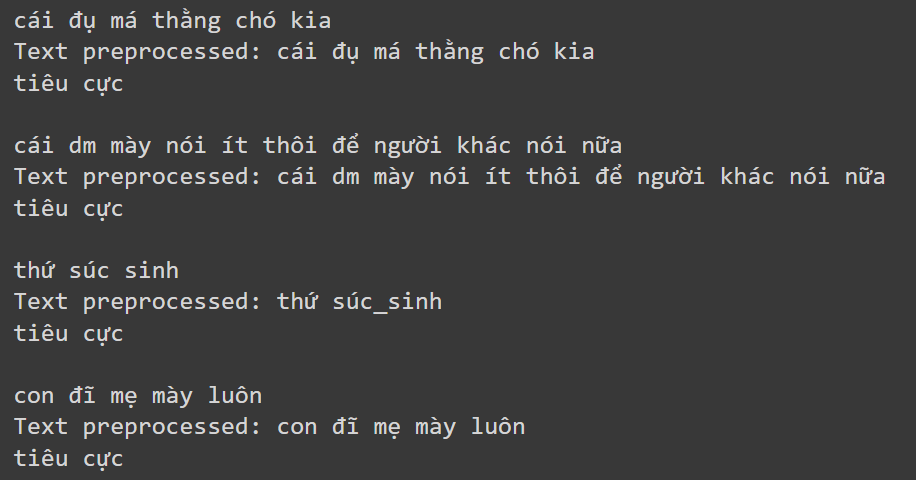
Hình . Các hàm để dự đoán Input mới

Các hàm `preprocess\_raw\_input`, `inference\_model`, `prediction` giúp xử lý đầu vào, thực hiện suy luận với mô hình và dự đoán kết quả. Sau đó sẽ thực hiện load mô hình đã huấn luyện và tokenizer từ tệp và sử dụng để thực hiện dự đoán trên một câu đầu vào.



Hình . Load mô hình và tokenizer để dự đoán Input mới

Có thể thấy kết quả chính xác với tỷ lệ khoảng 93%. Ta thử thêm một số input khác thì kết quả trả ra cũng tương đối chính xác.



Hình . Kết quả Input mới

# CHƯƠNG III. THỰC NGHIỆM MÔ HÌNH VÀO TRANG WEB MẠNG XÃ HỘI

## 1. Sơ lược về trang web Mạng xã hội

### 1.1. Xây dựng

#### a. Giao diện Frontend

Phần giao diện phía người dùng công nghệ em sử dụng chủ yếu là ReactJs. ReactJs là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook. Nó được sử dụng để xây dựng các giao diện người dùng (UI) cho ứng dụng web, giúp cho việc phát triển trở nên dễ dàng, hiệu quả và bảo trì tốt hơn. Các tính năng nổi bật của ReactJs bao gồm khả năng tạo các thành phần giao diện có thể tái sử dụng, quản lý trạng thái ứng dụng và tối ưu hóa hiệu suất thông qua việc sử dụng Virtual DOM.

Một số thành phần cơ bản trong ReactJs

**- Component:**

* Functional Components: Các component được định nghĩa bằng hàm, không có state riêng và sử dụng hooks để quản lý state và side effects.
* Class Components: Các component được định nghĩa bằng class, có state riêng và sử dụng lifecycle methods.

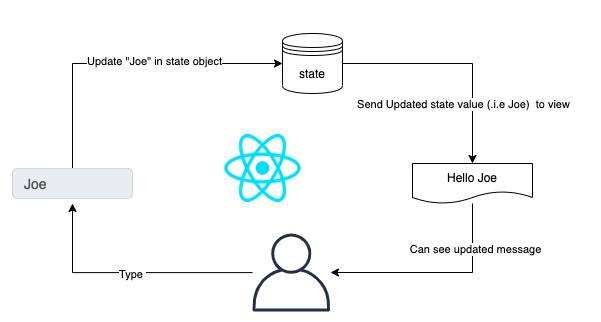
**- JSX (JavaScript XML):** JSX (nói ngắn gọn là JavaScript extension) là một React extension giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản. Và kể từ lúc ReactJS browser hỗ trợ toàn bộ những trình duyệt Web hiện đại, bạn có thể tự tin sử dụng JSX trên bất kỳ trình duyệt nào mà bạn đang làm việc.

**- Props (Properties):**

* Props được sử dụng để gửi dữ liệu đến component.
* Mọi component được coi là một hàm javascript thuần khiết (Pure Function).
* Trong ReactJS, props tương đương với các tham số của hàm javascript thuần khiết.
* Props là bất biến (không thể thay đổi được). Bởi vì điều này được phát triển trong khái niệm về các hàm thuần khiết. Trong các hàm thuần khiết, chúng ta không thể thay đổi dữ liệu của các tham số. Vì vậy, cũng không thể thay đổi dữ liệu của prop trong ReactJS.

**- State:**

* State cũng tương tự như props, nhưng nó là của riêng component và được kiểm soát hoàn toàn bởi chúng và state có thể thay đổi được và mỗi khi state thì đổi thì component đó sẽ được render lại.
* State chỉ tồn tại trong class component hoặc được quản lý bởi hooks trong functional component.



Hình . Vòng đời của State trong ReactJs

**- Lifecycle Methods:**

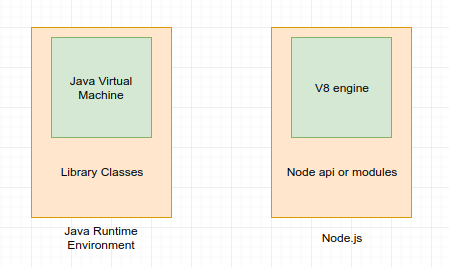
* Các phương thức được gọi tại các giai đoạn khác nhau trong vòng đời của một component.
* Các giai đoạn bao gồm mounting, updating và unmounting.

**- Hook**: Các hàm đặc biệt cho phép sử dụng state và các tính năng khác của React trong functional components. Một số hooks phổ biến là *useState*, *useEffect*,…

* *useState* là một hook trong React được sử dụng để khởi tạo và quản lý trạng thái (state) trong Functional Components. Hook này cho phép ta thêm trạng thái vào Functional Components mà trước đây chỉ có thể được quản lý trong Class Components. useState trả về một mảng với hai phần tử: State variable (biến trạng thái) và State updater function (hàm cập nhật trạng thái).
* *useEffect* là một trong những hooks quan trọng trong React, được sử dụng để thực hiện các tác vụ phụ (side effects) trong Functional Components. Các tác vụ phụ bao gồm lệnh gọi API, thay đổi trạng thái, đăng ký và hủy đăng ký sự kiện, và các tác vụ không thuộc về việc render giao diện người dùng. *useEffect* giúp ta thực hiện các tác vụ này tại các thời điểm cụ thể trong vòng đời của component.

#### b. Xử lý Backend

Về phần xử lý backend thì em sử dụng một trong những framework phổ biến hiện nay là Node.Js. Node.js là một môi trường runtime JavaScript mã nguồn mở, đa nền tảng, cho phép thực thi mã JavaScript trên phía server. Được xây dựng trên nền tảng V8 JavaScript engine của Google Chrome, Node.js được thiết kế để xây dựng các ứng dụng mạng có khả năng mở rộng cao.



Hình . So sánh giữa Node.js và Java

Node.js được biết đến với một số điểm nổi bật như sau

**- Event-Driven và Non-Blocking I/O:**

* Event-Driven: Node.js sử dụng một kiến trúc dựa trên sự kiện, nơi một vòng lặp sự kiện (event loop) duy nhất quản lý tất cả các yêu cầu đến. Điều này giúp Node.js xử lý nhiều yêu cầu đồng thời mà không cần phải tạo ra các luồng (threads) riêng lẻ.
* Non-Blocking I/O: Các hoạt động nhập/xuất (I/O) như đọc/ghi tệp hoặc giao tiếp mạng không chặn luồng thực thi chính. Thay vào đó, chúng được thực hiện không đồng bộ, cho phép xử lý các yêu cầu khác trong khi chờ kết quả.

**- Single-Threaded nhưng Scalable**: Node.js chạy trên một luồng đơn (single-threaded) nhưng có thể xử lý hàng nghìn kết nối đồng thời nhờ vào mô hình non-blocking I/O và event loop. Điều này giúp tăng khả năng mở rộng mà không cần tăng số lượng luồng xử lý.

**- Cross-Platform**: Node.js hoạt động trên nhiều hệ điều hành khác nhau bao gồm Windows, macOS, và các bản phân phối của Linux. Điều này làm cho nó trở thành một công cụ linh hoạt và phổ biến trong phát triển phần mềm.

**- High Performance**: Được xây dựng trên V8 JavaScript engine của Google, Node.js tận dụng hiệu suất cao của V8 để thực thi mã JavaScript nhanh chóng. Hơn nữa, mô hình non-blocking I/O của Node.js giúp tối ưu hóa hiệu suất xử lý các yêu cầu I/O.

**- NPM (Node Package Manager)**: NPM là trình quản lý gói đi kèm với Node.js, cung cấp hàng trăm nghìn module và thư viện sẵn có để sử dụng. NPM giúp quản lý các phụ thuộc của dự án dễ dàng và cho phép chia sẻ mã nguồn với cộng đồng.

**- JSON Support**: Node.js hỗ trợ JSON (JavaScript Object Notation) một cách tự nhiên, làm cho việc trao đổi dữ liệu giữa server và client trở nên dễ dàng và hiệu quả.

**- Real-Time Web Applications**: Node.js đặc biệt phù hợp cho việc xây dựng các ứng dụng web thời gian thực như chat, hệ thống thông báo, và các dịch vụ theo dõi thời gian thực.

**- Microservices and Serverless Architectures**: Node.js hỗ trợ phát triển các kiến trúc microservices và serverless, cho phép xây dựng các ứng dụng với các thành phần nhỏ, độc lập, dễ dàng triển khai và mở rộng.

Ngoài ra một số tính năng của trang web như trò chuyện, thông báo thì em sử dụng thư viện Socket.IO. Nó là một thư viện mạnh mẽ và linh hoạt cho phép các ứng dụng web có khả năng giao tiếp hai chiều (real-time) giữa client và server. Nó được xây dựng trên nền tảng Node.js và có thể dễ dàng tích hợp vào các ứng dụng web để tạo ra các tính năng như chat, thông báo tức thì, cập nhật dữ liệu trực tiếp và nhiều hơn nữa.

Một số tính năng chính của SocketIO

* **Real-time Communication:** Hỗ trợ giao tiếp hai chiều giữa client và server trong thời gian thực.
* **Cross-browser:** Tương thích với nhiều trình duyệt khác nhau, bao gồm cả những trình duyệt cũ.
* **Auto-reconnection:** Tự động kết nối lại nếu kết nối bị mất.
* **Room and Namespace:** Hỗ trợ phân chia các kết nối vào các phòng (room) và không gian tên (namespace) để quản lý giao tiếp hiệu quả.
* **Binary Support:** Hỗ trợ gửi và nhận dữ liệu nhị phân.
* **Scalability:** Dễ dàng mở rộng để hỗ trợ hàng nghìn kết nối đồng thời với sự hỗ trợ từ các adapter như Redis.

Cách hoạt động của SocketIO

* **Client-side library:** Chạy trên trình duyệt và kết nối đến server qua WebSocket hoặc các phương thức dự phòng khác.
* **Server-side library:** Chạy trên Node.js và quản lý kết nối từ client.

Cuối cùng là về cơ sở dữ liệu, ở đây em sử dụng một hệ quản trị cơ sở dữ liệu phi quan hệ (NoSQL) mã nguồn mở đó chính là MongoDB. Nó được phát triển để lưu trữ và truy vấn dữ liệu theo mô hình tài liệu (document-oriented model). MongoDB lưu trữ dữ liệu dưới dạng các tài liệu JSON linh hoạt và cho phép lưu trữ dữ liệu có cấu trúc không đồng nhất.

MongoDB có khả năng mở rộng tốt, cho phép lưu trữ và xử lý các tải trọng công việc lớn và phức tạp. Nó hỗ trợ các tính năng như replica set (bộ sao chép dữ liệu), sharding (phân vùng dữ liệu) và indexing (chỉ mục) để tăng hiệu suất và đảm bảo sẵn sàng cao.

Với MongoDB, việc thay đổi cấu trúc dữ liệu trở nên dễ dàng và linh hoạt, giúp phát triển ứng dụng nhanh chóng và linh hoạt hơn. Nó phù hợp cho các ứng dụng web, mobile, IoT và nhiều ngữ cảnh lưu trữ dữ liệu khác.

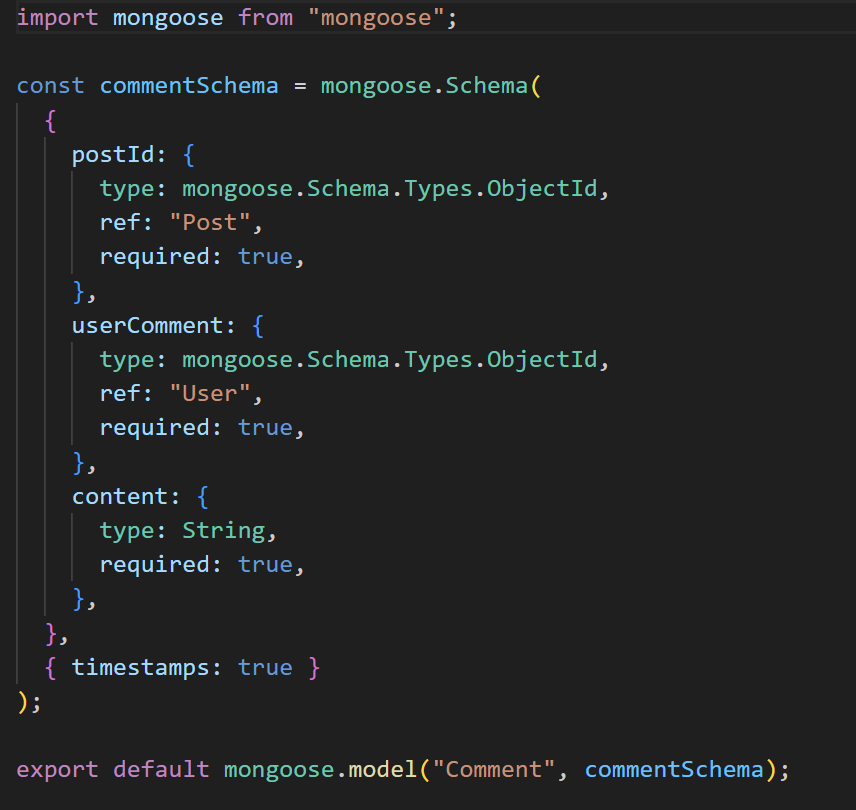
Dưới đây là một số cơ sở dữ liệu được xây dựng bằng MongoDB



Hình . Cơ sở dữ liệu Người dùng



Hình . Cơ sở dữ liệu Bài viết



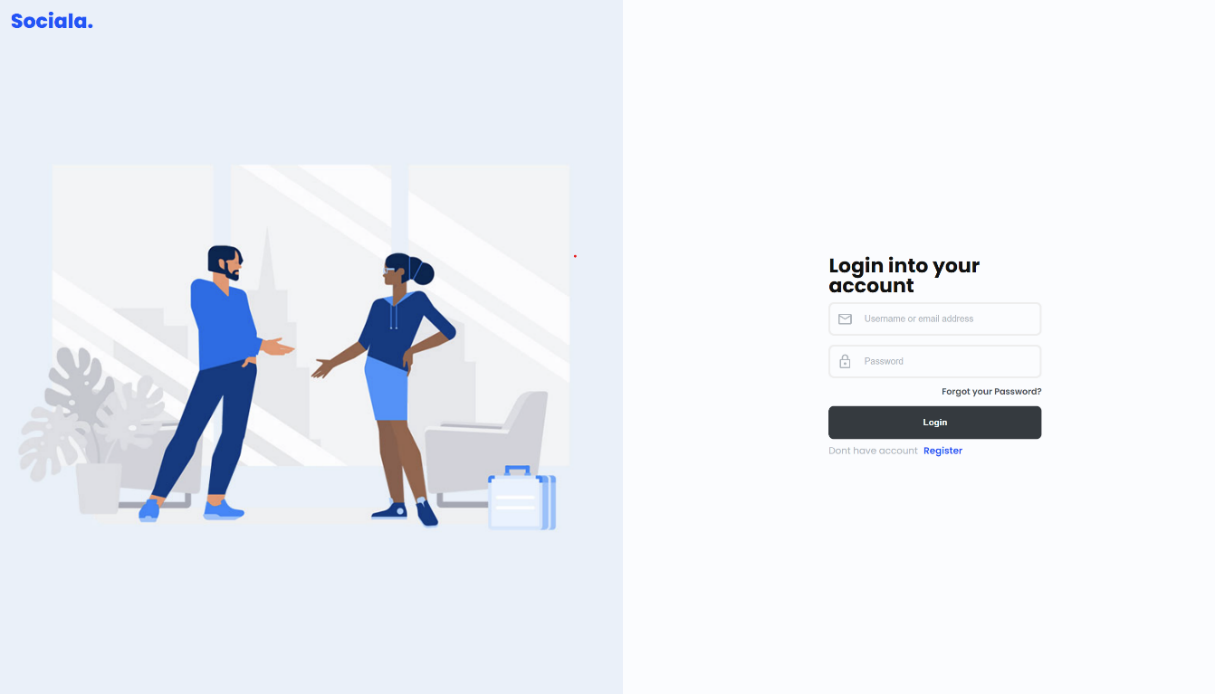
Hình . Cơ sở dữ liệu Bình luận



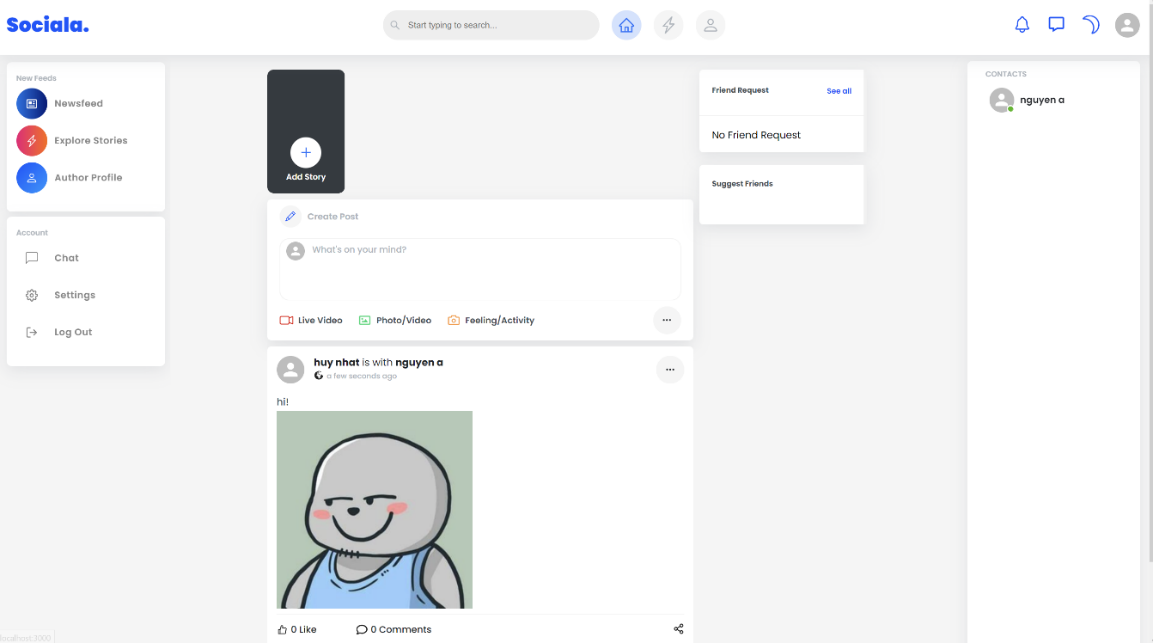
Hình . Cơ sở dữ liệu Trò chuyện

### 1.2. Giao diện

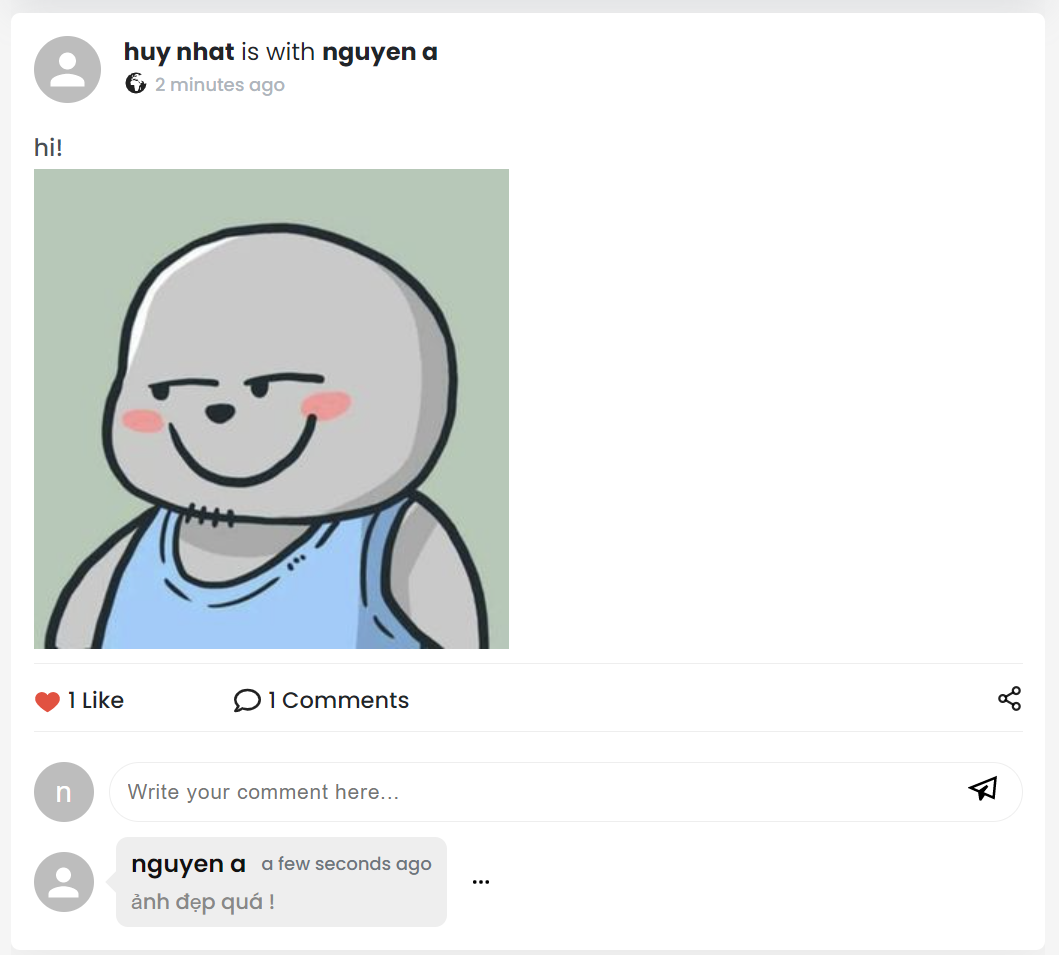
Dưới đây là một số giao diện của trang web Mạng xã hội



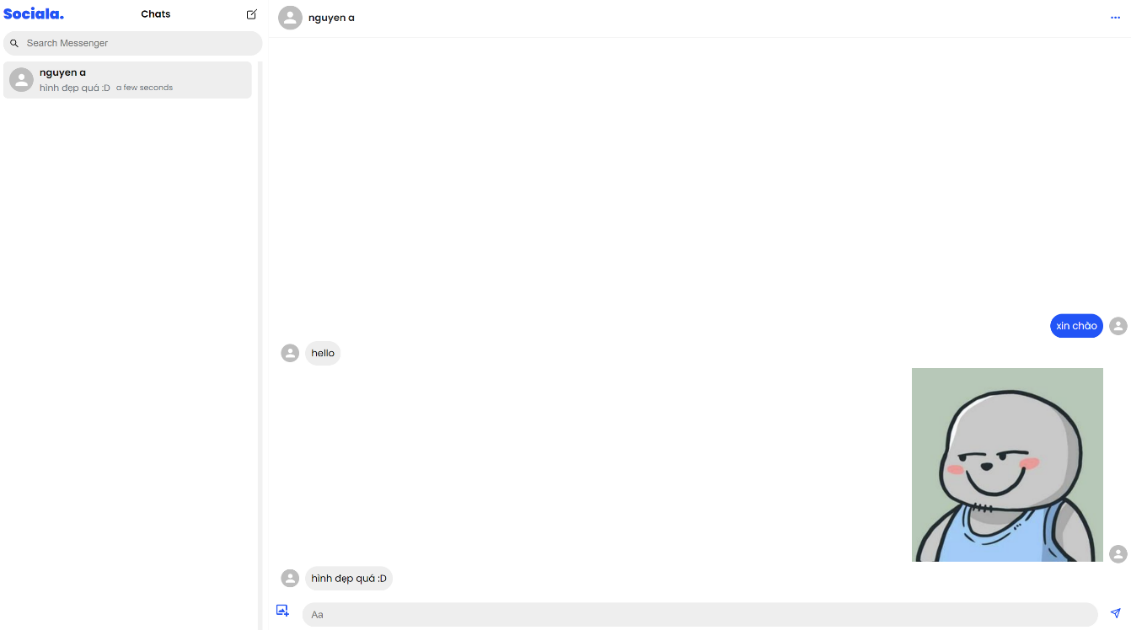
Hình . Giao diện đăng nhập



Hình . Giao diện Trang chủ



Hình . Giao diện Bài viết



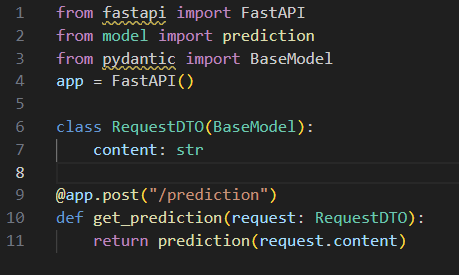
Hình . Giao diện Trò chuyện

## 2. Xây dựng API cho mô hình

Để xây dựng API cho mô hình em sử dụng một framework là FastAPI. Đây là một framework hiện đại và nhanh cho việc xây dựng các API web. Một số tính năng nổi bật của FastAPI như:

* Nhanh: hiệu suất cao khi so sánh với NodeJS và Go
* Code nhanh: tăng tốc độ phát triển tính năng từ 200% tới 300%
* Ít lỗi hơn: Giảm khoảng 40% những lỗi phát sinh bởi con người (nhà phát triển).
* Trực giác tốt hơn: Được các trình soạn thảo hỗ tuyệt vời. Completion mọi nơi. Ít thời gian gỡ lỗi.
* Dễ dàng: Được thiết kế để dễ dàng học và sử dụng. Ít thời gian đọc tài liệu.
* Ngắn: Tối thiểu code bị trùng lặp. Nhiều tính năng được tích hợp khi định nghĩa tham số. Ít lỗi hơn.
* Tăng tốc: Có được sản phẩm cùng với tài liệu (được tự động tạo) có thể tương tác.
* Được dựa trên các tiêu chuẩn: Dựa trên (và hoàn toàn tương thích với) các tiêu chuẩn mở cho APIs : OpenAPI (trước đó được biết đến là Swagger) và JSON Schema.

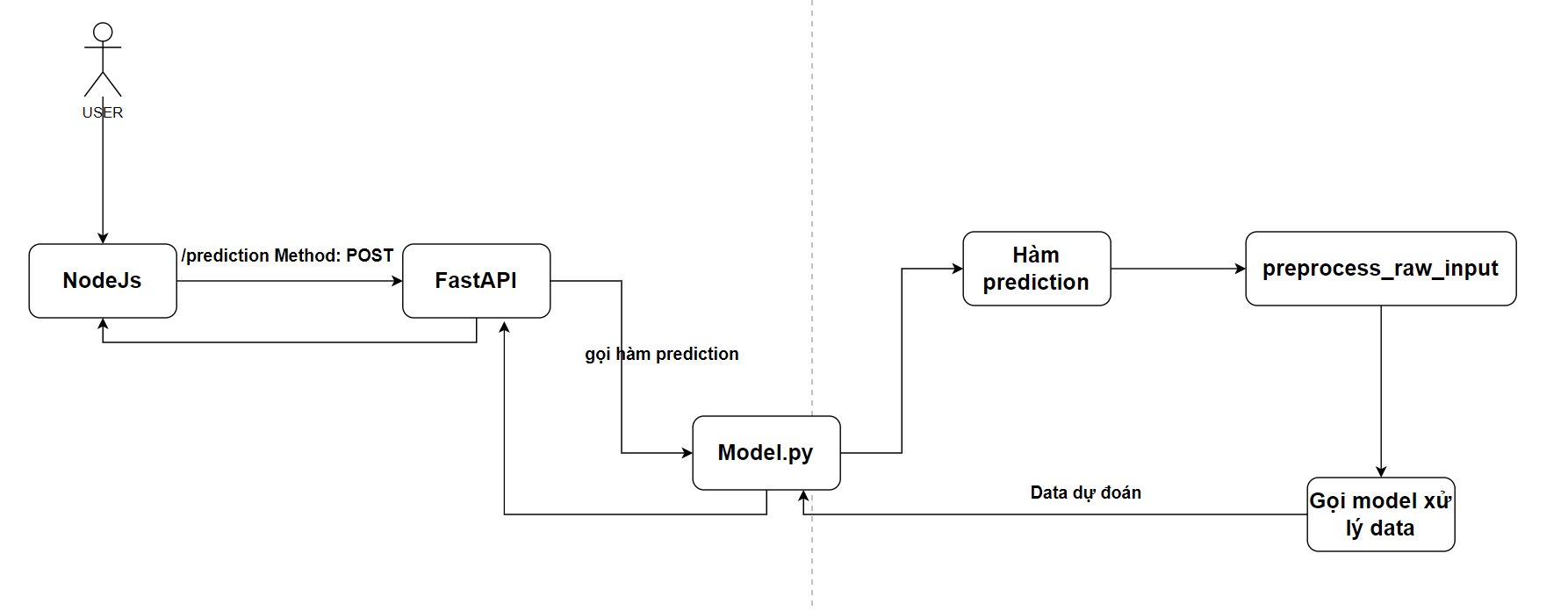
Dưới đây em sẽ sử dụng FastAPI để tạo một API cho model:



Hình. Tạo API từ FastAPI

Cách FastAPI hoạt động sẽ như sau:

* Người dùng tạo bình luận sẽ gửi yêu cầu tới NodeJs
* NodeJs sẽ gọi qua FastAPI với đường dẫn là **/prediction** và **Method: POST**
* FastAPI sẽ gọi hàm prediction từ file model.py
* Từ hàm prediction trong file model.py sẽ gọi hàm preprocess\_raw\_input để tiền xử lý dữ liệu và gọi model xử lý data
* Sau đó sẽ gửi lại data dự đoán cho file model.py để trả về FastAPI và trả về NodeJs.



Hình. Mô hình FastAPI hoạt động

## 3. Thử nghiệm vào trang web Mạng xã hội

# KẾT LUẬN VÀ KIẾN NGHỊ

## I. Kết quả đạt được

## II. Những vấn đề còn tồn đọng

## III. Hướng phát triển tiếp theo của đề tài

# TÀI LIỆU THAM KHẢO

* 1. Rungta, K. “*TensorFlow in 1 Day: Make your own Neural Network*", (2017) (truy cập 03/2024). [Book]
  2. Goyal, P., Pandey, S., & Jain, K. “*Deep Learning for Natural Language Processing: Creating Neural Networks with Python*”, (2018) (truy cập 04/2024). [Book]
  3. Nguyễn Thanh Tuấn. “*Deep Leaning Basic*”, (2020) (truy cập 03/2024). [Book]
  4. *Tài liệu về ReactJs*. Link: [React Introduction](https://www.w3schools.com/REACT/) (truy cập 05/2024). [Online]
  5. *Tài liệu về Node.Js*. Link: [Node.Js Introduction](https://www.w3schools.com/nodejs/)(truy cập 05/2024). [Online]
  6. *Tài liệu về MongoDB*. Link: [Tìm hiểu về MongoDB](https://viblo.asia/p/tim-hieu-ve-mongodb-4P856ajGlY3) (truy cập 05/2024). [Online]
  7. *NLP - Neuro Linguistic Programming*. Link: [NLP là gì?](https://aws.amazon.com/vi/what-is/nlp/) (truy cập 03/2024). [Online]
  8. *Deep Learning – Học sâu*. Link: [Tổng quan về Deep Learning](https://aws.amazon.com/vi/what-is/deep-learning/) (truy cập 03/2024). [Online]
  9. *CNN - Convolutional Neural Network*. Link: [CNN - Mạng tích chập là gì?](https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2) (truy cập 04/2024). [Online]

**[10]** *LSTM – Long Short-Term Memory*. Link: [Bộ nhớ dài-ngắn hạn](https://nttuan8.com/bai-14-long-short-term-memory-lstm/) (truy cập 04/2024). [Online]

**[11]** *FastAPI*. Link: [FastAPI là gì?](https://fastapi.tiangolo.com/vi/) (truy cập 05/2024). [Online]

**[12]** *Evaluation Model with precision – f1\_score - recall*. Link: [Evaluation Model](https://www.miai.vn/2020/06/16/oanh-gia-model-ai-theo-cach-mi-an-lien-chuong-2-precision-recall-va-f-score/) (truy cập 05/2024). [Online]

**[13]** *Tensorflow*. Link: [Tensorflow API Documendation](https://www.tensorflow.org/api_docs/python/tf) (truy cập 04/2024) [Online]

**[14]** *Tokenization*. Link: [Tokenization in NLP](https://www.geeksforgeeks.org/nlp-how-tokenizing-text-sentence-words-works/) (truy cập 04/2024) [Online]