**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: NGHIÊN CỨU, TÌM HIỂU HỆ MẬT PACE VÀ EAC ĐẢM BẢO AN TOÀN THÔNG TIN TRÊN THẺ E-ID**

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

**Sinh viên thực hiện:** VÕ KHẮC MẠNH

**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:** 6151071071  **Họ tên SV:** VÕ KHẮC MẠNH **Khóa:** 61 **Lớp:** CQ. 61.CNTT

**1. Tên đề tài: Nghiên cứu, tìm hiểu hệ mật PACE và EAC đảm bảo an toàn thông tin trên thẻ e-ID**

**2. Mục tiêu**

Tìm hiểu cách thức hoạt động của thẻ danh tính điện tử e-ID, nghiên cứu hai hệ mật PACE và EAC để đảm bảo an toàn thông tin được lưu trữ trên thẻ và xây dựng mô hình mô phỏng quá trình đọc ghi thông tin lên thẻ e-ID.

**3. Nội dung thực hiện**

- Tìm hiểu về thẻ e-ID, cách lưu trữ dữ liệu trên thẻ, các mối nguy hiểm có thể bị tấn công trên thẻ

- Nghiên cứu tìm hiểu về hai hệ mật Password Authenticated Connection Establishment (PACE) và Extended Access Control (EAC)

- Nghiên cứu bài toán giả định gPACE-DH

- Đưa ra mô hình xác thực e-ID

- Xây dựng mô hình Iot mô phỏng quá trình đọc ghi dữ liệu lên thẻ e-ID

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

- Thiết bị sử dụng: Module ESP 8266, Module RC522, Thẻ RFID Mirafe 1KB.

- Công cụ lập trình: Arduino IDE, Visual Studio Code

- Ngôn ngữ lập trình: C/C++, HTML, CSS, Javascripts.

**5. Các kết quả chính dự kiến**

- Đưa ra được mô hình xác thực e-ID dựa trên hai hai cơ chế PACE và EAC

- Xây dựng mô hình mô phỏng quá trình đọc ghi dữ liệu lên thẻ e-ID

**6. Kế hoạch thực hiện**

- Tuần 1-2: Tìm và chọn đề tài

- Tuần 3: Xác định đề tài

- Tuần 4-5: Tìm hiểu và đọc các tài liệu liên quan đến hộ chiếu điển tử (biometric passport), thẻ danh tính điện tử (e-ID), các tài liệu nghiên cứu về các thuật toán.

- Tuần 6 đến 10: Nghiên cứu về các mô hình các bài toán về hai hệ mật PACE và EAC, xây dựng mô hình mô phỏng

- 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

Quãng thời gian thực làm đồ án vừa qua thật sự là một hành trình đáng trân trọng, mà từ đó em rút ra được nhiều bài học quý giá, đồng thời tích lũy những kinh nghiệm giá trị cho tương lai sự nghiệp của mình. Điều quan trọng không chỉ là sự cố gắng của bản thân, mà còn là sự hỗ trợ và đồng hành đầy ý nghĩa từ mọi người xung quanh.

Để hoàn thành đề tài này trước hết em xin gửi đến quý thầy, cô **Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông Vận tải tại Thành phố Hồ Chí Minh** lời cảm ơn chân thành vì đã truyền đạt cho em những kiến thức không chỉ từ sách vở, mà còn những kinh nghiệm quý giá từ cuộc sống trong khoảng thời gian học tập tại trường.

Trong quá trình làm đồ án, kiến thức của em vẫn còn hạn chế và nhiều bờ ngỡ. Do vậy, những thiếu sót là điều chắc chắn không thể tránh khỏi, em rất mong nhận được những ý kiến đóng góp của thầy cô và các bạn để kiến thức của em được hoàn thiện hơn.

Sau cùng, em xin kính chúc Quý Thầy Cô trong **Bộ môn Công nghệ thông tin** lời chúc sức khỏe, luôn 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**

**Võ Khắc Mạnh**

# 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](#_Toc169090189)

[LỜI CẢM ƠN iii](#_Toc169090190)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN iv](#_Toc169090191)

[MỤC LỤC v](#_Toc169090192)

[DANH MỤC HÌNH ẢNH viii](#_Toc169090193)

[DANH MỤC BẢNG x](#_Toc169090194)

[DANH MỤC BẢNG VIẾT TẮT xi](#_Toc169090195)

[TỔNG QUAN 1](#_Toc169090196)

[1. Lý do chọn đề tài 1](#_Toc169090197)

[2. Mục tiêu và nhiệm vụ của đồ án 2](#_Toc169090198)

[3. Bố cục đồ án 2](#_Toc169090199)

[CHƯƠNG I. CƠ SỞ LÝ THUYẾT 3](#_Toc169090200)

[1. Tổng quan về AI – Machine Learning 3](#_Toc169090201)

[a. AI là gì? 3](#_Toc169090202)

[b. Machine Learning là gì? 3](#_Toc169090203)

[2. Xử lý ngôn ngữ tự nhiên 5](#_Toc169090204)

[a. Định nghĩa 5](#_Toc169090205)

[b. Các bài toán và ứng dụng 6](#_Toc169090206)

[3. Sentiment Analysis 7](#_Toc169090207)

[a. Định nghĩa 7](#_Toc169090208)

[b. Các ứng dụng của Sentiment Analysis 8](#_Toc169090209)

[c. Sentiment Analysis hoạt động như thế nào? 8](#_Toc169090210)

[4. Deep Learning 11](#_Toc169090211)

[a. Deep Learning là gì? 11](#_Toc169090212)

[b. Mạng Neural nhân tạo 12](#_Toc169090213)

[c. CNN 13](#_Toc169090214)

[5. BiLSTM 15](#_Toc169090215)

[a. LSTM 15](#_Toc169090216)

[b. BiLSTM 17](#_Toc169090217)

[CHƯƠNG II. XÂY DỰNG MÔ HÌNH 19](#_Toc169090218)

[1. Chuẩn bị tập dữ liệu, môi trường và các thư viện cần thiết 19](#_Toc169090219)

[a. Môi trường triển khai 19](#_Toc169090220)

[b. Tập dữ liệu 19](#_Toc169090221)

[c. Các thư viện. 20](#_Toc169090222)

[2. Tiền xử lý dữ liệu 22](#_Toc169090223)

[3. Xây dựng mô hình CNN và BiLSTM 24](#_Toc169090224)

[4. Huấn luyện và đánh giá mô hình 28](#_Toc169090225)

[5. Dự đoán với mô hình đã huấn luyện 30](#_Toc169090226)

[CHƯƠNG III. THỰC NGHIỆM MÔ HÌNH VÀO TRANG WEB MẠNG XÃ HỘI 31](#_Toc169090227)

[1. Sơ lược về trang web Mạng xã hội 31](#_Toc169090228)

[1.1. Xây dựng 31](#_Toc169090229)

[a. Giao diện Frontend 31](#_Toc169090230)

[b. Xử lý Backend 33](#_Toc169090231)

[1.2. Giao diện 38](#_Toc169090232)

[2. Xây dựng API cho mô hình 40](#_Toc169090233)

[3. Thử nghiệm vào trang web Mạng xã hội 41](#_Toc169090234)

[KẾT LUẬN VÀ KIẾN NGHỊ 42](#_Toc169090235)

[1. Kết quả đạt được 42](#_Toc169090236)

[2. Những vấn đề còn tồn đọng 42](#_Toc169090237)

[3. Hướng phát triển tiếp theo của đề tài 42](#_Toc169090238)

[TÀI LIỆU THAM KHẢO 43](#_Toc169090239)

# DANH MỤC HÌNH ẢNH

[Hình 1. Sentiment Analysis 8](#_Toc169090093)

[Hình 2. Feature Engineering 9](#_Toc169090094)

[Hình 3. Sử dụng Supervised Learning 9](#_Toc169090095)

[Hình 4. Sử dụng Deep Learning 10](#_Toc169090096)

[Hình 5. Sử dụng BERT Model 11](#_Toc169090097)

[Hình 6. Cấu tạo của mạng Neural 13](#_Toc169090098)

[Hình 7. Mô hình mạng CNN 14](#_Toc169090099)

[Hình 8. Lớp Layer của mạng CNN 14](#_Toc169090100)

[Hình 9. Lớp Pooling của mạng CNN 15](#_Toc169090101)

[Hình 10. Vấn đề phụ thuộc quá dài của mạng LSTM 16](#_Toc169090102)

[Hình 11. Trạng thái băng truyền LSTM 16](#_Toc169090103)

[Hình 12. Cổng nơi để LSTM băng qua 17](#_Toc169090104)

[Hình 13. Mô-đun lặp lại trong một LSTM chứa bốn lớp tương tác 17](#_Toc169090105)

[Hình 14. Tập dữ liệu 20](#_Toc169090106)

[Hình 15. Các thư viện cần để xây dựng mô hình 20](#_Toc169090107)

[Hình 16. Tokenization bằng thư viện pyvi 22](#_Toc169090108)

[Hình 17. Bộ từ điển được mapping 23](#_Toc169090109)

[Hình 18. Độ dài mỗi câu trước khi padding 23](#_Toc169090110)

[Hình 19. Visualization tập dữ liệu 24](#_Toc169090111)

[Hình 20. Độ dài mỗi câu sau khi padding 24](#_Toc169090112)

[Hình 21. Chia tập dữ liệu 24](#_Toc169090113)

[Hình 22. Siêu tham số và bộ khởi tạo 25](#_Toc169090114)

[Hình 23. Khởi tạo Embedding 25](#_Toc169090115)

[Hình 24. Tạo các lớp mạng CNN 25](#_Toc169090116)

[Hình 25. Tạo lớp mạng BiLSTM 26](#_Toc169090117)

[Hình 26. Nối các lớp đặc trưng 26](#_Toc169090118)

[Hình 27. Kiến trúc mô hình 27](#_Toc169090119)

[Hình 28. Huấn luyện mô hình 28](#_Toc169090120)

[Hình 29. Kết quả huấn luyện 28](#_Toc169090121)

[Hình 30. Confusion Matrix 29](#_Toc169090122)

[Hình 31. Các hàm để dự đoán Input mới 30](#_Toc169090123)

[Hình 32. Load mô hình và tokenizer để dự đoán Input mới 30](#_Toc169090124)

[Hình 33. Kết quả Input mới 30](#_Toc169090125)

[Hình 34. Vòng đời của State trong ReactJs 32](#_Toc169090126)

[Hình 35. So sánh giữa Node.js và Java 33](#_Toc169090127)

[Hình 36. Cơ sở dữ liệu Người dùng 36](#_Toc169090128)

[Hình 37. Cơ sở dữ liệu Bài viết 36](#_Toc169090129)

[Hình 38. Cơ sở dữ liệu Bình luận 37](#_Toc169090130)

[Hình 39. Cơ sở dữ liệu Trò chuyện 37](#_Toc169090131)

[Hình 40. Giao diện đăng nhập 38](#_Toc169090132)

[Hình 41. Giao diện Trang chủ 38](#_Toc169090133)

[Hình 42. Giao diện Bài viết 39](#_Toc169090134)

[Hình 43. Giao diện Trò chuyện 39](#_Toc169090135)

[Hình 44. Tạo FastAPI 40](#_Toc169090136)

[Hình 45. Mô hình FastAPI hoạt động 41](#_Toc169090137)

# DANH MỤC BẢNG

# DANH MỤC BẢNG VIẾT TẮT

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

# TỔNG QUAN

## 1. Lý do chọn đề tài

Trong thời đại kỷ nguyên số, định danh điện tử là yếu tố quan trọng để thực hiện các giao dịch điện tử và xây dựng chính phủ số. Chính phủ đang tập trung phát triển dịch vụ hành chính công trực tuyến để thúc đẩy cuộc cách mạng công nghiệp lần thứ tư. Do đó, bảo mật thông tin trở thành mối quan tâm hàng đầu của người dân và doanh nghiệp khi thực hiện giao dịch điện tử, bao gồm cả dịch vụ công trực tuyến và các giao dịch khác trên mạng.

Bộ Thông tin và Truyền thông đã xây dựng một nghị định về định danh điện tử, đang lấy ý kiến từ công chúng và các bộ ngành liên quan. Nghị định này sẽ tạo cơ sở pháp lý cho việc cung cấp và sử dụng dịch vụ xác thực và định danh điện tử, đảm bảo an ninh thông tin trong các giao dịch điện tử. Điều này cũng giúp nâng cao hiệu quả của các cổng dịch vụ công quốc gia và hệ thống thông tin điện tử, góp phần hoàn thành mục tiêu xây dựng chính phủ điện tử.

Phát triển thẻ căn cước công dân có gắn chip là bước đầu cho việc phát triển thẻ danh tính điện tử ở Việt Nam. Tuy nhiên, còn nhiều hạn chế và vấn đề cần giải quyết trong việc phát triển xác thực điện tử, đặc biệt là đảm bảo an toàn khi sử dụng thẻ danh tính điện tử.

Vì vậy, em chọn đề tài “Nghiên cứu, tìm hiểu hệ mật PACE và EAC đảm bảo an toàn thông tin trên thẻ e-ID” để tập trung vào định danh và xác thực điện tử, đảm bảo truy cập an toàn và dễ dàng vào các dịch vụ công trực tuyến, giúp cơ quan tổ chức và doanh nghiệp bảo vệ hệ thống máy chủ web khỏi các cuộc tấn công, đồng thời thúc đẩy quá trình chuyển đổi số của nước ta.

## 2. Mục tiêu và nhiệm vụ của đồ án

Đưa ra được mô hình xác thực e-ID dựa trên hai hệ mật PACE và EAC, xây dựng mô hình mô phỏng quá trình đọc ghi dữ liệu trên thẻ e-ID

## 3. Bố cục đồ án

TỔNG QUAN

CHƯƠNG 1: TỔNG QUAN VỀ THẺ DANH TÍNH ĐIỆN TỬ

CHƯƠNG 2: DỮ LIỆU TRÊN THẺ DANH TÍNH ĐIỆN TỬ

CHƯƠNG 3: HỆ MẬT ĐẢM BẢO AN TOÀN THÔNG TIN TRÊN THẺ DANH TÍNH ĐIỆN TỬ

XÂY DỰNG HỆ THỐNG MÔ PHỎNG ĐỌC GHI DỮ LIỆU LÊN THẺ RFID

KẾT LUẬN VÀ KIẾN NGHỊ

TÀI LIỆU THAM KHẢO

# CHƯƠNG I. TỔNG QUAN VỀ THẺ DANH TÍNH ĐIỆN TỬ

## 1.1. Giới thiệu về thẻ danh tính điện tử

### 1.1.1. Lịch sử ra đời

Thẻ danh tính điện tử (e-ID) đã phát triển từ nhu cầu về các phương pháp nhận dạng và xác thực danh tính đáng tin cậy và hiệu quả trong bối cảnh số hóa ngày càng tăng. Dưới đây là một cái nhìn tổng quan về lịch sử ra đời và phát triển của thẻ danh tính điện tử:

* **Thời Kỳ Đầu - Thẻ nhựa và mã vạch (1970s-1980s):**
* 1970s: Các thẻ nhựa với mã vạch hoặc dải từ được sử dụng rộng rãi cho các mục đích nhận dạng cá nhân, như thẻ nhân viên hoặc thẻ ngân hàng.
* 1980s: Hệ thống nhận dạng bằng thẻ thông minh (smart card) bắt đầu phát triển với các thẻ chứa chip vi mạch (integrated circuit chip), cho phép lưu trữ và xử lý dữ liệu trên thẻ.
* **Phát Triển Thẻ Thông Minh (1990s):**
* 1991: Các tiêu chuẩn ISO/IEC 7816 cho thẻ thông minh được ban hành, tạo ra cơ sở cho việc phát triển các thẻ với khả năng lưu trữ và bảo mật cao hơn.
* 1997: Thẻ thông minh bắt đầu được sử dụng rộng rãi ở châu Âu, đặc biệt là trong các ứng dụng tài chính và viễn thông.
* 1997: Thẻ thông minh bắt đầu được sử dụng rộng rãi ở châu Âu, đặc biệt là trong các ứng dụng tài chính và viễn thông.
* **Sự Ra Đời của Thẻ Danh Tính Điện Tử (2000s)**
* 2000s: Sự xuất hiện của thẻ danh tính điện tử, với việc các quốc gia như Estonia, Bỉ và Malaysia bắt đầu phát hành thẻ e-ID cho công dân của mình. Những thẻ này không chỉ lưu trữ thông tin cá nhân mà còn tích hợp chữ ký số và các chức năng bảo mật khác.
* 2002: Estonia phát hành thẻ căn cước điện tử, một trong những thẻ e-ID tiên tiến nhất vào thời điểm đó, cho phép công dân truy cập các dịch vụ công trực tuyến và thực hiện các giao dịch điện tử an toàn.
* 2004: Bỉ giới thiệu thẻ danh tính điện tử, được sử dụng cho các dịch vụ công và bảo mật cao.
* **Phổ Biến Toàn Cầu (2010s - Nay):**
* 2010s: Nhiều quốc gia trên toàn thế giới bắt đầu triển khai thẻ danh tính điện tử như một phần của chiến lược chính phủ điện tử và an ninh mạng. Thẻ e-ID được sử dụng cho nhiều mục đích khác nhau, từ nhận dạng cá nhân, xác thực giao dịch trực tuyến, đến truy cập các dịch vụ công cộng và tư nhân.
* 2010: Malaysia phát hành thẻ MyKad, một thẻ đa chức năng tích hợp nhiều dịch vụ.
* 2013: Ấn Độ bắt đầu triển khai chương trình thẻ căn cước điện tử Aadhaar, một trong những hệ thống nhận dạng lớn nhất thế giới.
* 2020s: Với sự phát triển của công nghệ blockchain và sinh trắc học, thẻ e-ID ngày càng trở nên tiên tiến hơn, cung cấp các biện pháp bảo mật mới và tích hợp nhiều dịch vụ số.
* **Tại Việt Nam:**
* 2021: Việt Nam bắt đầu triển khai thẻ căn cước công dân gắn chip, với mục tiêu cung cấp một phương tiện nhận dạng an toàn và hiệu quả, tích hợp nhiều tiện ích và dịch vụ công trực tuyến.
* Nghị định về định danh điện tử: Bộ Thông tin và Truyền thông đã xây dựng các quy định và nghị định liên quan đến định danh điện tử, tạo cơ sở pháp lý cho việc sử dụng và phát triển thẻ danh tính điện tử

Tuy nhiên, việc triển khai và sử dụng thẻ danh tính điện tử cũng đối mặt với nhiều thách thức, bao gồm vấn đề bảo mật thông tin cá nhân và quyền riêng tư. Do đó, việc phát triển và quản lý thẻ danh tính điện tử đòi hỏi sự cân nhắc kỹ lưỡng về các vấn đề pháp lý và chính sách, cũng như việc áp dụng các biện pháp bảo mật hiệu quả để bảo vệ thông tin cá nhân của người dùng.

CMND cũ.

### 1.1.2. Khái niệm về danh tính điện tử (e-Identity)

Danh tính điện tử hay danh tính số (e-Identity) là tập hợp các thông tin điện tử phục vụ việc xác định duy nhất một cá nhân, tổ chức.

Theo đó, danh tính điện tử của công dân Việt Nam bao gồm: Số định danh cá nhân; họ, tên đệm và tên; ngày, tháng, năm sinh; giới tính; ảnh chân dung và vân tay.Danh tính điện tử của người nước ngoài bao gồm: Số hộ chiếu hoặc số giấy tờ có giá trị đi lại quốc tế; họ, tên đệm và tên; ngày, tháng, năm sinh; giới tính; quốc tịch; ảnh chân dung và vân tay (nếu có).



Hình 1.1: Danh tính điện tử của công dân Việt Nam

Việc định danh, xác thực điện tử gồm nhiều hình thức như: Tài khoản/mật khẩu; thiết bị lưu mã (token); điện thoại di động và thẻ SIM. Định danh, xác thực điện tử cung cấp vai trò đặc biệt quan trọng trong Hệ sinh thái số để đảm bảo truy cập an toàn và dễ dàng tới các dịch vụ công trực tuyến.

Nhận dạng điện tử (" eID ") là một giải pháp kỹ thuật số để chứng minh danh tính của công dân hoặc tổ chức. Chúng có thể được sử dụng để xem để truy cập các lợi ích hoặc dịch vụ do cơ quan chính phủ, ngân hàng hoặc các công ty khác cung cấp cho thanh toán di động, v.v. Ngoài xác thực và đăng nhập trực tuyến, nhiều dịch vụ nhận dạng điện tử cũng cung cấp cho người dùng tùy chọn để ký tài liệu điện tử bằng kỹ thuật số chữ ký.

**1.1.3. Giới thiệu sơ lược về thẻ danh tính điện tử (e-ID)**

Thẻ danh tính điện tử (eID card) là một loại thẻ nhận dạng kỹ thuật số được trang bị vi mạch điện tử, giúp xác thực danh tính của người sở hữu một cách an toàn và tiện lợi. Thẻ danh tính điện tử thường được chính phủ cấp phát và có thể được sử dụng cho nhiều mục đích khác nhau, bao gồm xác thực danh tính, truy cập các dịch vụ trực tuyến của chính phủ và giao dịch điện tử.

1. **Cấu tạo của thẻ danh tính điện tử bao gồm:**

* Vỏ thẻ
* Chất liệu: Thẻ thường được làm từ nhựa bền, chẳng hạn như polycarbonate hoặc PVC, để đảm bảo độ bền và khả năng chống chịu mài mòn.
* Kích thước: Theo tiêu chuẩn quốc tế ISO/IEC 7810 ID-1 (85.60 mm × 53.98 mm).
* Vi mạch điện tử (chip)
* Vi mạch: Chứa các dữ liệu cá nhân và sinh trắc học, cũng như các chứng chỉ số và khóa mã hóa.
* Giao diện: Có thể giao tiếp với đầu đọc thẻ qua tiếp xúc (contact) hoặc không tiếp xúc (contactless). Một số thẻ có cả hai giao diện.
* Ống dây Antenna
* Antenna: Được nhúng trong thẻ để hỗ trợ giao tiếp không tiếp xúc bằng cách truyền dữ liệu giữa thẻ và đầu đọc qua công nghệ RFID hoặc NFC.
* Các thành phần bảo mật
* Chứng chỉ số: Được lưu trữ trong chip để xác thực danh tính và ký số.
* Khóa mã hóa: Được sử dụng để bảo vệ dữ liệu trên thẻ và trong quá trình giao tiếp với đầu đọc.
* Dữ liệu in trên thẻ
* Thông tin cá nhân: Tên, ngày sinh, quốc tịch, số thẻ, ngày phát hành và ngày hết hạn.
* Ảnh chân dung: Ảnh khuôn mặt của người sở hữu thẻ.
* Chữ ký số: Chữ ký của người sở hữu (nếu có).
* Mã vạch hoặc mã QR: Chứa thông tin bổ sung hoặc liên kết đến cơ sở dữ liệu trực tuyến.
* Dữ liệu lưu trữ trong chip
* Thông tin cơ bản: Tên, ngày sinh, quốc tịch, số thẻ, ngày phát hành và ngày hết hạn.
* Dữ liệu sinh trắc học: Ảnh khuôn mặt, dấu vân tay, và có thể bao gồm quét mống mắt.
* Chứng chỉ số và khóa mã hóa: Được sử dụng cho các giao dịch điện tử và bảo mật thông tin.
* Các tính năng bảo mật vật lí
* In chống giả: Các kỹ thuật in ấn tiên tiến như in siêu nhỏ, in hologram, và các yếu tố bảo mật khác để ngăn chặn việc làm giả.
* Lớp phủ bảo vệ: Một lớp bảo vệ trong suốt để chống trầy xước và bảo vệ các thông tin in trên thẻ.

A close-up of a computer id card

Description automatically generated

Mặt trước

A text on a white background

Description automatically generated

Mặt sau

1. **Thẻ phải dựa trên các tiêu chuẩn như:**

* ISO/IEC 14443-1: Đặc điểm vật lý
* ISO/IEC 14443-2: Giao diện RF (Radio Frequency Interface)
* ISO/IEC 14443-3: Giao thức chống xung đột và xử lý lệnh (Initialization and Anti-collision)
* ISO/IEC 14443-4: Giao thức truyền thông (Transmission Protocol)

Thẻ CCCD có gắn chip điện tử, còn gọi là thẻ căn cước điện tử (e-ID) là một loại giấy tờ tùy thân của công dân Việt Nam, có thể đóng vai trò thiết bị nhận diện, xác thực danh tính và chìa khóa truy cập thông tin công dân trong hệ thống cơ sở dữ liệu quốc gia. Nó có giá trị chứng minh về căn cước công dân và cho phép người dùng tiếp cận nhiều dịch vụ vốn đòi hỏi hàng loạt giấy tờ khác nhau.

Thẻ CCCD gắn chip điện tử về cơ bản cũng giống như thẻ CCCD mã vạch. Tuy nhiên, trên thẻ không có các dòng trạng thái thể hiện mã vạch mà nó sẽ thay thế bằng chip điện tử dung lượng lớn. Thẻ CCCD gắn chip sẽ lưu trữ những đặc điểm nhận dạng bằng hình ảnh, vân tay và sinh trắc học.



Hộ chiếu điện tử (ePassport) hay hộ chiếu sinh trắc học (Biometric passport) có gắn chip chứa thông tin cá nhân và các đặc điểm nhận dạng được cài phía trong bìa để vừa tăng thêm giá trị cho hộ chiếu (nhờ độ bảo mật cao hơn), giảm nguy cơ làm giả hoặc sửa đổi thông tin, vừa tương thích với các thiết bị kiểm tra hiện đại được trang bị ở các sân bay quốc tế.

**1.2. Đặc điểm kĩ thuật của và các tiêu chuẩn của thẻ e-ID**

**1.2.1. Đặc điểm vật lí của MRTD**

MRTD (Machine Readable Travel Document) là thuật ngữ dùng để chỉ các loại giấy tờ thông hành có thể đọc được bằng máy, chẳng hạn như hộ chiếu, thẻ căn cước. Các tài liệu này được thiết kế theo các tiêu chuẩn quốc tế, cho phép máy móc đọc và trích xuất dữ liệu một cách tự động và nhanh chóng, đảm bảo tính chính xác và hiệu quả trong quá trình kiểm tra và xác thực thông tin cá nhân tại các cửa khẩu biên giới, sân bay và các điểm kiểm soát khác.

Các quốc gia và tổ chức phát hành có thể lựa chọn các tài liệu được sử dụng để sản xuất các thẻ của họ.

* ***Kích thước và hình dạng:***
* Hộ chiếu tiêu chuẩn: Kích thước là 125mm x 88mm, giống với kích thước của sổ tay nhỏ.
* Thẻ căn cước: Kích thước là 85.60mm x 53.98mm, tương đương với kích thước của thẻ tín dụng.
* Chất liệu:
* Hộ chiếu: Bìa hộ chiếu thường được làm từ chất liệu giấy tổng hợp hoặc vật liệu nhựa có độ bền cao, giúp bảo vệ các trang bên trong. Các trang bên trong làm từ giấy đặc biệt chống giả mạo.
* Thẻ căn cước: Thường được làm từ nhựa polycarbonate hoặc vật liệu tổng hợp khác, đảm bảo độ bền và khả năng chống mài mòn.
* Dải quang học (Machine Readable Zone - MRZ):
* MRZ: Là một dải chứa các ký tự có thể đọc bằng máy, thường nằm ở cuối trang nhận diện của hộ chiếu hoặc mặt sau của thẻ căn cước. MRZ chứa thông tin như tên, số hộ chiếu, quốc tịch, ngày sinh và ngày hết hạn.
* Định dạng MRZ: Được tiêu chuẩn hóa theo ICAO với hai hoặc ba dòng ký tự OCR-B, tùy thuộc vào loại tài liệu.
* Chip điện tử (đối với e-Passport và e-ID):
* Chip RFID: Nhiều MRTD hiện đại, như e-Passport và e-ID, tích hợp một chip RFID chứa dữ liệu sinh trắc học và thông tin cá nhân. Chip này giúp tăng cường bảo mật và chống làm giả.
* Dữ liệu sinh trắc học: Bao gồm ảnh khuôn mặt, dấu vân tay và/hoặc mống mắt, được lưu trữ trong chip để xác thực danh tính của người sở hữu.
* Yếu tố bảo mật:
* Hình mờ (Watermark): Các hộ chiếu và thẻ căn cước thường có hình mờ hoặc các yếu tố bảo mật nhìn thấy dưới ánh sáng đặc biệt để chống giả mạo.
* In siêu mịn (Microprinting): Sử dụng in siêu mịn với các dòng chữ nhỏ khó sao chép.
* Mực chống giả mạo: Sử dụng mực UV, mực thay đổi màu sắc hoặc mực phát quang để tăng cường bảo mật.
* Hologram và lớp phủ bảo mật: Sử dụng các hình ảnh ba chiều và lớp phủ bảo mật đặc biệt để bảo vệ thông tin và chống giả mạo.
* Thông tin cá nhân và ảnh:
* Trang dữ liệu cá nhân: Chứa thông tin cá nhân như tên, quốc tịch, ngày sinh, và ảnh chân dung của người sở hữu.
* Ảnh chân dung: Được chụp theo tiêu chuẩn ICAO để đảm bảo khả năng nhận diện và xác thực sinh trắc học.

A diagram of a card

Description automatically generated

*Ba kích thước của MRTD bao gồm MRP (kích thước TD3) với các vị trí được đề xuất cho tính năng xác minh tài liệu hỗ trợ của máy.*

**1.2.2. Các thông số ký thuật của thẻ**

**A diagram of a rectangular object with numbers and lines

Description automatically generated with medium confidence**

*Minh họa các kích thước ISO / IEC 7810 tính bằng milimét*

* ID-1

Định dạng ID-1 chỉ định kích thước 85,60 x 53,98 mm (3+3 ⁄ 8 in × 2+1 ⁄ 8 in) và các góc tròn có bán kính 2,88–3,48 mm (khoảng1 ⁄ 8 in). Nó thường được sử dụng cho các loại thẻ thanh toán (thẻ ATM, thẻ tín dụng, thẻ ghi nợ, v.v.). Ngày nay nó cũng được sử dụng cho giấy phép lái xe và chứng minh thư cá nhân ở nhiều quốc gia, thẻ hệ thống thu phí tự động cho phương tiện giao thông công cộng, trong thẻ khách hàng thân thiết bán lẻ.

* ID-2

Định dạng ID-2 chỉ định kích thước 105 x 74 mm (4+1 ⁄ 8 in × 2+15 ⁄ 16 in). Kích thước này là định dạng A7. Ví dụ: định dạng ID-2 được sử dụng cho thị thực.

* ID-3

ID-3 chỉ định kích thước 125 x 88 mm (4+15 ⁄ 16 in × 3+7 ⁄ 16 in). Kích thước này làđịnh dạng B7. Định dạng này thường được sử dụng cho các cuốn sổ hộ chiếu.

* ID-000

ID-000 chỉ định kích thước 25 x 15 mm (1 in × 9 ⁄ 16 in), với một góc hơi vát (3 mm hoặc 1 ⁄ 8 in). Kích thước này được sử dụng cho định dạng " mini-SIM " của mô-đun nhận dạng thuê bao.

1.6. Cách thức hoạt động

Thẻ danh tính điện tử hoạt động dựa trên nhận dạng tần số vô tuyến (RFID). Đây là một công nghệ nhận dạng truyền dữ liệu thông qua việc sử dụng liên lạc không dây bằng sóng vô tuyến. RFID công nghệ lần đầu tiên được sử dụng trong Thế chiến II (WWII) cho hệ thống nhận dạng, bạn hay thù (IFF). RFID đã được sử dụng cho mục đích xác định một đối tượng hoặc một người. Cách truyền dữ liệu được thực hiện giữa một đầu đọc và một chip điện tử gắn vào một đồ vật hoặc một người. RFID hệ thống cho Hộ chiếu được Kích hoạt (E-passport) bao gồm một con chip, một đầu đọc, một ăng-ten và Cơ sở hạ tầng khóa công khai (PKI).

Ở cấp độ cơ bản mỗi thẻ hoạt động theo một cách:

* Dữ liệu được lưu trữ trong vi mạch của thẻ chờ được đọc
* Ăng-ten của thẻ nhận được năng lượng từ ăng-ten của đầu đọc RFID.
* Sử dụng nguồn điện từ pin bên trong của nó hoặc năng lượng thu được từ trường điện từ của đầu đọc, thẻ sẽ gửi các sóng vô tuyến trở lại đầu đọc.

Người đọc thu nhận các sóng vô tuyến của thẻ và giải thích các tần số dữ liệu là có ý nghĩa.

**1.2.3. Mức độ an toàn của thẻ danh tính điện tử**

Mức độ bảo đảm của danh tính điện tử (được phân loại theo 03 mức độ: Thấp (Low), trung bình (Substantial), cao (High) được mô tả chi tiết trong Bảng 1 dưới đây:

|  |  |  |
| --- | --- | --- |
| **Mức độ bảo đảm** | **Mức độ bảo đảm bảo của danh tính điện tử (chứng minh danh tính khi đăng ký)** | **Mức độ xác thực** |
| **Thấp** | Cung cấp danh tính từ cơ quan chức năng (từ xa hoặc trực tiếp) | Một yếu tố (mật khẩu hoặc mã pin) |
| **Trung bình** | Cung cấp danh tính (từ xa hoặc trực tiếp); Xác thực danh tính bởi cơ quan đăng ký | Nhiều yếu tố (điện thoại di động kết hợp với mã PIN |
| **Cao** | Cung cấp giấy tờ danh tính trực tiếp tại cơ quan đăng ký; Xác thực danh tính sử dụng các nguồn chính thống và các tài liệu của cơ quan quản lý | Nhiều yếu tố; Phải truy nhập tới dữ liệu/khóa cá nhân trên các thiết bị vật lý; Có giải pháp mã hóa bảo vệ thông tin định danh cá nhân |

*Bảng 1-1: Mức độ đảm bảo danh tính điện tử*

# CHƯƠNG II. DỮ LIỆU TRÊN THẺ DANH TÍNH ĐIỆN TỬ

## 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.

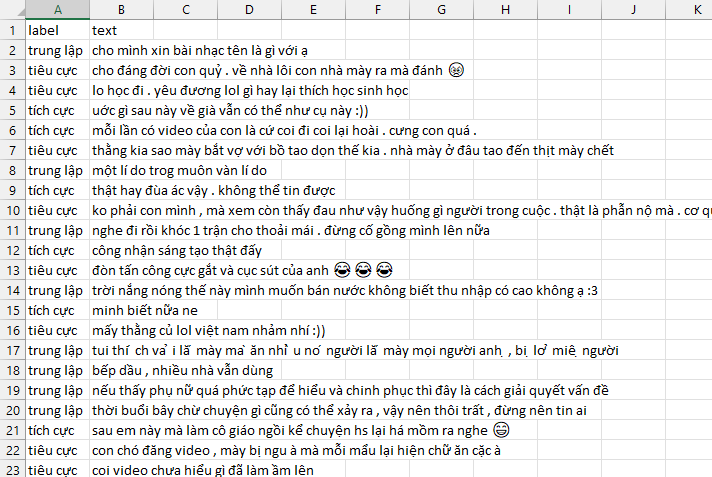
### b. Tập dữ liệu

* **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.

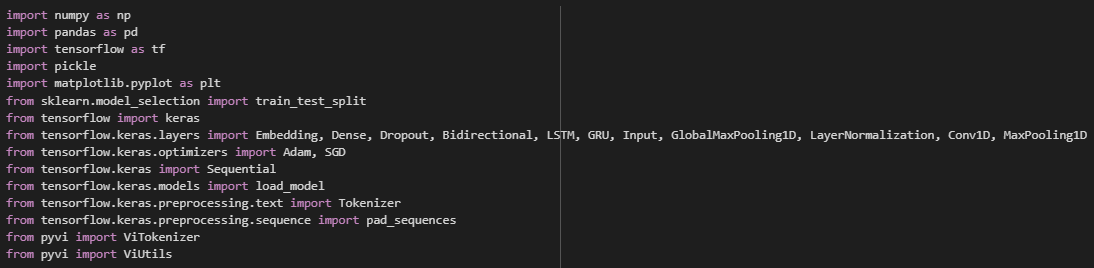
Trong bộ dữ liệu này, Vietnamese Social Media Emotion – (UIT-VSMEC) với khoảng hơn 6.900 câu có chú thích của con người với 3 nhãn, góp phần vào nghiên cứu nhận dạng cảm xúc trong tiếng Việt, vốn là ngôn ngữ sử dụng ít tài nguyên trong Xử lý ngôn ngữ tự nhiên (NLP). Với dữ liệu này giúp chúng ta huấn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

Mô tả file **\*.CSV** dữ liệu:



Hình 14. Tập dữ liệu

### c. Các thư viện.



Hình 15. 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:

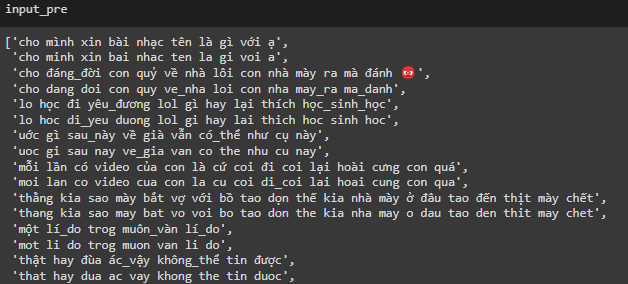
* Core Layers: các khối xây dựng cơ bản như lớp Dense, lớp Activation, lớp Dropout,v.v
* Convolutional Layers: được sử dụng cho những dữ liệu liên quan tới mô hình Convolutional Neural Network
* Recurrent Layers: được sử dụng cho dữ liệu tuần tự, ví dụ: LSTM, GRU
* Normalization Layers: được sử dụng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

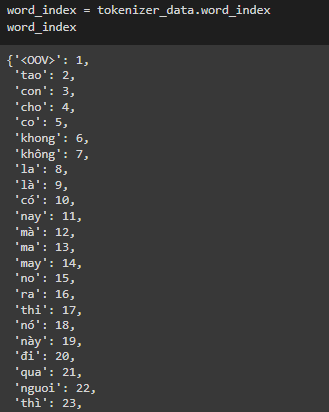
Trong các bài toán Xử lý ngôn ngữ tự nhiên (NLP) khi xử lý dữ liệu ta thường có một bước quan trọng gọi là Tokenization. Quá trinh này giúp phân chia văn bản thành các đơn vị nhỏ hơn, gọi là token. Token có thể là từ đơn, cụm từ, câu hoặc thậm chí là các ký tự riêng lẻ, tuỳ thuộc vào ứng dụng cụ thể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



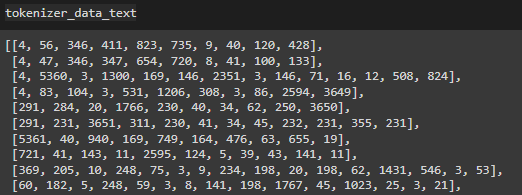
Hình 16. Tokenization bằng thư viện pyvi

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.



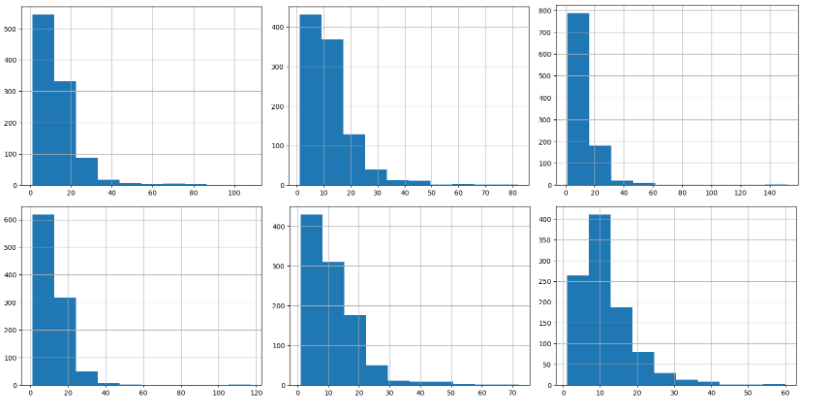
Hình 17. Bộ từ điển được mapping

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.

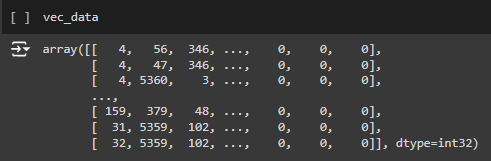


Hình 18. Độ dài mỗi câu trước khi padding

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.

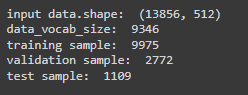


Hình 19. Visualization tập dữ liệu



Hình 20. 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.

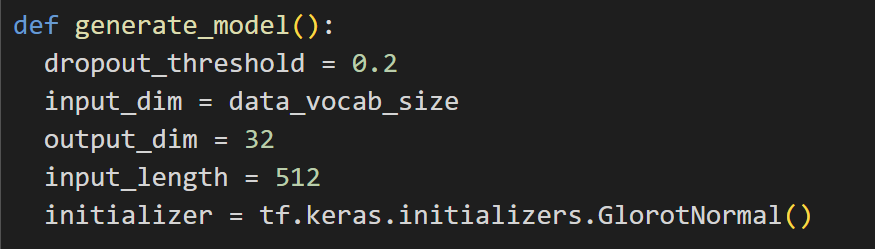


Hình 21. Chia tập dữ liệu

## 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.

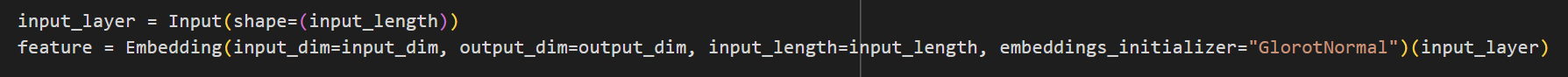
Trước khi xây dựng các lớp model em sẽ định nghĩa các siêu tham số và bộ khởi tạo như sau:



Hình 22. Siêu tham số và bộ khởi tạo

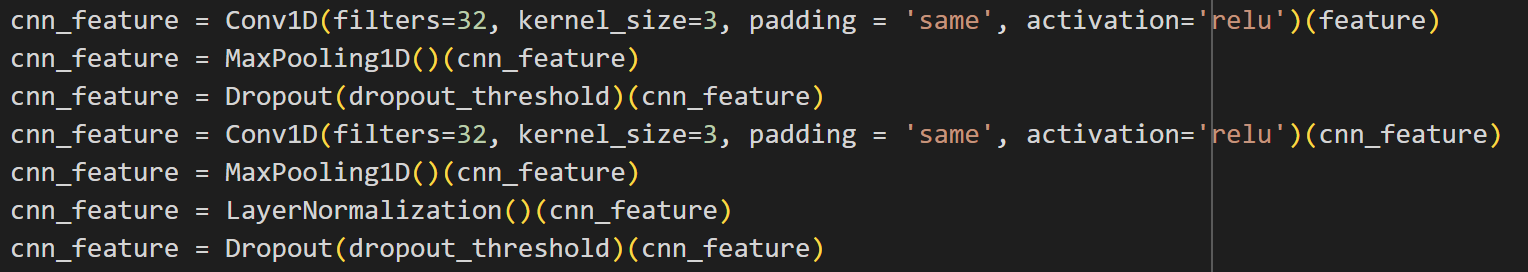
* **Dropout\_threshold**: tỷ lệ dropout để tránh overfitting.
* **Input\_dim** là kích thước từ vựng của dữ liệu đầu vào, em sẽ để nó bằng với data\_vocab\_size để đảm bảo rằng lớp Embedding có thể ánh xạ tất cả các từ trong tập từ vựng một cách chính xác và hiệu quả.
* **Input\_length** sẽ tương đương 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 23. Khởi tạo Embedding

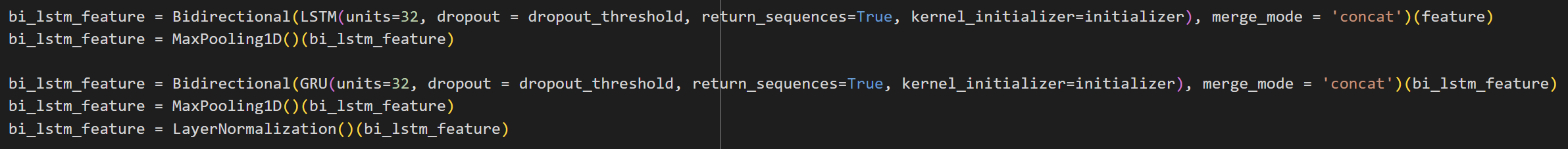
Sau khi có được lớp feature Embedding ở trên em sẽ tiến hành xây dựng các feature từ CNN và Bi-LSTM.



Hình 24. 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.

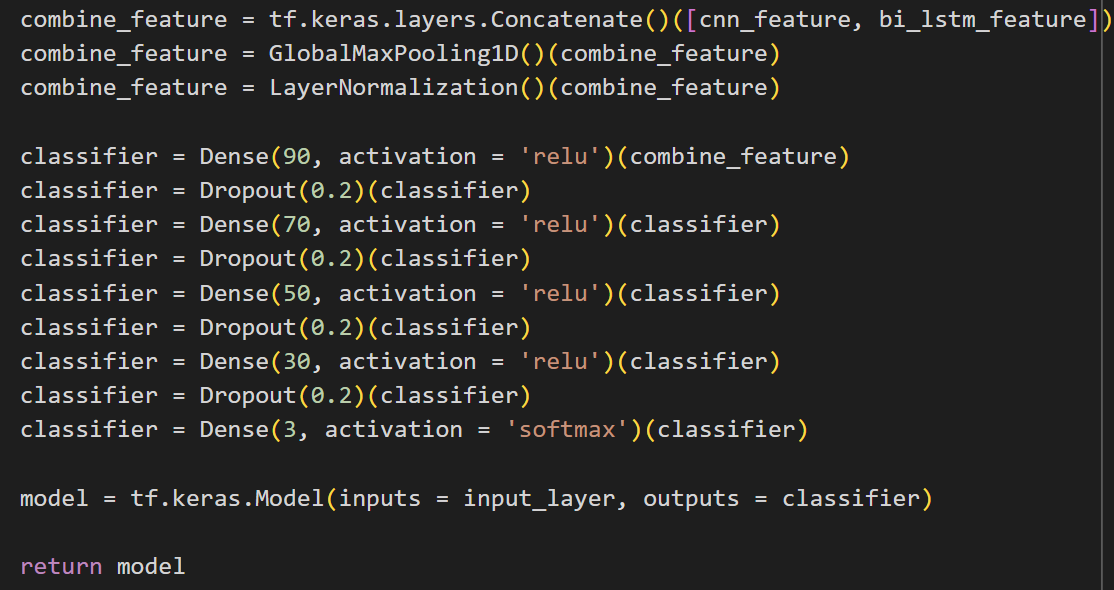
Tiếp theo em tạo thêm các lớp LSTM và GRU có tính chất song song hai chiều (Bidirectional) cùng với các lớp MaxPooling1D và LayerNormalization để xử lý và chuẩn hoá đầu ra



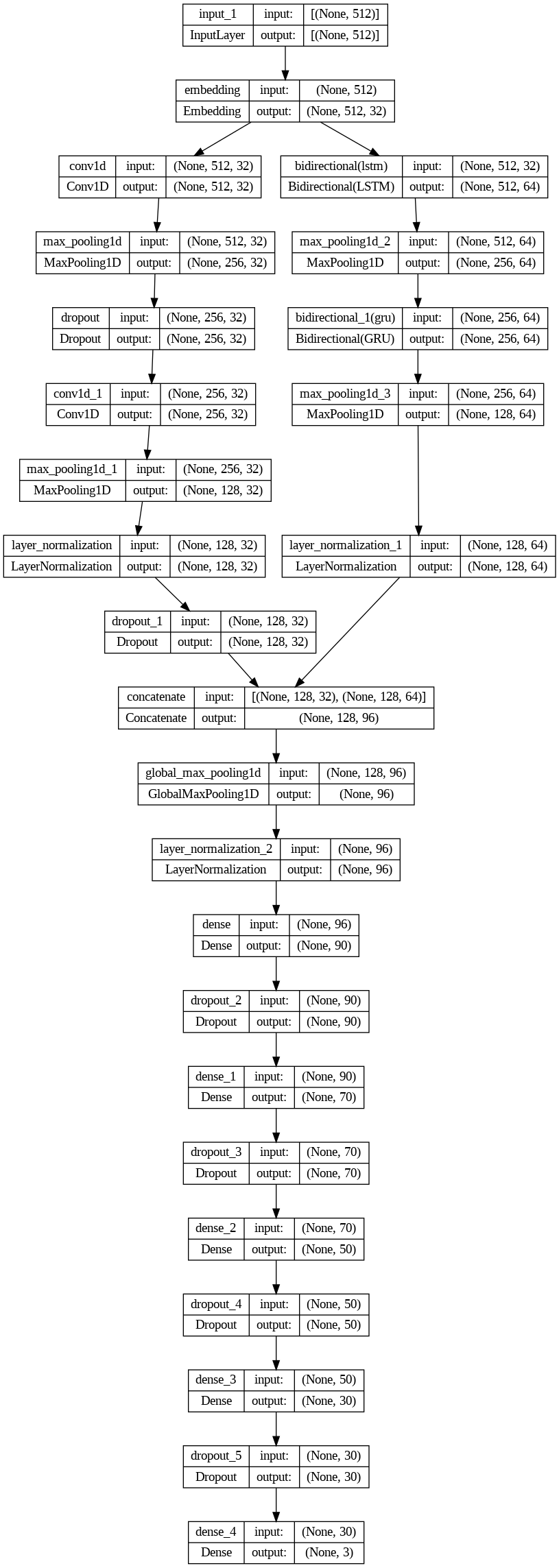
Hình 25. Tạo lớp mạng BiLSTM

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.



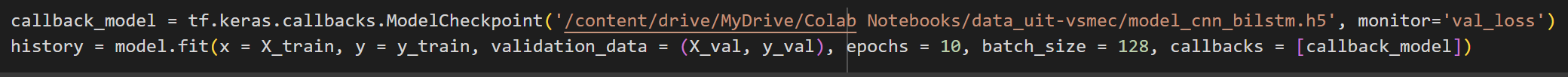
Hình 26. Nối các lớp đặc trưng



Hình 27. Kiến trúc mô hình

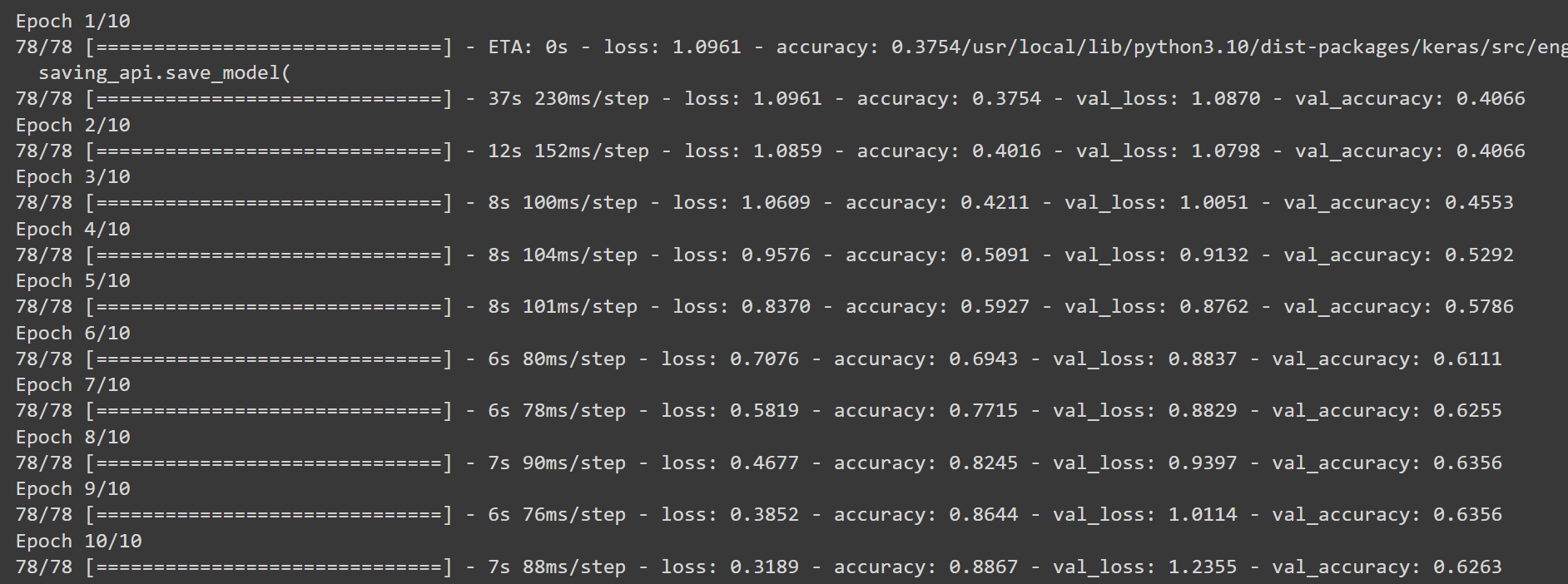
## 4. Huấn luyện và đánh giá mô hình

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 28. Huấn luyện mô hình

Kết quả sau khi train xong 10 epochs



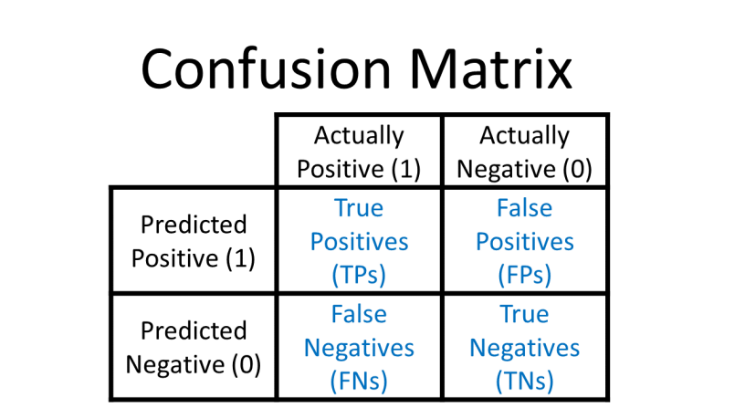
Hình 29. Kết quả huấn luyện

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 30. 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 =

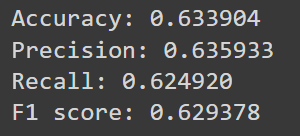
* Recall: kiểm tra các kết quả dự đoán tích cực chính xác trong số các kết quả tích cực

Recall =

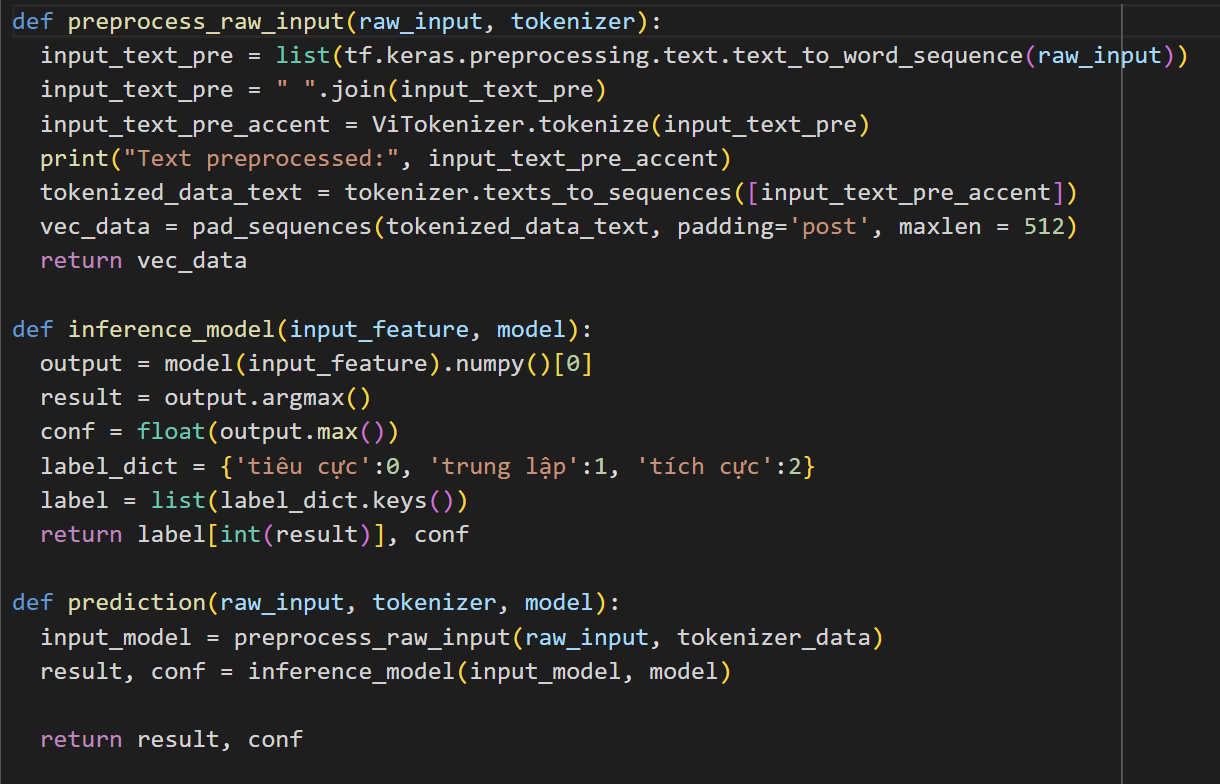
* F1\_score: là trung bình hài hòa của Accuracy và recall, thể hiện sự đóng góp của cả hai. Sự đóng góp phụ thuộc vào giá trị beta, nếu sự đóng của cả 2 là như nhau thì ta có:

=

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

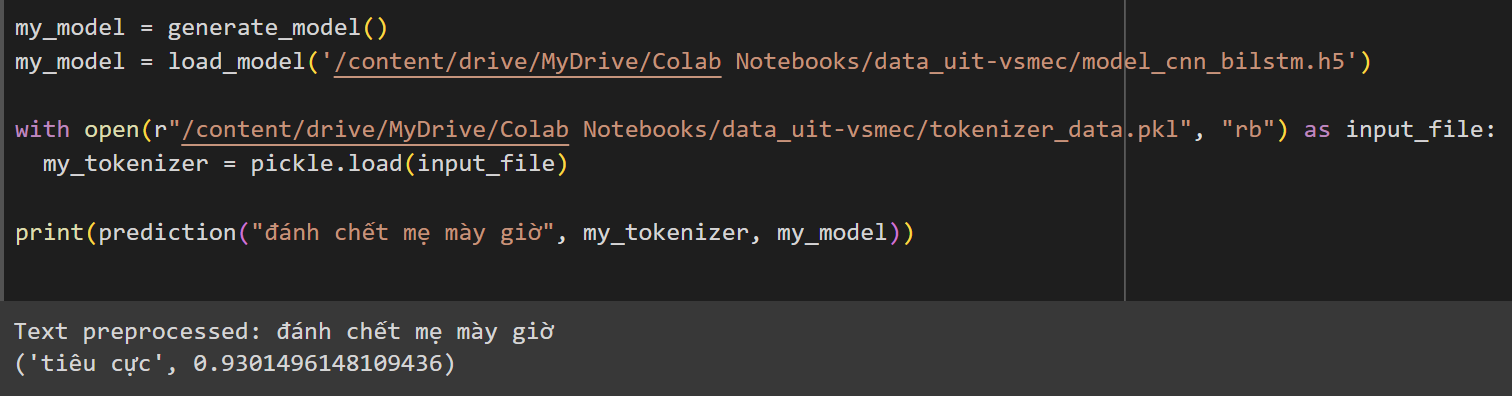


## 5. Dự đoán với mô hình đã huấn luyện



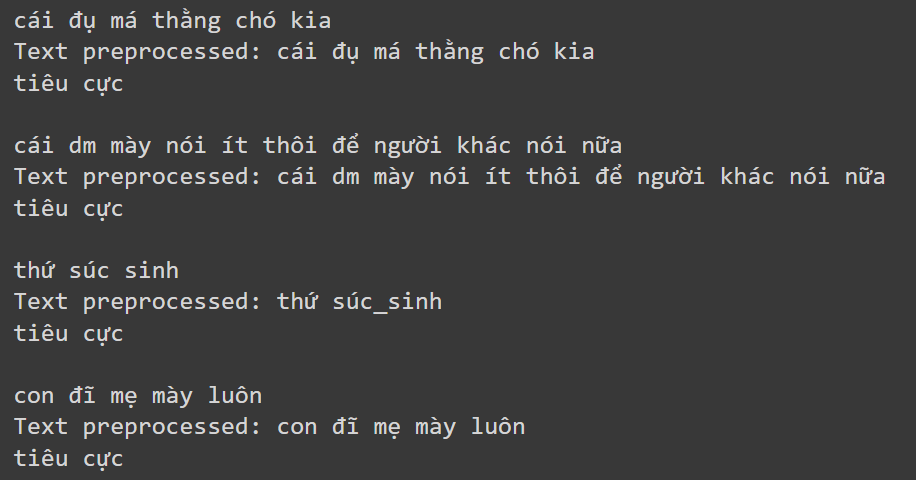
Hình 31. Các hàm để dự đoán Input mới

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 32. 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 33. 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.

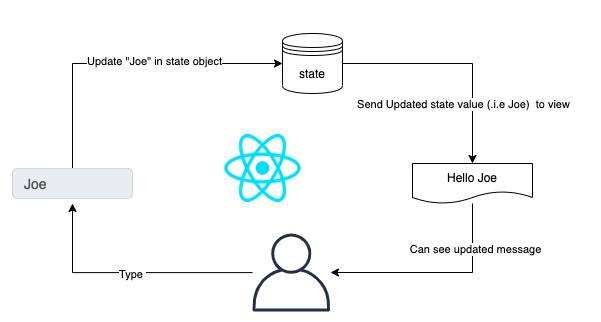
**- JSX (JavaScript XML):** JSX (nói ngắn gọn là JavaScript extension) là một React extension giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản. Và kể từ lúc ReactJS browser hỗ trợ toàn bộ những trình duyệt Web hiện đại, bạn có thể tự tin sử dụng JSX trên bất kỳ trình duyệt nào 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 34. Vòng đời của State trong ReactJs

**- Lifecycle Methods:**

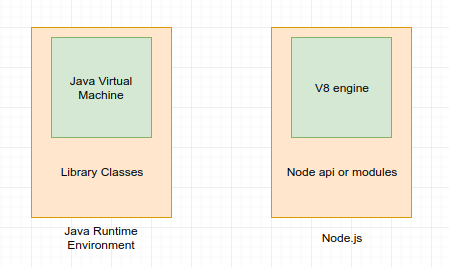
* Các phương thức được gọi tại các giai đoạn khác nhau trong vòng đời của một component.
* Các giai đoạn bao gồm mounting, updating và unmounting.

**- 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 35. 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.

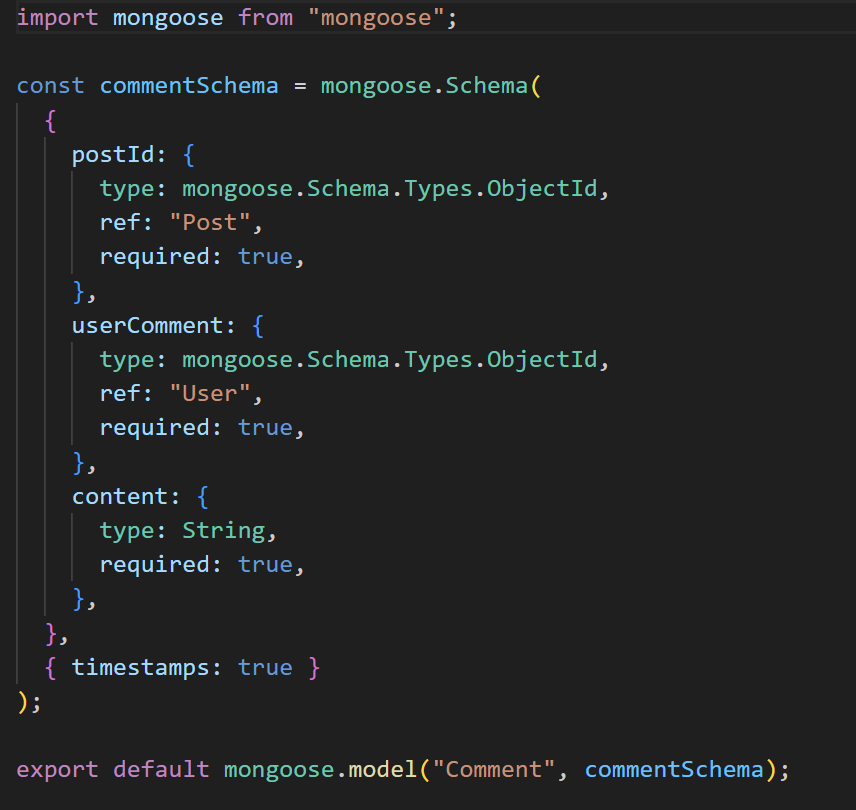
Dưới đây là một số cơ sở dữ liệu được xây dựng bằng MongoDB



Hình 36. Cơ sở dữ liệu Người dùng



Hình 37. Cơ sở dữ liệu Bài viết



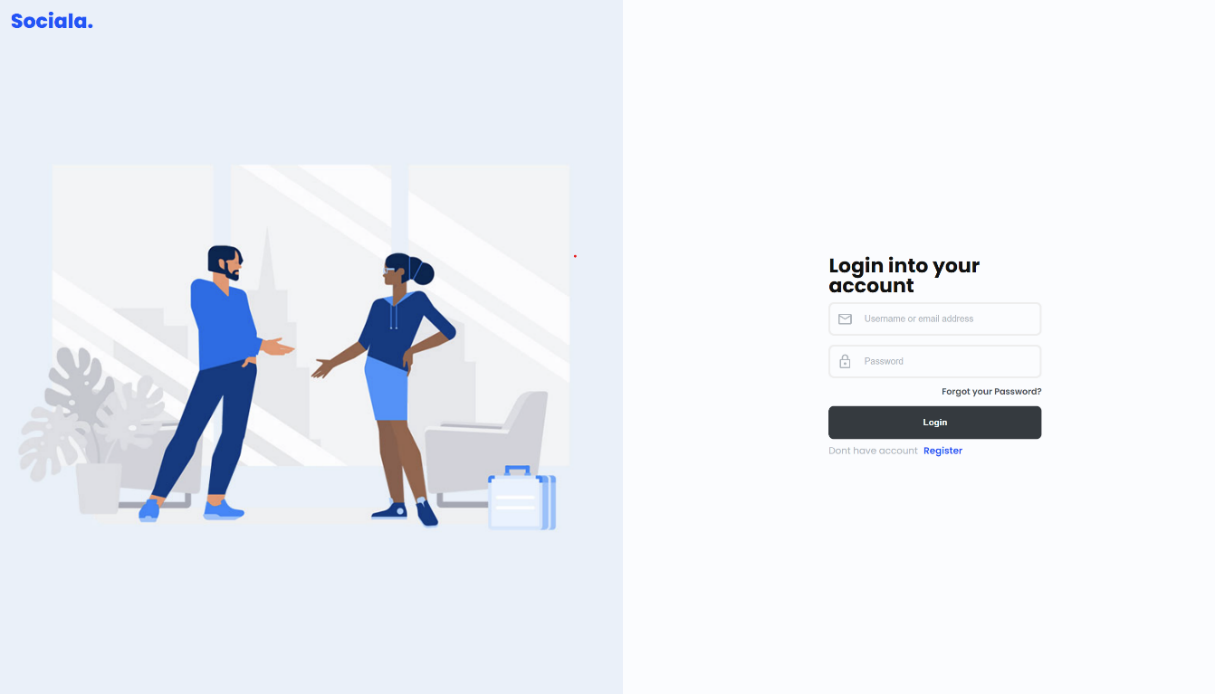
Hình 38. Cơ sở dữ liệu Bình luận



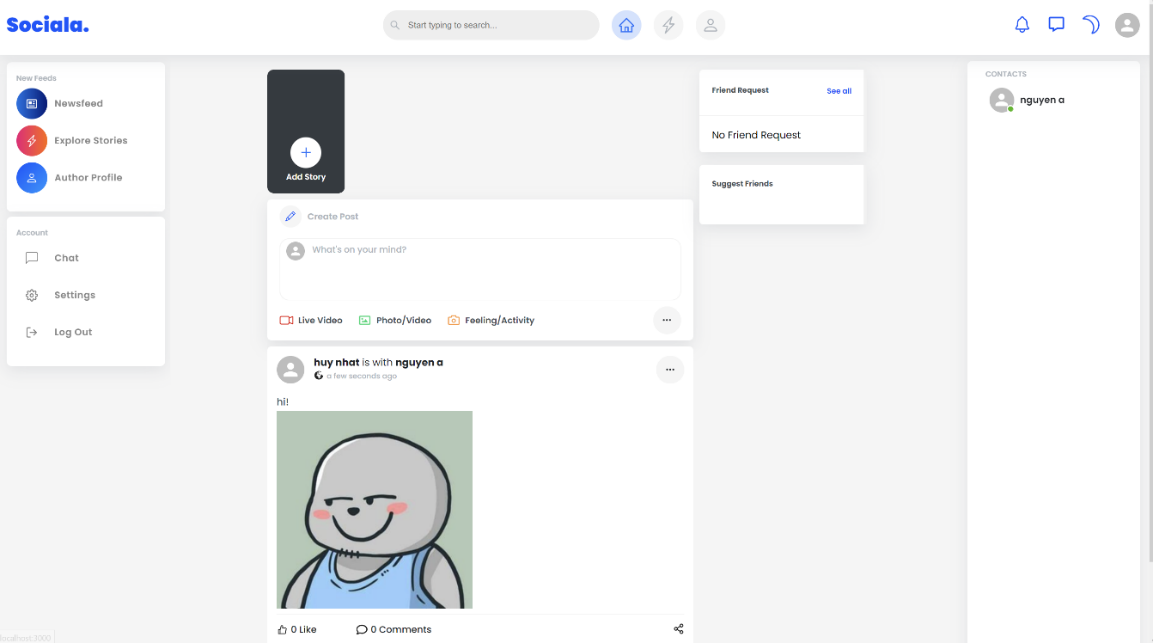
Hình 39. Cơ sở dữ liệu Trò chuyện

### 1.2. Giao diện

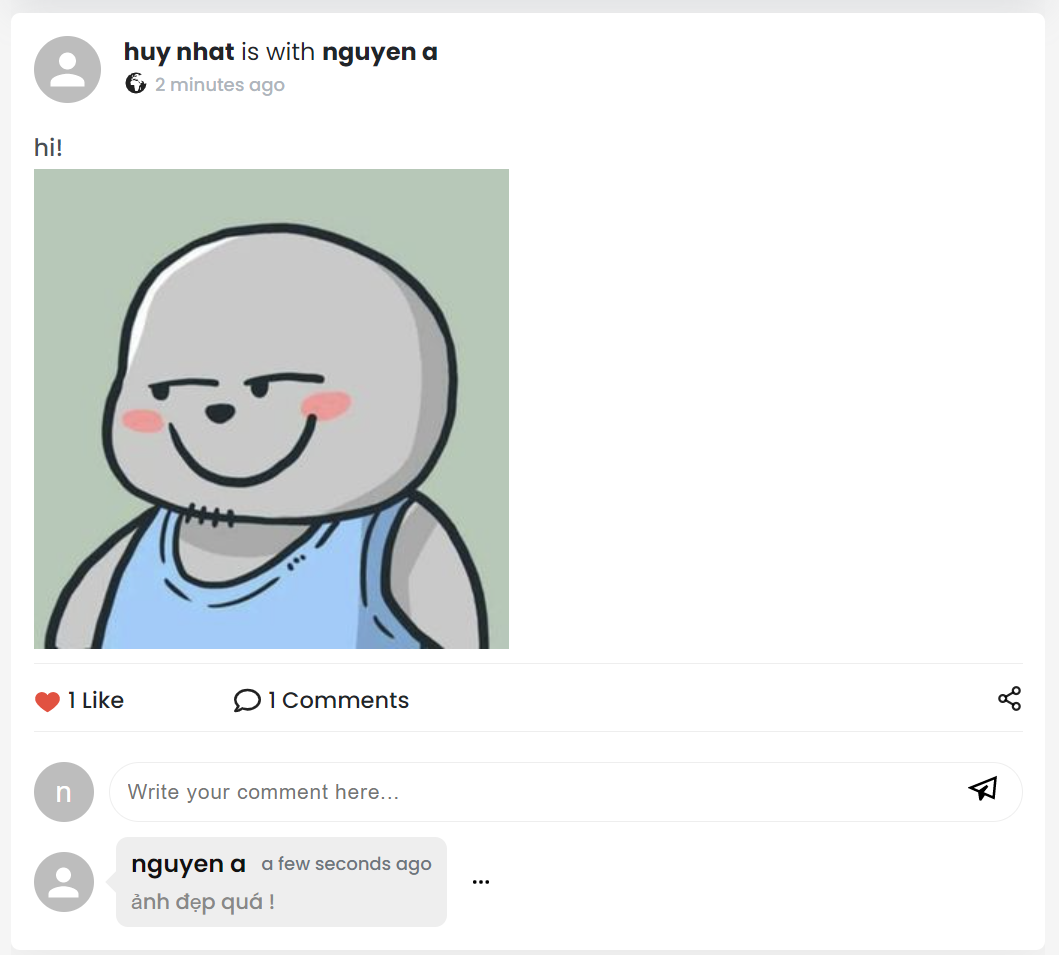
Dưới đây là một số giao diện của trang web Mạng xã hội



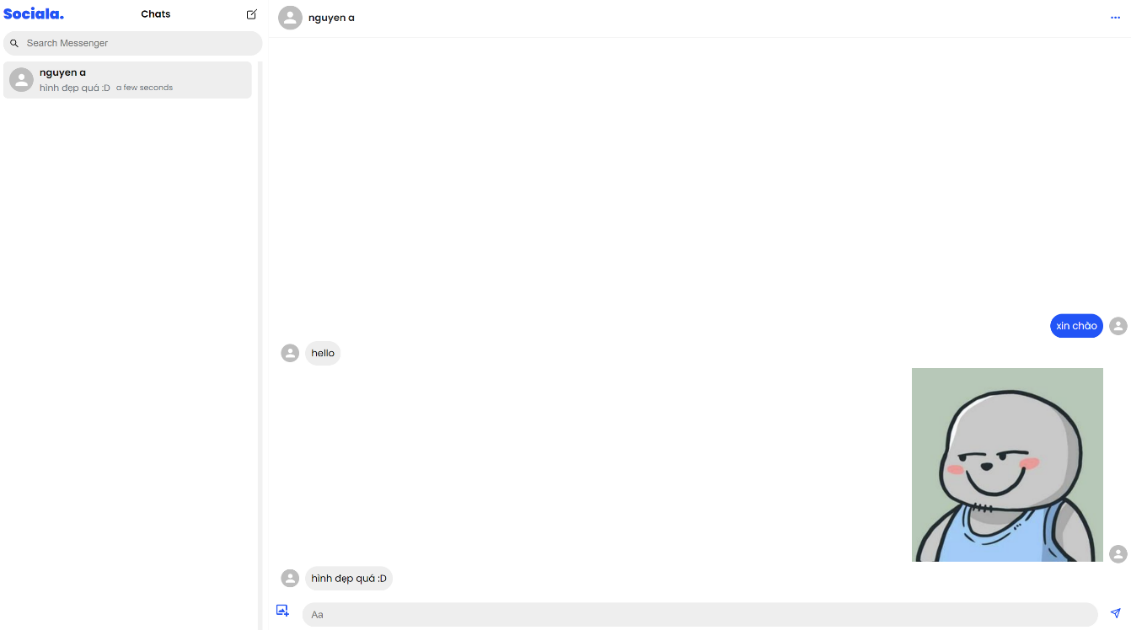
Hình 40. Giao diện đăng nhập



Hình 41. Giao diện Trang chủ



Hình 42. Giao diện Bài viết



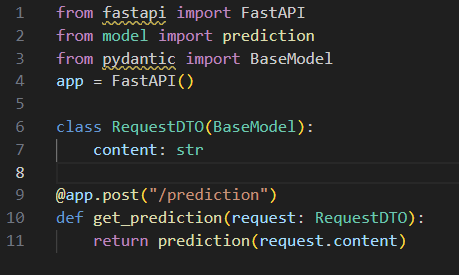
Hình 43. Giao diện Trò chuyện

## 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.

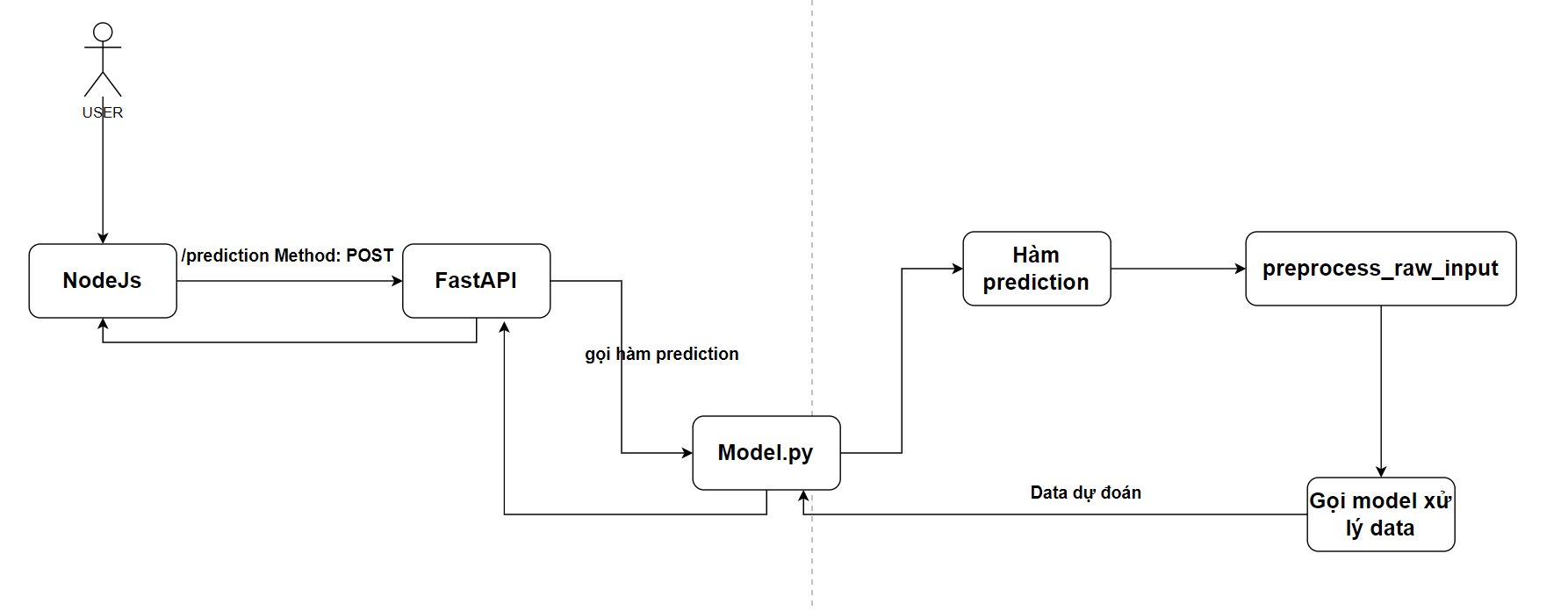
Dưới đây em sẽ sử dụng FastAPI để tạo một API cho model:



Hình 44. Tạo 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 45. Mô hình FastAPI hoạt động

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

Hình ảnh demo khi người dùng bình luận những câu văn nhạy cảm thì bình luận sẽ bị làm mờ

# KẾT LUẬN VÀ KIẾN NGHỊ

## 1. Kết quả đạt được

## 2. Những vấn đề còn tồn đọng

## 3. 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]