TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

**----- o0o -----**

A blue and white logo

Description automatically generated with low confidence

**ĐỒ ÁN TỐT NGHIỆP**

**Tên đề tài:**

**NGHIÊN CỨU XỬ LÝ NGÔN NGỮ TỰ NHIÊN VÀ XÂY DỰNG ỨNG DỤNG TRẢ LỜI VĂN BẢN HÀNH CHÍNH BẰNG PYTHON**

|  |  |
| --- | --- |
|  |  |
| **SINH VIÊN THỰC HIỆN:** | **HOÀNG TUẤN VŨ** |
| **MÃ SINH VIÊN :**  **LỚP:** | **2055010293**  **20CN5** |

**Hà Nội, tháng 2 năm 2025**

TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

**----- o0o -----**

A blue and white logo

Description automatically generated with low confidence

**ĐỒ ÁN TỐT NGHIỆP**

**Tên đề tài:**

**NGHIÊN CỨU XỬ LÝ NGÔN NGỮ TỰ NHIÊN VÀ XÂY DỰNG ỨNG DỤNG TRẢ LỜI VĂN BẢN HÀNH CHÍNH BẰNG PYTHON**

|  |  |
| --- | --- |
| **GIÁO VIÊN HƯỚNG DẪN:** | **Th.S NGUYỄN THỊ HUỆ** |
| **SINH VIÊN THỰC HIỆN:** | **HOÀNG TUẤN VŨ** |
| **MÃ SINH VIÊN :**  **LỚP:** | **2055010293**  **20CN5** |

**Hà Nội, tháng 2 năm 2025**

# **LỜI CẢM ƠN**

Qua một thời gian tìm hiểu và thực hiện đến nay đề tài “***Nghiên cứu xử lý ngôn ngữ tự nhiên và xây dựng ứng dụng trả lời văn bản hành chính bằng python***” đã hoàn thành. Trong suốt quá trình thực hiện đề tài, em đã nhận được rất nhiều sự giúp đỡ nhiệt tình của thầy cô và các mọi người xung quanh.

Em xin chân thành cảm ơn các thầy cô đã trang bị những kiến thức quý báu cho em trong suốt quá trình học tập tại trường Đại Học Kiến Trúc Hà Nội . Đặc biệt là các thầy các cô trong Khoa Công nghệ thông tin đã tận tình giảng dạy, chỉ bảo, trang bị cho em những kiến thức cần thiết nhất trong suốt quá trình học tập và nghiên cứu tại khoa, thầy cô đã tạo điều kiện thuận lợi giúp em thực hiện đề tài này.

Em xin chân thành gửi lời cảm ơn sâu sắc đến **ThS. Nguyễn Thị Huệ**, người đã tận tâm chỉ bảo em trong quá trình thực hiện đồ án. Những lời khuyên của cô đã giúp em nhận ra rất nhiều điều, rút ra nhiều kinh nghiệm khi thực hiện đồ án này.

Mặc dù em đã cố gắng nỗ lực hết mình để thực hiện để tài nhưng em không thể tránh khỏi thiếu sót, kính mong sự đóng góp và hướng dẫn của các thầy cô.

**Em xin chân thành cảm ơn!**

# **LỜI CAM ĐOAN**

Em xin cam đoan bài báo cáo tốt nghiệp “***Nghiên cứu xử lý ngôn ngữ tự nhiên và xây dựng ứng dụng trả lời văn bản hành chính bằng python***” là công trình nghiên cứu của cá nhân em, các kết quả nghiên cứu có tính độc lập riêng, không sao chép bất kì tài liệu nào và chưa công bố nội dung này ở bất kì đâu, có tính kế thừa, phát triển từ các tài liệu, tạp chí, các công trình nghiên cứu đã được công bố, các website.

Em xin hoàn toàn chịu trách nhiệm về lời cam đoan của mình.

**Sinh Viên**

**Hoàng Tuấn Vũ**

# **LỜI NÓI ĐẦU**

Ngày nay với sự phát triển của khoa học công nghệ và đặc biệt là sự bùng nổ trong lĩnh vực Trí tuệ nhân tạo, đã mang lại cho con người nhiều tiện ích trong cuộc sống. Dựa trên tiêu chí đó để có thể giúp người dùng trong lĩnh vực văn bản hành chính trong công việc và đời sống, đồ án này nhằm nghiên cứu về lĩnh vực Xử lý Ngôn ngữ Tự nhiên với mục tiêu xây dựng một ứng dụng hỗ trợ trả lời các văn bản hành chính.

Xử lý Ngôn ngữ Tự nhiên là một lĩnh vực không còn mới tuy nhiên vẫn luôn là một trong những lĩnh vực quan trọng nhất của Trí tuệ nhân tạo hiện nay. Những thành tựu trong lĩnh vực này không chỉ đơn thuần thay đổi cách con người tương tác với công nghệ mà còn góp phần thúc đẩy quá trình chuyển đổi số, mang lại những giá trị to lớn cho xã hội.

Thông qua việc nghiên cứu chi tiết về đề tài ứng dụng xử lý ngôn ngữ tự nhiên về văn bản hành chính, với mục đích để có thể hiểu hơn về kiến thức trí tuệ nhân tạo nói chung cũng như mong muốn có thể giúp đỡ được người dùng.

Đồ án bao gồm 4 chương:

Chương 1. Cơ sở lý thuyết

Chương 2. Quy trình chuẩn bị và huấn luyện mô hình

Chương 3. Triển khai và thực nghiệm

Chương 4. Kết luận và hướng phát triển

# **MỤC LỤC**

[**LỜI CẢM ƠN** i](#_Toc189754502)

[**LỜI CAM ĐOAN** ii](#_Toc189754503)

[**LỜI NÓI ĐẦU** iii](#_Toc189754504)

[**MỤC LỤC** iv](#_Toc189754505)

[**DANH MỤC HÌNH ẢNH** v](#_Toc189754506)

[**DANH MỤC TỪ VIẾT TẮT** vi](#_Toc189754507)

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT** 1](#_Toc189754508)

[**1.1. Tổng quan về trí tuệ nhân tạo** 1](#_Toc189754509)

[**1.1.1. Khái niệm** 1](#_Toc189754510)

[**1.1.2. Các hướng nghiên cứu** 2](#_Toc189754511)

[**1.1.3. Phân loại trí tuệ nhân tạo** 3](#_Toc189754512)

[**1.1.4. Ứng dụng** 4](#_Toc189754513)

[**1.1.5. Mặt tích cực và hạn chế của trí tuệ nhân tạo** 6](#_Toc189754514)

[**1.2. Tổng quan Deep Learning** 7](#_Toc189754515)

[**1.3. Tổng quan xử lý ngôn ngữ tự nhiên** 11](#_Toc189754516)

[**1.4. Giới thiệu về Transformer** 15](#_Toc189754517)

[**1.5. BERT (Bidirectional Encoder Representations from Transformers)** 20](#_Toc189754518)

[**CHƯƠNG 2: QUY TRÌNH CHUẨN BỊ VÀ HUẤN LUYỆN MÔ HÌNH** 22](#_Toc189754519)

[**2.1. Tổng quan mô hình** 22](#_Toc189754520)

[**2.2. Chuẩn bị tập dữ liệu** 24](#_Toc189754521)

[**2.3. Tiền xử lý dữ liệu** 26](#_Toc189754522)

[**2.4. Phân chia dữ liệu** 28](#_Toc189754523)

[**2.5. Huấn luyện mô hình** 29](#_Toc189754524)

[**2.6. Đánh giá kết quả** 30](#_Toc189754525)

[**CHƯƠNG 3: TRIỂN KHAI VÀ THỰC NGHIỆM** 33](#_Toc189754526)

[**3.1. Mô tả bài toán** 33](#_Toc189754527)

[**3.2. Công nghệ sử dụng** 35](#_Toc189754528)

[**3.2.1. Python** 35](#_Toc189754529)

[**3.2.2. Nuxt** 36](#_Toc189754530)

[**3.3. Phân tích thiết kế hệ thống** 37](#_Toc189754531)

[**3.3.1. Yêu cầu chức năng** 37](#_Toc189754532)

[**3.3.2. Kiến trúc tổng quan** 37](#_Toc189754533)

[**3.3.3. Mô hình tổng quan chức năng ứng dụng** 38](#_Toc189754534)

[**3.3.4. Mô hình phân rã chức năng** 39](#_Toc189754535)

[**3.3.6. Giao diện ứng dụng** 40](#_Toc189754536)

[**3.4. Quy trình thực nghiệm** 43](#_Toc189754537)

[**3.5. Đánh giá kết quả** 45](#_Toc189754538)

[**CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 46](#_Toc189754539)

[**4.1. Kết luận về đề tài** 46](#_Toc189754540)

[**4.2. Hướng phát triển của đề tài** 47](#_Toc189754541)

[**TÀI LIỆU THAM KHẢO** 50](#_Toc189754542)

[**PHỤ LỤC** 51](#_Toc189754543)

[**5.1. Hướng dẫn cài đặt ứng dụng** 51](#_Toc189754544)

[**5.2. Hướng dẫn sử dụng ứng dụng** 51](#_Toc189754545)

# **DANH MỤC HÌNH ẢNH**

[Hình 1-1: Trí tuệ nhân tạo 6](#_Toc187892202)

[Hình 1-3: Deep learning 12](#_Toc187892203)

[Hình 1-4: Mô hình transformer 19](#_Toc187892204)

[Hình 1-5: Hình ảnh gán nhãn đầu vào 20](#_Toc187892205)

[Hình 1-6: Multihead Attention 22](#_Toc187892206)

[Hình 1-7: Feed Forward 22](#_Toc187892207)

[Hình 1-8: Masked Multi-head Attention 23](#_Toc187892208)

[Hình 1-9: Next Sentence Prediction 25](#_Toc187892209)

[Hình 1-10: Logo Python 26](#_Toc187892210)

[Hình 1-11: Logo Nuxt 27](#_Toc187892211)

[Hình 3-1: Mô hình tổng quan chức năng ứng dụng 40](#_Toc187892212)

[Hình 3-2: Mô hình chức năng ứng dụng 41](#_Toc187892213)

[Hình 3-3: Giao diện ban đầu của ứng dụng 41](#_Toc187892214)

[Hình 3-4: Thanh lịch sử trò chuyện 42](#_Toc187892215)

[Hình 3-5: Thanh trò chuyện dùng để người dùng đặt câu hỏi 42](#_Toc187892216)

[Hình 3-6: Giao diện cuộc trò chuyện của ứng dụng 43](#_Toc187892217)

# **DANH MỤC TỪ VIẾT TẮT**

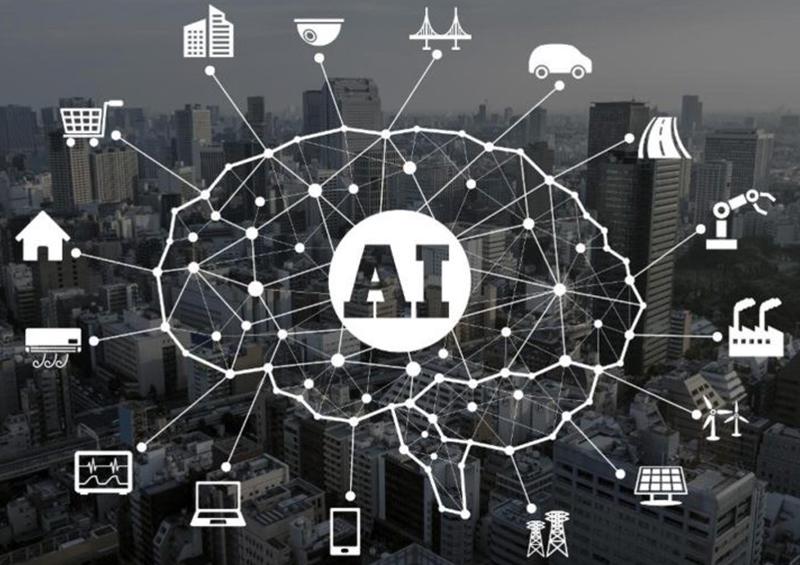
|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Nội dung đầy đủ** |
| 1 | AI | Artificial Intelligence |
| 2 | NLP | Natural Language Processing |
| 3 | BERT | Bidirectional Encoder Representations from Transformers |
| 4 | RNN | Recurrent Neural Network |
| 5 | LSTM | Long short term memory |
| 6 | NN | Neural Network |
| 7 | CNN | Convolutional Neural Network |
| 8 | MLP | Multi-Layer Perceptron |
| 9 | TP | True Positive |
| 10 | TN | True Negative |
| 11 | FP | False Positive |
| 12 | FN | False Negative |
| 13 | API | Application Programming Interface |

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

## **1.1. Tổng quan về trí tuệ nhân tạo**

### **1.1.1. Khái niệm**

Trí tuệ nhân tạo (AI) là một lĩnh vực trong khoa học máy tính và công nghệ thông tin tập trung vào việc phát triển máy tính và hệ thống có khả năng thực hiện các nhiệm vụ thông minh mà trước đây chỉ có con người có thể thực hiện. Mục tiêu của AI là tạo ra các chương trình máy tính hoặc hệ thống thông minh, cho phép chúng học hỏi, tự điều chỉnh và thực hiện các tác vụ yêu cầu sự hiểu biết, lý thuyết cũng như khả năng giải quyết vấn đề.



Hình 1-1: Trí tuệ nhân tạo

### **1.1.2. Các hướng nghiên cứu**

Trí tuệ nhân tạo (AI) là một lĩnh vực đa dạng và rất rộng với nhiều hướng nghiên cứu khác nhau mà các nhà nghiên cứu trên khắp thế giới đang tiến hành. Dưới đây là một số hướng nghiên cứu quan trọng trong lĩnh vực này:

* Học máy và Học sâu (Machine Learning and Deep Learning): Nghiên cứu về các thuật toán học máy và học sâu để cải thiện khả năng học và tự điều chỉnh của các hệ thống AI. Điều này bao gồm việc tối ưu hóa các mô hình học máy hiện có và phát triển các mô hình mới.
* Thị giác máy tính (Computer Vision): Nghiên cứu về khả năng của máy tính để nhận diện, hiểu và xử lý hình ảnh và video. Các ứng dụng bao gồm nhận diện khuôn mặt, phân tích hình ảnh y tế và xe tự hành.
* Xử lý ngôn ngữ tự nhiên (NLP): Nghiên cứu về khả năng của máy tính để hiểu và tạo ra ngôn ngữ tự nhiên. Điều này liên quan đến chatbots, dịch máy, phân tích cảm xúc và nhiều ứng dụng khác.
* Trí Tuệ Nhân Tạo Đàm Thoại (Conversational AI): Là 1 nhánh của xử lý ngôn ngữ tự nhiên có khả năng phân tích và hiểu giọng nói của con người. Có thể giao tiếp phát âm bằng giọng nói giống như con người.
* Robotics và Trí tuệ nhân tạo trong Robot (Robotics and AI in Robotics): tập trung vào phát triển robot có khả năng tương tác với môi trường xung quanh một cách tự động. Các robot có thể thực hiện nhiệm vụ từ đơn giản đến phức tạp, như trong công nghiệp, y tế và dịch vụ khách hàng.
* Hệ thống thông minh: Hệ thống thông minh là các hệ thống có khả năng xử lý dữ liệu phức tạp, học hỏi từ kinh nghiệm và đưa ra quyết định hoặc hành động thông minh. Chúng thường kết hợp các kỹ thuật từ nhiều lĩnh vực AI.

### **1.1.3. Phân loại trí tuệ nhân tạo**

Bốn loại của trí tuệ nhân tạo đó là: Máy phản ứng, Bộ nhớ hạn chế, Lý thuyết tâm trí, Tự nhận thức.

**Máy phản ứng (Reactive Machines):**

* Máy phản ứng là cấp độ đơn giản nhất của AI. AI sẽ có khả năng phân tích những động thái khả nghi nhất của mình và đối thủ. Sau đó, sẽ đưa ra giải pháp tốt nhất.
* Ví dụ: Deep Blue của IBM, một cỗ máy được thiết kế để chơi cờ vua với con người. Deep Blue đánh giá các quân cờ trên bàn cờ và phản ứng với chúng, dựa trên các chiến lược cờ vua được mã hóa trước. Deep Blue không học hỏi hoặc cải thiện khi chơi – nó chỉ đơn giản là ‘phản ứng’. Và nó đánh bại kiện tướng cờ vua Garry Kasparov vào năm 1997.



*Hình 1-2: Deep Blue thi đấu cùng với Karparov*

**Bộ nhớ hạn chế (Limited Memory):**

* Máy có bộ nhớ hạn chế, có thể giữ lại một số thông tin học được từ việc quan sát các sự kiện hoặc dữ liệu trước đó. AI có thể xây dựng kiến thức bằng cách sử dụng bộ nhớ đó kết hợp với dữ liệu được lập trình sẵn.
* Ví dụ: Chatbot Thông Minh ghi nhớ câu hỏi trước của người dùng trong cuộc trò chuyện từ đó đưa ra phản hồi tốt hơn.

**Lý thuyết tâm trí (Theory of Mind):**

* Con người có những suy nghĩ và cảm xúc, ký ức hoặc các mô hình não khác điều khiển và ảnh hưởng đến hành vi của họ.
* Dựa trên tâm lý này, các nhà nghiên cứu lý thuyết về tâm trí hy vọng phát triển các máy tính có khả năng bắt chước các mô hình tinh thần của con người. Máy móc có thể hiểu rằng con người và động vật có những suy nghĩ và cảm xúc có thể ảnh hưởng đến hành vi của chính chúng.
* Lý thuyết về máy móc tâm trí sẽ được yêu cầu sử dụng thông tin thu được từ con người và học hỏi từ nó, sau đó sẽ thông báo bằng cách máy móc giao tiếp hoặc phản ứng với một tình huống khác
* Ví dụ: AI hỗ trợ bác sĩ trong việc đánh giá trạng thái cảm xúc của bệnh nhân, đặc biệt trong các lĩnh vực như tâm lý học hoặc trị liệu.

**Tự nhận thức (Self-awareness):**

* Công nghệ AI này có khả năng tự nhận thức về bản thân, có ý thức và hành xử như con người. Thậm chí, chúng còn có thể bộc lộ cảm xúc cũng như hiểu được những cảm xúc của con người. Đây được xem là bước phát triển cao nhất của công nghệ AI và đến thời điểm hiện tại, công nghệ này vẫn chưa khả thi.

### **1.1.4. Ứng dụng**

**Xử lý ngôn ngữ tự nhiên (NLP):**

* Chatbots và Virtual Assistants: AI được sử dụng để xây dựng các trợ lý ảo và chatbot để tương tác với người dùng và cung cấp thông tin hoặc giải quyết vấn đề.
* Phân tích tình cảm: AI có thể phân tích tình cảm trong văn bản, giúp doanh nghiệp hiểu đánh giá của khách hàng về sản phẩm hoặc dịch vụ.

**Thị giác máy tính (Computer Vision):**

* Nhận diện khuôn mặt: AI có thể nhận diện và xác định khuôn mặt trong hình ảnh hoặc video, thường được sử dụng trong hệ thống đảm bảo an ninh và xác thực.
* Xử lý hình ảnh y tế: Giúp bác sĩ chẩn đoán bệnh dựa trên hình ảnh y tế như X-quang, MRI, CT scans.

**Tự động hóa và Robotica:**

* Ô tô tự lái: AI được sử dụng trong xe ô tô tự lái để phát hiện và tránh các vật thể, biển báo giao thông, và điều khiển xe.
* Robot công nghiệp: Robot được trang bị trí tuệ nhân tạo để thực hiện các nhiệm vụ trong môi trường sản xuất, từ lắp ráp đến kiểm tra chất lượng.

**Dự đoán Tài chính:**

* Dự đoán thị trường tài chính: AI được sử dụng để dự đoán xu hướng thị trường chứng khoán và thực hiện giao dịch tự động.
* Quản lý rủi ro: Các công ty bảo hiểm sử dụng AI để định giá và quản lý rủi ro.

**Y tế và Chăm sóc sức khỏe:**

* Chẩn đoán bệnh: AI có thể phân tích dữ liệu y tế và hình ảnh để hỗ trợ bác sĩ trong việc chẩn đoán và điều trị bệnh.
* Quản lý dữ liệu bệnh nhân: Hệ thống AI có thể quản lý và xử lý thông tin liên quan đến bệnh nhân, giúp cải thiện chất lượng dịch vụ y tế.

**Giáo dục:**

* Học tập trực tuyến và cá nhân hóa giáo dục: AI có thể tạo ra nội dung giảng dạy cá nhân hóa dựa trên nhu cầu và khả năng của từng học sinh.
* Đánh giá học sinh: Hệ thống AI có thể phân tích kết quả học tập và đề xuất cải thiện cho học sinh.

**Điều khiển và Tự động hóa nhà thông minh:**

* Quản lý năng lượng: Hệ thống AI có thể điều khiển thiết bị để tiết kiệm năng lượng trong nhà thông minh.
* Bảo mật: AI có thể giúp giám sát và bảo vệ nhà thông minh khỏi các mối đe dọa an ninh.

**Quảng cáo và Tiếp thị:**

* Tối ưu hóa quảng cáo trực tuyến: AI giúp tối ưu hóa chiến dịch quảng cáo trực tuyến dựa trên dữ liệu về tương tác của người dùng.

**Khám phá dữ liệu và Nghiên cứu khoa học:**

* Xử lý dữ liệu lớn (Big Data): AI giúp phân tích và rút ra thông tin quan trọng từ các nguồn dữ liệu lớn và phức tạp.

### **1.1.5. Mặt tích cực và hạn chế của trí tuệ nhân tạo**

Trí tuệ nhân tạo (AI) mang lại nhiều lợi ích và tiềm năng cách mạng hóa nhiều lĩnh vực khác nhau. Tuy nhiên, cũng có những tích cực và hạn chế cần xem xét:

**Tích cực:**

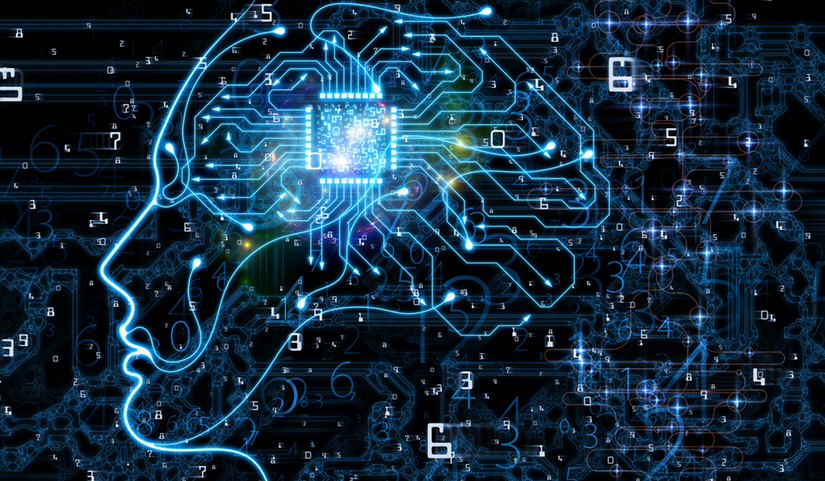
* Tăng năng suất và hiệu suất công việc: AI có thể thực hiện nhiều nhiệm vụ một cách nhanh chóng và chính xác, giúp tăng năng suất và hiệu suất làm việc trong các ngành công nghiệp.
* Dự đoán và ứng dụng trong quản lý tài chính: AI có khả năng phân tích dữ liệu tài chính phức tạp và dự đoán xu hướng thị trường, giúp đưa ra quyết định tài chính hiệu quả.
* Quản lý y tế và chẩn đoán bệnh: Trong lĩnh vực y tế, AI có thể hỗ trợ các bác sĩ trong việc chẩn đoán bệnh và lập kế hoạch điều trị dựa trên phân tích hình ảnh và dữ liệu bệnh lý.
* Ô tô tự hành: AI có tiềm năng thúc đẩy phát triển xe tự hành, giảm tai nạn giao thông và giúp người già hoặc khuyết tật di chuyển dễ dàng hơn.
* Dự đoán thời tiết và khí hậu: AI có khả năng xử lý dữ liệu khí hậu lớn và dự đoán thời tiết một cách chính xác, giúp cảnh báo thiên tai và khắc phục hậu quả.
* Hỗ trợ trong giáo dục: AI có thể cung cấp các công cụ học tập thông minh, cá nhân hóa giáo dục và tạo điều kiện học tập tốt hơn cho học sinh.

**Hạn chế:**

* Sự lo ngại về đạo đức và quyền riêng tư: Sử dụng AI có thể dẫn đến các vấn đề về quyền riêng tư và đạo đức, bao gồm việc thu thập và sử dụng dữ liệu cá nhân một cách không đúng mục đích.
* Thất nghiệp và thay thế công việc: Các hệ thống tự động hóa dự kiến sẽ thay thế một số công việc, gây ra lo ngại về thất nghiệp và sự bất ổn kinh tế.
* Giới hạn trong việc hiểu và giải quyết bài toán phức tạp: AI hiện tại vẫn gặp khó khăn trong việc hiểu và giải quyết bài toán phức tạp mà con người có thể làm.
* Nguy cơ trục trặc và lỗi hệ thống: AI có thể gặp trục trặc và lỗi hệ thống, đặc biệt khi dựa vào dữ liệu không chính xác hoặc không đủ lớn.
* Khả năng phân biệt đạo đức và quyết định etic: AI hiện tại không có khả năng phân biệt đạo đức và quyết định theo tiêu chuẩn etic, gây ra các vấn đề đạo đức liên quan đến quyết định của chúng.
* Phụ thuộc vào dữ liệu lớn: AI cần dữ liệu lớn để hoạt động hiệu quả, và điều này có thể tạo ra vấn đề về riêng tư và bảo mật dữ liệu.

## **1.2. Tổng quan Deep Learning**

Deep Learning (học sâu) có thể được xem là một lĩnh vực con của Machine Learning (học máy) – ở đó các máy tính sẽ học và cải thiện chính nó thông qua các thuật toán. Deep Learning được xây dựng dựa trên các khái niệm phức tạp hơn rất nhiều, chủ yếu hoạt động với các mạng Neural nhân tạo để bắt chước khả năng tư duy và suy nghĩ của bộ não con người.



Hình 1-3: Deep learning

Mạng Neural nhân tạo chính là động lực chính để phát triển Deep Learning. Các mạng Neural sâu (DNN) bao gồm nhiều lớp Neural khác nhau, có khả năng thực hiện các tính toán có độ phức tạp rất cao. Deep Learning hiện đang phát triển rất nhanh và được xem là một trong những bước đột phá lớn nhất trong Machine Learning.

Mạng thần kinh nhân tạo trong Deep Learning được xây dựng để mô phỏng khả năng tư duy của bộ não con người. Cách thức hoạt động như sau:

* Một mạng nơ-ron bao gồm nhiều lớp (layer) khác nhau, càng nhiều lớp thì mạng sẽ càng “sâu”. Trong mỗi lớp là các node (nút mạng) và được liên kết với các lớp khác bên cạnh.
* Mỗi kết nối giữa các nút sẽ có một trọng số tương ứng, trọng số càng cao thì mức độ ảnh hưởng của kết nối này đến mạng nơ-ron càng lớn.
* Mỗi nơ-ron sẽ có một chức năng kích hoạt, về cơ bản chịu trách nhiệm "chuẩn hóa" đầu ra từ nơ-ron này.
* Dữ liệu được người dùng nhập vào mạng thần kinh đi qua tất cả các lớp và trả về kết quả ở layer cuối cùng, được gọi là lớp đầu ra (output layer).
* Trong quá trình đào tạo mô hình mạng nơ-ron, các trọng số sẽ được thay đổi và nhiệm vụ của mô hình là tìm tập giá trị của các trọng số sao cho phán đoán đúng nhất.

**Ưu điểm học sâu:**

Deep Learning là một bước ngoặt lớn trong lĩnh vực trí tuệ nhân tạo - artificial intelligence. Nó cho phép các nhà khoa học dữ liệu khác xây dựng nhiều mô hình có độ chính xác cao trong các lĩnh vực nhận dạng hình ảnh, xử lý ngôn ngữ tự nhiên, data…

* Cấu trúc neural networks linh hoạt, dễ dàng thay đổi để phù hợp với nhiều thuật toán khác nhau.
* Có khả năng giải nhiều vấn đề phức tạp với độ chính xác rất cao.
* Khả năng tự động hóa cao, tự điều chỉnh và tự tối ưu hóa.
* Có khả năng thực hiện tính toán song song, hiệu năng tốt, xử lý lượng dữ liệu lớn.

**Hạn chế của học sâu:**

* Cần khối lượng dữ liệu khổng lồ để khai thác tối đa khả năng của Deep Learning.
* Chi phí tính toán cao vì phải xử lý nhiều mô hình phức tạp.
* Không có nền tảng lý thuyết vững chắc để chọn các công cụ tối ưu cho Deep Learning.

**Các kỹ thuật Deep Learning**

* **Mạng nơ-ron cổ điển:** Mạng nơ-ron cổ điển sử dụng kiến trúc mạng kết nối đầy đủ, thông thường xây dựng từ các perceptron đa tầng, nhằm phân loại các dữ liệu. Perceptron là một thuật toán đơn giản dùng để tìm ranh giới siêu phẳng cho bài toán phân lớp nhị phân.
* **Mạng nơ-ron tích chập (CNN):** Convolutional Neural Network (CNN) là một kiến trúc Neural Network tiên tiến, được phát triển để giải quyết các bài toán phức tạp, đặc biệt là trong lĩnh vực xử lý hình ảnh.
* **Mạng nơ-ron hồi quy (RNN):** Recurrent Neural Network (RNN) là một thuật toán được đánh giá cao. Trong các mô hình mạng nơ-ron truyền thống, đầu vào và đầu ra được xử lý độc lập với nhau. Tuy nhiên, RNN thực hiện cùng một tác vụ cho tất cả các phần tử trong một chuỗi dữ liệu, và đầu ra phụ thuộc vào cả quá trình tính toán trước đó. Điều này cho phép mạng RNN ghi nhớ thông tin từ các phép tính trước đó.
* **Mạng sinh đối nghịch (GAN):** Generative Adversarial Networks (GAN) là một lớp mô hình được thiết kế nhằm tạo ra dữ liệu giả mà có đặc điểm tương tự dữ liệu thực. Tên của mạng này xuất phát từ kiến trúc gồm hai mạng có mục tiêu đối nghịch nhau: Generator và Discriminator. Trong đó, Generator học cách tạo ra dữ liệu giả để đánh lừa mô hình Discriminator, trong khi Discriminator học cách phân biệt giữa dữ liệu giả và dữ liệu thật. Qua quá trình huấn luyện, cả hai mô hình này cùng cải thiện khả năng của mình.
* **Boltzmann machine:** Mạng Boltzmann (Boltzmann machine) là một mô hình mạng không có hướng xác định, trong đó các node của mạng được kết nối với nhau thành một vòng tròn. Mô hình này thường được sử dụng để tạo ra các tham số cho mạng. Các ứng dụng phổ biến của mạng này bao gồm giám sát hệ thống và xây dựng hệ thống khuyến nghị nhị phân.
* **Học tăng cường sâu:** Deep Reinforcement Learning là quá trình mà các tác tử (agent) tương tác với môi trường để thay đổi trạng thái của chính nó và đạt được mục tiêu. Trong quá trình này, các tác tử có khả năng quan sát môi trường và thực hiện hành động phù hợp để tối đa hóa phần thưởng hoặc điểm số.
* **Autoencoder:** Autoencoder là một trong những kỹ thuật Deep Learning phổ biến nhất hiện nay, có khả năng học các biểu diễn của dữ liệu đầu vào mà không cần nhãn, hay nói cách khác thì mạng này có khả năng học không giám sát (unsupervised learning).
* **Backpropagation**: Backpropagation (lan truyền ngược) là một kỹ thuật quan trọng trong mạng nơ-ron. Nó cho phép tính toán gradient ngược từ layer cuối cùng đến layer đầu tiên của mạng. Quá trình này bắt đầu bằng việc mạng phân tích các tham số và điều chỉnh chúng thông qua hàm mất mát. Sau đó, giá trị lỗi được tính toán sẽ được lan truyền ngược lại để điều chỉnh các tham số một cách phù hợp.
* **Gradient Descent:** Trong Deep Learning và tối ưu hoá, việc tìm giá trị nhỏ nhất hoặc lớn nhất của một hàm số là một thách thức phức tạp, và thường khó có thể đạt được nghiệm tối ưu toàn cục.

## **1.3. Tổng quan xử lý ngôn ngữ tự nhiên**

Xử lý ngôn ngữ tự nhiên 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 (speech) hoặc văn bản (text). Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

Xử lý ngôn ngữ tự nhiên có thể được chia ra thành hai nhánh lớn, không hoàn toàn độc lập, bao gồm xử lý tiếng nói (speech processing) và xử lý văn bản (text processing). Xử lý tiếng nói tập trung nghiên cứu, phát triển các thuật toán, chương trình máy tính xử lý ngôn ngữ của con người ở dạng tiếng nói (dữ liệu âm thanh). Các ứng dụng quan trọng của xử lý tiếng nói bao gồm nhận dạng tiếng nói và tổng hợp tiếng nói. Nếu như nhận dạng tiếng nói là chuyển ngôn ngữ từ dạng tiếng nói sang dạng văn bản thì ngược lại, tổng hợp tiếng nói chuyển ngôn ngữ từ dạng văn bản thành tiếng nói. Xử lý văn bản tập trung vào phân tích dữ liệu văn bản. Các ứng dụng quan trọng của xử lý văn bản bao gồm tìm kiếm và truy xuất thông tin, dịch máy, tóm tắt văn bản tự động, hay kiểm lỗi chính tả tự động. Xử lý văn bản đôi khi được chia tiếp thành hai nhánh nhỏ hơn bao gồm hiểu văn bản và sinh văn bản. Nếu như hiểu liên quan tới các bài toán phân tích văn bản thì sinh liên quan tới nhiệm vụ tạo ra văn bản mới như trong các ứng dụng về dịch máy hoặc tóm tắt văn bản tự động.

Xử lý văn bản bao gồm 4 bước chính sau:

* **Phân tích hình vị:** là sự nhận biết, phân tích, và miêu tả cấu trúc của hình vị trong một ngôn ngữ cho trước và các đơn vị ngôn ngữ khác, như từ gốc, biên từ, phụ tố, từ loại, v.v. Trong xử lý tiếng Việt, hai bài toán điển hình trong phần này là tách từ (word segmentation) và gán nhãn từ loại (part-of-speech tagging).
* **Phân tích cú pháp:** là quy trình phân tích một chuỗi các biểu tượng, ở dạng ngôn ngữ tự nhiên hoặc ngôn ngữ máy tính, tuân theo văn phạm hình thức. Văn phạm hình thức thường dùng trong phân tích cú pháp của ngôn ngữ tự nhiên bao gồm Ngữ pháp phi ngữ cảnh (Context-free grammar – CFG), Ngữ pháp danh mục kết nối (Combinatory categorial grammar – CCG), và Ngữ pháp phụ thuộc (Dependency grammar – DG). Đầu vào của quá trình phân tích là một câu gồm một chuỗi từ và nhãn từ loại của chúng, và đầu ra là một cây phân tích thể hiện cấu trúc cú pháp của câu đó.
* **Phân tích ngữ nghĩa:** là quá trình liên hệ cấu trúc ngữ nghĩa, từ cấp độ cụm từ, mệnh đề, câu và đoạn đến cấp độ toàn bài viết, với ý nghĩa độc lập của chúng. Nói cách khác, việc này nhằm tìm ra ngữ nghĩa của đầu vào ngôn từ. Phân tích ngữ nghĩa bao gồm hai mức độ: Ngữ nghĩa từ vựng biểu hiện các ý nghĩa của những từ thành phần, và phân biệt nghĩa của từ; Ngữ nghĩa thành phần liên quan đến cách thức các từ liên kết để hình thành những nghĩa rộng hơn.
* **Phân tích diễn ngôn:** là phân tích văn bản có xét tới mối quan hệ giữa ngôn ngữ và ngữ cảnh sử dụng (context-of-use). Phân tích diễn ngôn, do đó, được thực hiện ở mức độ đoạn văn hoặc toàn bộ văn bản thay vì chỉ phân tích riêng ở mức câu.

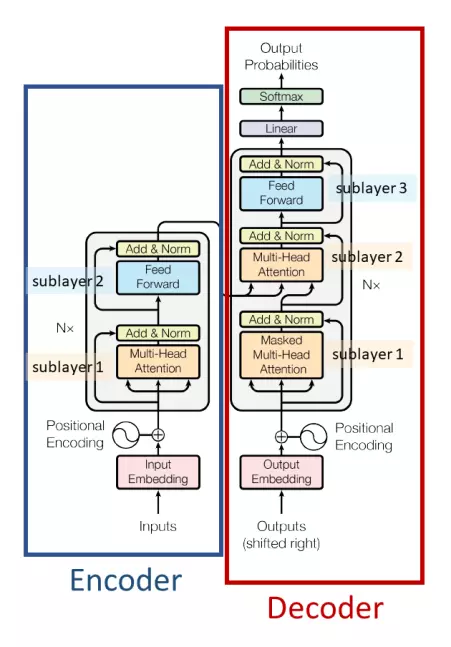
**Một số ứng dụng của NLP**

* **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.
* **Trích chọn thông tin** (Information Extraction – IE) nhận diện một số loại thực thể được xác định trước, mối quan hệ giữa các thực thể và các sự kiện trong văn bản ngôn ngữ tự nhiên. Khác với truy xuất thông tin trả về một danh sách các văn bản hợp lệ thì trích chọn thông tin trả về chính xác thông tin mà người dùng cần. Những thông tin này có thể là về con người, địa điểm, tổ chức, ngày tháng, hoặc thậm chí tên công ty, mẫu sản phẩm hay giá cả.
* **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.

Công nghệ xử lý ngôn ngữ tự nhiên ngày càng có nhiều ứng dụng tốt, dần phổ biến, và gần gũi với cuộc sống hàng ngày. Với sự tiến bộ không ngừng về mặt kỹ thuật, tốc độ xử lý, độ chính xác được cải thiện, xử lý ngôn ngữ tự nhiên đã từng bước trở thành công nghệ giúp máy tính hiểu và giao tiếp với con người thông qua ngôn ngữ của chính con người.

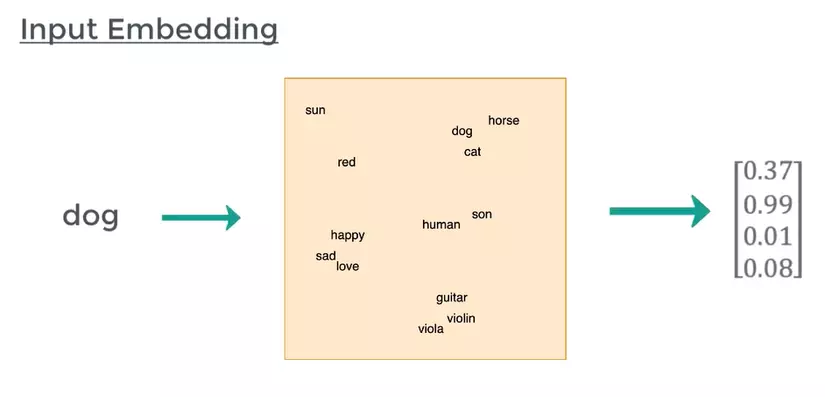
## **1.4. Giới thiệu về Transformer**

Transformer là một kiến trúc mô hình học sâu ra đời từ bài báo “Attention is All You Need” của Vaswani và cộng sự (2017). Transformer đã cách mạng hóa lĩnh vực Xử lý ngôn ngữ tự nhiên nhờ khả năng xử lý dữ liệu theo cách song song và tận dụng hiệu quả các mối quan hệ ngữ nghĩa trong văn bản.



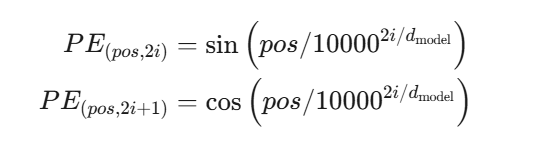
Hình 1-4: Mô hình transformer

**Input Embedding:** Máy tính không hiểu câu chữ mà chỉ đọc được số, vector, ma trận; vì vậy cần phải biểu diễn câu chữ dưới dạng vector, gọi là input embedding. Điều này đảm bảo các từ gần nghĩa có vector gần giống nhau.

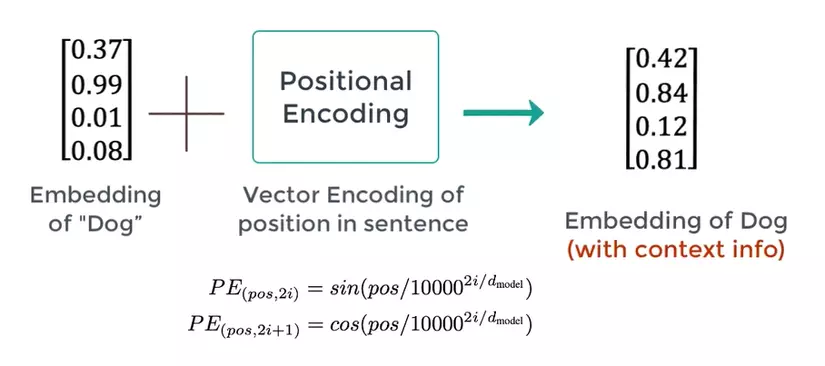
****

Hình 1-5: Hình ảnh gán nhãn đầu vào

**Positional Encoding:** Word embeddings giúp biểu diễn ngữ nghĩa của một từ, tuy nhiên cùng một từ ở vị trí khác nhau của câu lại mang ý nghĩa khác nhau. Do đó Transformers có thêm một phần Positional Encoding để inject thêm thông tin về vị trí của một từ



Trong đó *pos* là vị trí của từ trong câu, PE là giá trị phần tử thứ *i* trong embeddings có độ dài *d*model​. Sau đó cộng PE vector và Embedding vector:



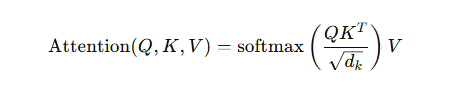
**Self-Attention:** Self-Attention là một thành phần cốt lõi giúp mô hình có khả năng tập trung vào các phần quan trọng của dữ liệu đầu vào khi xử lý thông tin, là cơ chế giúp Transformers hiểu được sự liên quan giữa các từ trong một câu, bất kể khoảng cách giữa chúng.

Cách hoạt động của self-attention

Self-Attention hoạt động bằng cách chuyển đổi dữ liệu đầu vào thành ba ma trận chính:

* **Query (Q):** Đại diện cho các từ cần tìm kiếm thông tin liên quan trong câu.
* **Key (K):** Đại diện cho các từ có khả năng cung cấp thông tin quan trọng.
* **Value (V):** Chứa nội dung thực tế của từ.

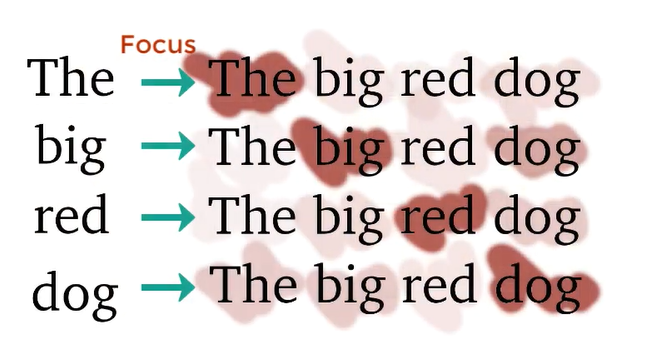
Sau đó, mô hình tính điểm mức độ liên quan giữa các từ bằng cách thực hiện phép nhân ma trận Query với Key, sau đó chia cho căn bậc hai của kích thước chiều d\_k để chuẩn hóa giá trị. Kết quả được đưa qua hàm softmax để tạo ra trọng số (attention score), rồi nhân với Value để trích xuất thông tin cần thiết:



**Multi-head Attention:** Vấn đề của Self-attention là attention của một từ sẽ luôn "chú ý" vào chính nó. Thay vì chỉ sử dụng một cặp (Q, K, V), Transformer áp dụng nhiều "head" attention (Multi-Head Attention) bằng cách chia các vector embedding thành nhiều phần nhỏ rồi thực hiện Self-Attention độc lập trên mỗi phần. Điều này giúp mô hình học được nhiều khía cạnh khác nhau của ngữ nghĩa trong dữ liệu. Mỗi **head** có trọng số riêng, giúp mô hình có khả năng nắm bắt nhiều mối quan hệ ngữ nghĩa khác nhau giữa các từ trong câu



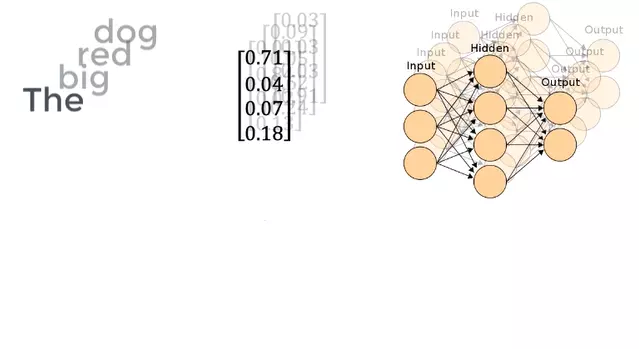
Ví dụ như sau:



Hình 1-6: Multihead Attention

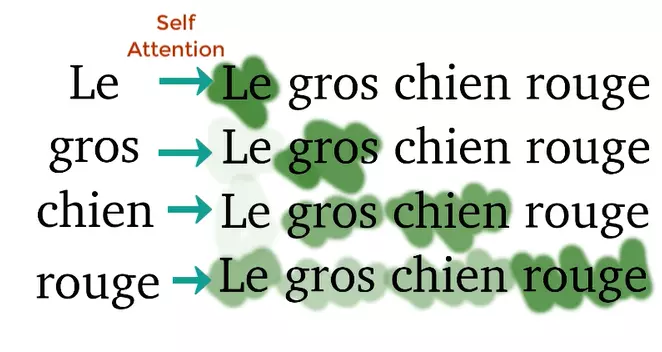
Vì mỗi "head" sẽ cho ra một ma trận attention riêng nên phải kết hợp (concatenate) các ma trận này và nhân với ma trận trọng số WO để ra một ma trận attention duy nhất (weighted sum).

**Feed Forward:** Sau khi được Normalize, các vectors z được đưa qua mạng fully connected trước khi đẩy qua Decoder. Vì các vectors này không phụ thuộc vào nhau nên có thể tận dụng được tính toán song song cho cả câu.



Hình 1-7: Feed Forward

**Masked Multi-head Attention:** Công việc của Decoder là giải mã thông tin từ Encoder và sinh ra từng từ dựa trên những từ trước đó. Vậy nên, nếu sử dụng Multi-head attention trên cả câu như ở Encoder, Decoder sẽ thấy luôn từ tiếp theo mà nó cần dự đoán. Để ngăn điều đó, khi Decoder dịch đến từ thứ i, phần sau của câu sẽ bị che lại (masked) và Decoder chỉ được phép nhìn thấy phần nó đã dịch trước đó.

****

Hình 1-8: Masked Multi-head Attention

**Quá trình decode:** Quá trình decode giống với encode, khác là Decoder decode từng từ một và input của Decoder bị masked. Sau khi masked input đưa qua sub-layer #1 của Decoder, nó sẽ không nhân với 3 ma trận trọng số để tạo ra Q, K, V mà chỉ nhân với 1 ma trận trọng số WQKV được lấy từ Encoder cùng với Q từ Masked multi-head attention đưa vào sub-layer #2 và #3 tương tự như Encoder. Cuối cùng, các vector được đẩy vào lớp Linear (là 1 mạng Fully Connected) theo sau bới Softmax để cho ra xác suất của từ tiếp theo.

Ưu điểm của Transformer:

* Khả năng xử lý song song, tăng tốc độ huấn luyện.
* Hiệu quả trong việc học mối quan hệ giữa các từ xa nhau trong văn bản dài.

## **1.5. BERT (Bidirectional Encoder Representations from Transformers)**

BERT là một mô hình dựa trên kiến trúc Transformer, được giới thiệu bởi Google vào năm 2018. Điểm nổi bật của BERT là khả năng học ngữ cảnh hai chiều (bidirectional), cho phép hiểu ý nghĩa của một từ dựa trên cả phía trước và phía sau trong câu.

BERT sử dụng Transformer là một mô hình **attention** (attention mechanism) học mối tương quan giữa các từ (hoặc 1 phần của từ) trong một văn bản. Transformer gồm có 2 phần chính: Encoder và Decoder, encoder thực hiện đọc dữ liệu đầu vào và decoder đưa ra dự đoán. Ở đây, BERT chỉ sử dụng Encoder.

Khác với các mô hình directional (các mô hình chỉ đọc dữ liệu theo 1 chiều duy nhất - trái→phải, phải→ trái) đọc dữ liệu theo dạng tuần tự, Encoder đọc toàn bộ dữ liệu trong 1 lần, việc này làm cho BERT có khả năng huấn luyện dữ liệu theo cả hai chiều, qua đó mô hình có thể học được ngữ cảnh (context) của từ tốt hơn bằng cách sử dụng những từ xung quanh nó (phải&trái).

Hình trên mô tả nguyên lý hoạt động của Encoder. Theo đó, input đầu vào là một chuỗi các token w1, w2,...được biểu diễn thành chuỗi các vector trước khi đưa vào trong mạng neural. Output của mô hình là chuỗi ccs vector có kích thước đúng bằng kích thước input. Trong khi huấn luyện mô hình, một thách thức gặp phải là các mô hình directional truyền thống gặp giới hạn khi học ngữ cảnh của từ. Để khắc phục nhược điểm của các mô hình cũ, BERT sử dụng 2 chiến lược training như sau:

**Masked LM (MLM)**

Trước khi đưa vào BERT, thì 15% số từ trong chuỗi được thay thế bởi token [MASK], khi đó mô hình sẽ dự đoán từ được thay thế bởi [MASK] với context là các từ không bị thay thế bởi [MASK]. Mask LM gồm các bước xử lý sau :

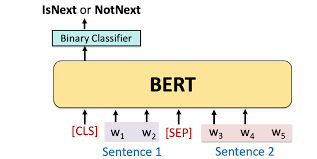
* Thêm một classification layer với input là output của Encoder.
* Nhân các vector đầu ra với ma trận embedding để đưa chúng về không gian từ vựng (vocabulary dimensional).
* Tính toán xác suất của mỗi từ trong tập từ vựng sử dụng hàm softmax.

Hàm lỗi (loss function) của BERT chỉ tập trung vào đánh giá các từ được đánh dấu [MASKED] mà bỏ qua những từ còn lại, do đó mô hình hội tụ chậm hơn so với các mô hình directional, nhưng chính điều này giúp cho mô hình hiểu ngữ cảnh tốt hơn.

**Next Sentence Prediction (NSP)**

Trong chiến lược này, thì mô hình sử dụng một cặp câu là dữ liệu đầu vào và dự đoán câu thứ 2 là câu tiếp theo của câu thứ 1 hay không. Trong quá trình huấn luyện, 50% lượng dữ liệu đầu vào là cặp câu trong đó câu thứ 2 thực sự là câu tiếp theo của câu thứ 1, 50% còn lại thì câu thứ 2 được chọn ngẫu nhiên từ tập dữ liệu. Một số nguyên tắc được đưa ra khi xử lý dữ liệu như sau:

* Chèn token [CLS] vào trước câu đầu tiên và [SEP] vào cuối mỗi câu.
* Các token trong từng câu được đánh dấu là A hoặc B.
* Chèn thêm vector embedding biểu diễn vị trí của token trong câu



Hình 1-9: Next Sentence Prediction

# **CHƯƠNG 2: QUY TRÌNH CHUẨN BỊ VÀ HUẤN LUYỆN MÔ HÌNH**

Trong chương 2 sẽ trình bày quy trình xây dựng ứng dụng trả lời văn bản hành chính, cách xây dựng tập dữ liệu, quá trình xử lý dữ liệu, tiền dữ liệu, tiến trình huấn luyện mô hình, đánh giá mô hình.

## **2.1. Tổng quan mô hình**

Trí tuệ nhân tạo nói chung và các mô hình chatbot nói riêng đang là xu thế đi đầu trong công nghệ nhằm mục đích giúp con người dễ dàng hơn trong các tác vụ thường ngày. Vì thế nhu cầu có 1 ứng dụng đóng vai trò giao tiếp và trả lời các câu hỏi của người dùng trong lĩnh vực văn bản hành chính là vô cùng quan trọng và cấp thiết. Đồ án sử dụng mô hình Transformer để ứng dụng vào mô hình giúp mô hình hiểu ngữ nghĩa trong câu, từ đó có thể giao tiếp và trả lời các câu hỏi của người dùng.

Transformer là mô hình học sâu được thiết kế để xử lý các bài toán ngôn ngữ tự nhiên, và đã tạo ra bước đột phá trong hiệu suất. Transformer giải quyết triệt để các vấn đề mà các mô hình trước đó gặp phải.

Cơ chế self-attention cho phép mô hình xem xét mối quan hệ giữa tất cả các từ trong câu cùng một lúc, không bị giới hạn bởi khoảng cách như RNN. Điều này giúp mô hình nắm bắt được các phụ thuộc xa trong văn bản một cách hiệu quả.

Kiến trúc song song của Transformer cũng cho phép huấn luyện nhanh hơn nhiều so với các mô hình tuần tự như RNN/LSTM. Thay vì phải xử lý từng từ một theo thứ tự, Transformer có thể xử lý toàn bộ câu đầu vào cùng lúc. Điều này giúp tăng tốc độ huấn luyện đáng kể và cho phép xây dựng các mô hình lớn hơn.

Cơ chế multi-head attention cho phép mô hình học nhiều kiểu mối quan hệ khác nhau giữa các từ. Mỗi head có thể tập trung vào một khía cạnh khác nhau của ngữ cảnh, ví dụ như quan hệ ngữ pháp, ngữ nghĩa hay đồng tham chiếu. Điều này giúp mô hình nắm bắt được các mối quan hệ phức tạp trong ngôn ngữ tự nhiên.

Transformer cũng giải quyết vấn đề mất mát thông tin trong các mô hình RNN/LSTM. Trong RNN, thông tin có thể bị mất đi khi truyền qua nhiều bước thời gian. Với cơ chế attention trực tiếp, Transformer luôn có thể truy cập đầy đủ thông tin từ toàn bộ câu đầu vào.

Kiến trúc Encoder-Decoder của Transformer cũng rất linh hoạt, cho phép áp dụng cho nhiều bài toán khác nhau như dịch máy, tóm tắt văn bản, trả lời câu hỏi. Encoder mã hóa thông tin từ câu đầu vào, trong khi Decoder tạo ra kết quả đầu ra dựa trên biểu diễn này.

Positional encoding trong Transformer khéo léo giải quyết việc mất thông tin về vị trí của các từ - một thách thức khi xử lý song song. Bằng cách thêm vector vị trí vào embedding của từng từ, mô hình vẫn có thể nắm bắt được thứ tự của các từ trong câu.

Các cải tiến này đã giúp Transformer đạt được hiệu suất vượt trội so với các kiến trúc trước đó trong hầu hết các bài toán NLP. Mô hình này đã trở thành nền tảng cho nhiều kiến trúc hiện đại như BERT, GPT và các mô hình ngôn ngữ lớn khác.

Ngoài ra, Transformer còn có khả năng mở rộng tốt cho các mô hình lớn. Kiến trúc của nó cho phép tăng kích thước mô hình một cách hiệu quả, dẫn đến việc phát triển các mô hình ngày càng lớn với hiệu suất ngày càng cao.

Quy trình xây dựng mô sẽ được thực hiện và mô tả qua các bước:

Bước 1: Tìm kiếm và thu thập Datasets về các câu hỏi văn bản hành chính

Bước 2: Tiền xử lý phân chia dữ liệu, thiết lập thông số thành tập Train và tập Test

Bước 3: Huấn luyện mô hình

Bước 4: Đánh giá mô hình

## **2.2. Chuẩn bị tập dữ liệu**

Trong quá trình huấn luyện mô hình AI, tập dữ liệu đóng vai trò nền tảng quyết định đến chất lượng và hiệu quả của mô hình. Một tập dữ liệu chất lượng sẽ giúp mô hình học được những mẫu chính xác, từ đó đưa ra các phản hồi phù hợp và đáp ứng tốt nhu cầu của người dùng. Ngược lại, nếu dữ liệu không đảm bảo chất lượng, mô hình có thể học những mẫu sai lệch và tạo ra các kết quả không mong muốn.

Một tập dữ liệu hoàn chỉnh cần có ba thành phần chính: Intent, Answer và Response. Intent đóng vai trò phân loại mục đích của dữ liệu, giúp mô hình hiểu được ngữ cảnh và định hướng xử lý phù hợp. Các intent phổ biến bao gồm giao tiếp thông thường, tìm kiếm thông tin, yêu cầu hướng dẫn, hoặc xác định các câu hỏi không thể trả lời. Việc phân loại intent chính xác giúp mô hình có thể đưa ra phản hồi phù hợp với mong đợi của người dùng.

Input là tập hợp các câu hỏi mẫu mà người dùng có thể đặt ra. Phần này cần đảm bảo sự đa dạng trong cách diễn đạt cho cùng một ý định, phản ánh chân thực cách người dùng thường hỏi trong thực tế. Điều quan trọng là phải bao gồm đủ các trường hợp ngoại lệ và các tình huống đặc biệt để mô hình có thể xử lý linh hoạt các câu hỏi từ người dùng.

Response là phần quan trọng cuối cùng, bao gồm các câu trả lời mẫu mà mô hình sẽ học để phản hồi. Các câu trả lời này cần đảm bảo tính chính xác, đa dạng trong cách diễn đạt, và phải thực sự hữu ích trong việc giải quyết vấn đề của người dùng. Ngoài ra, các phản hồi cũng cần phù hợp với ngữ cảnh sử dụng của người dùng.

Để xây dựng được một tập dữ liệu chất lượng, cần đảm bảo ba yếu tố: tính đầy đủ, tính sạch và tính chất lượng. Tính đầy đủ thể hiện ở việc dữ liệu phải bao phủ được đủ các trường hợp sử dụng, đa dạng về cách diễn đạt và cân bằng giữa các loại intent khác nhau. Tính sạch đòi hỏi dữ liệu phải không có lỗi chính tả, đúng ngữ pháp và nhất quán trong cách ghi nhận. Tính chất lượng được thể hiện qua việc thông tin phải chính xác, cập nhật và phù hợp với mục tiêu sử dụng.

Dữ liệu được thu thập từ các văn bản hành chính, cụ thể trong đồ án là văn bản về pháp luật năm 2025 lấy từ trang thuvienphapluat.

Ví dụ:

|  |  |  |
| --- | --- | --- |
| Intent | Input | Response |
| Trò chuyện | Xin chào | Chào bạn! Bạn hãy đặt câu hỏi để tôi có thể trả lời bạn! |
| Trò chuyện | ??? | Xin lỗi, tôi không hiểu ý bạn. Bạn có thể nói rõ hơn được không? |
| Không thể trả lời | World Cup 2026 sẽ tổ chức ở đâu? | Xin lỗi, tôi chỉ hỗ trợ các câu hỏi liên quan đến luật giao thông. Bạn cần hỏi gì về luật giao thông? |
| Không thể trả lời | Cách chăm sóc cây cảnh trong nhà? | Xin lỗi, tôi không biết câu trả lời. Xin hãy hỏi câu hỏi khác! |
| Luật giao thông | Các biện pháp xử phạt khi vi phạm giao thông đường bộ là gì? | Đối với mỗi hành vi vi phạm hành chính về trật tự, an toàn giao thông trong lĩnh vực giao thông đường bộ, cá nhân, tổ chức vi phạm phải chịu một trong các hình thức xử phạt chính như cảnh cáo hoặc phạt tiền. Ngoài ra, có thể áp dụng các hình thức xử phạt bổ sung như tước quyền sử dụng giấy phép lái xe, chứng chỉ bồi dưỡng kiến thức pháp luật về giao thông đường bộ có thời hạn; tịch thu tang vật, phương tiện được sử dụng để vi phạm hành chính. |
| Luật giao thông | Lái xe ô tô gây tai nạn mà không trình báo cơ quan chức năng sẽ bị xử lý ra sao? | Người điều khiển xe ô tô gây tai nạn giao thông nhưng không dừng lại, không giữ nguyên hiện trường, không trợ giúp người bị nạn, hoặc không trình báo cơ quan chức năng sẽ bị phạt tiền từ 16.000.000 đồng đến 18.000.000 đồng. |

## **2.3. Tiền xử lý dữ liệu**

Trong bước tiền xử lý, dữ liệu văn bản thường được chuẩn hóa về chữ thường, loại bỏ các ký tự đặc biệt và dấu câu không cần thiết. Quá trình tokenization được thực hiện để tách văn bản thành các đơn vị nhỏ hơn như từ hoặc subword. Đối với tiếng Việt, việc tách từ (word segmentation) đặc biệt quan trọng do đặc thù của ngôn ngữ. Ngoài ra, các stop words - những từ xuất hiện thường xuyên nhưng không mang nhiều ý nghĩa - cũng có thể được loại bỏ.

Tăng cường dữ liệu (data augmentation) là kỹ thuật quan trọng giúp mở rộng kích thước tập dữ liệu và tăng tính đa dạng của dữ liệu huấn luyện. Đối với dữ liệu văn bản, có nhiều phương pháp tăng cường khác nhau. Phương pháp thay thế từ đồng nghĩa (synonym replacement) sử dụng các từ điển đồng nghĩa để tạo ra các câu có cùng ý nghĩa nhưng cách diễn đạt khác nhau. Back translation là kỹ thuật dịch văn bản sang ngôn ngữ khác rồi dịch ngược lại, tạo ra các biến thể mới nhưng vẫn giữ được ý nghĩa gốc.

Một phương pháp tăng cường dữ liệu hiệu quả khác là hoán đổi vị trí từ (word swap) trong phạm vi cho phép mà không làm thay đổi ý nghĩa của câu. Kỹ thuật chèn từ ngẫu nhiên (random insertion) và xóa từ ngẫu nhiên (random deletion) cũng được sử dụng để tạo ra các biến thể mới của câu, nhưng cần được thực hiện cẩn thận để đảm bảo tính nhất quán về ngữ nghĩa.

Trong quá trình tăng cường dữ liệu, việc kiểm soát chất lượng các dữ liệu được tạo ra là vô cùng quan trọng. Cần đảm bảo các câu được tạo ra vẫn giữ được ý nghĩa gốc, đúng ngữ pháp và phù hợp với ngữ cảnh. Đồng thời, cần tránh tạo ra quá nhiều dữ liệu tăng cường giống nhau, điều này có thể dẫn đến hiện tượng overfitting trong quá trình huấn luyện mô hình.

Việc theo dõi và đánh giá chất lượng của dữ liệu sau khi qua các bước tiền xử lý này là rất quan trọng, giúp đảm bảo tập dữ liệu cuối cùng đạt được các yêu cầu về chất lượng và đa dạng cho việc huấn luyện mô hình.

Ví dụ

|  |  |  |  |
| --- | --- | --- | --- |
| Văn bản gốc | Tokens | Token IDs | Decoded |
| Những ai thuộc đối tượng của nghị định này? | Những ai thuộc đối tượng của nghị định này@@ ? | 0 217 277 187 3959 2634 7 12813 2608 45110 114 2 | <s> Những ai thuộc đối tượng của nghị định này@@ ? </s> |
| Các hành vi liên quan đến giao thông nhưng không được đề cập tại Nghị định này sẽ xử lý ra sao? | Các hành vi liên quan đến giao thông nhưng không được đề cập tại Nghị định này sẽ xử lý ra sao@@ ? | 0 146 2537 5237 2657 2665 30 574 2178 51 17 11 1294 5530 35 10658 2608 23 38 3224 3902 40 42524 114 2 | <s> Các hành vi liên quan đến giao thông nhưng không được đề cập tại Nghị định này sẽ xử lý ra sao@@ ? </s> |
| Cơ quan nhà nước có nằm trong phạm vi áp dụng không? | Cơ quan nhà nước có nằm trong phạm vi áp dụng khô@@ ng@@ ? | 0 8917 2665 69 58 10 339 12 2283 5237 2925 8410 32628 1701 114 2 | <s> Cơ quan nhà nước có nằm trong phạm vi áp dụng khô@@ ng@@ ? </s> |

## **2.4. Phân chia dữ liệu**

Sau khi có được tệp dữ liệu tiến hành chia tất cả dữ liệu đã thu thập thành các tập hợp con, quá trình này được gọi là phân tách dữ liệu. Dữ liệu được chia nhỏ thành tập dữ liệu Train - đào tạo, Validation - xác thực và Test - thử nghiệm.

Về cơ bản, có nhiều cách phân chia các tập dữ liệu khác nhau phụ thuộc vào số lượng dữ liệu quan sát được cung cấp và mô hình thực tế được huấn luyện. Có 3 cách chia tập dữ liệu thường được sử dụng trên thực tế đó là:

* Hold-out: Đây là phương pháp chỉ chia tập dữ liệu quan sát được cung cấp thành hai tập con, trong đó một tập để huấn luyện (thường chiếm 80% lượng dữ liệu hoặc 70% dữ liệu) và một tập được bỏ ra (Hold-out) ngoài tập huấn luyện đóng vai trò là testing set, là phần còn lại của bộ dữ liệu được cung cấp (thường là 20% lượng dữ liệu hoặc 30% dữ liệu còn lại).
* Train-Validation-Test Split: Đây là phương pháp có sử dụng bộ dữ liệu validation set. Cách thức chia tương tự như trường hợp Hold-out, tức là để riêng ra một bộ dữ liệu để kiểm thử (testing set), và trong lượng dữ liệu còn lại lại chia nhỏ thành 2 phần, 1 phần để train mô hình (thường khoảng 80% lượng dữ liệu còn lại), và một phần để kiểm chứng và hiệu chỉnh mô hình (validation – 20% dữ liệu còn lại).
* Cross Validation: là phương pháp chủ yếu được sử dụng để đánh giá thuật toán huấn luyện hoặc dạng của mô hình huấn luyện chứ không phải mô hình huấn luyện. Khi đó, quá trình huấn luyện và kiểm thử sẽ được thực hiện lặp đi lặp lại nhiều lần. Trong mỗi lần tập dữ liệu quan sát sẽ được lần lượt chia thành 2 phần (training và testing), phần training được sử dụng cho việc huấn luyện, còn phần testing được sử dụng cho kiểm thử.

Trong bài toán nhận diện biển báo giao thông tập dữ liệu không quá lớn, để đảm bảo dữ liệu cho huấn luyện được đầy đủ và cân bằng nên dữ liệu sẽ được phân chia theo tỉ lệ 60/20/20: Trong đó  tập Train và tập Validation chiếm 80% và tập Test chiếm 20% tập dữ liệu.

* Dữ liệu Train lớn nhất vì đây là tập dùng cho huấn luyện mô hình
* Dữ liệu Validation dùng cho đánh giá chất lượng của mô hình, và từ đó có các quyết định thay đổi liên quan đến mô hình (thuật toán học, siêu tham số, đặc trưng, ...)
* Dữ liệu Test dùng cho đánh giá chất lượng của mô hình (nhưng không được dùng để quyết định các thay đổi liên quan đến mô hình - như vậy là gián tiếp tham gia huấn luyện mô hình)
* Kích thước dữ liệu Validation thường bằng dữ liệu test để dữ liệu Validation có thể mô phỏng được dữ liệu test (nếu Validation và tập Test quá khác nhau thì việc hiệu chỉnh mô hình để tốt với Validation không đảm bảo mô hình tốt với Test, như vậy thì Validation sinh ra cũng không có tác dụng)

## **2.5. Huấn luyện mô hình**

Sau khi đã có dữ liệu tiếp tục sử dụng Transformers để thiết lập quá trình huấn luyện mô hình. Các tham số huấn luyện được thiết lập bao gồm:

* **num\_train\_epochs**: Số vòng lặp (epochs) huấn luyện trên tập dữ liệu. Giá trị được đặt là **4** để đảm bảo quá trình huấn luyện vừa đủ nhanh mà không gây quá tải mô hình.
* **learning\_rate**: Tốc độ học của mô hình, được đặt là **3e-5**, một giá trị phổ biến và phù hợp cho các mô hình Transformer.
* **weight\_decay**: Tham số để giảm overfitting, với giá trị **0.01**.
* **per\_device\_train\_batch\_size**: Số mẫu trong mỗi batch cho mỗi thiết bị (GPU/CPU) khi huấn luyện, đặt là **4**.
* **gradient\_accumulation\_steps**: Tổng hợp gradient qua 2 bước trước khi cập nhật, tăng hiệu quả khi sử dụng batch nhỏ.

Trong quá trình huấn luyện mô hình, giám sát và điều chỉnh là những bước quan trọng để đảm bảo mô hình hoạt động hiệu quả, tránh các vấn đề như quá khớp, không hội tụ hoặc lãng phí tài nguyên. Trước tiên, cần theo dõi các chỉ số hiệu suất như hàm mất mát (loss) trên tập huấn luyện và tập kiểm tra, cũng như các chỉ số đánh giá cụ thể như Accuracy, F1-score. Nếu loss trên tập huấn luyện giảm nhưng tăng trên tập kiểm tra, đây là dấu hiệu của quá khớp, đòi hỏi các kỹ thuật như Dropout, Weight Decay hoặc tăng cường dữ liệu (Data Augmentation). Ngược lại, nếu cả hai giá trị loss đều cao, mô hình có thể đang underfit, cần được tăng độ phức tạp hoặc cải thiện dữ liệu.

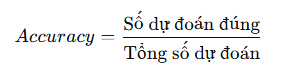
Điều chỉnh siêu tham số như learning rate, batch size, số epoch cũng là yếu tố quan trọng. Learning rate quá lớn khiến mô hình dao động và không hội tụ, trong khi learning rate quá nhỏ làm chậm quá trình học. Sử dụng các kỹ thuật như Early Stopping để dừng huấn luyện khi hiệu suất trên tập kiểm tra không cải thiện sau một số epoch nhất định giúp tiết kiệm thời gian và tránh overfitting. Đồng thời, việc kiểm tra gradient trong quá trình lan truyền ngược giúp phát hiện và xử lý các vấn đề như vanishing gradient hoặc exploding gradient bằng cách áp dụng các kỹ thuật như gradient clipping.

Ngoài ra, cần giám sát thời gian huấn luyện và tài nguyên, đảm bảo batch size phù hợp với bộ nhớ GPU. Các công cụ như TensorBoard hoặc Weights & Biases hỗ trợ trực quan hóa quá trình huấn luyện và quản lý thí nghiệm. Cuối cùng, kiểm tra và làm sạch dữ liệu, vì dữ liệu không đầy đủ hoặc chứa lỗi có thể làm giảm hiệu suất mô hình. Bằng cách kết hợp giám sát chặt chẽ với điều chỉnh hợp lý, mô hình có thể hội tụ nhanh hơn, đạt hiệu suất cao và khả năng tổng quát hóa tốt.

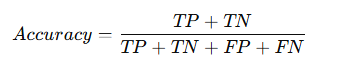
## **2.6. Đánh giá kết quả**

**ACCURACY**

Accuracy là một thuật ngữ được sử dụng trong ngữ cảnh của máy học và học máy để mô tả mức độ đúng đắn của mô hình trong việc dự đoán hay phân loại dữ liệu. Chính xác thường được tính bằng tỷ lệ giữa số lượng dự đoán đúng và tổng số lượng dự đoán. Công thức tính chính xác như sau:



Trong bài toán phân loại nhị phân, công thức này có thể biểu diễn rõ hơn bằng cách sử dụng các giá trị từ ma trận nhầm lẫn (Confusion Matrix):



Điều này đo lường khả năng của mô hình trong việc đưa ra dự đoán chính xác. Accuracy là một phép đo phổ quát và dễ hiểu, nhưng có nhược điểm khi mô hình phải đối mặt với dữ liệu mất cân bằng (imbalance), nơi mà một số lớp xuất hiện nhiều hơn so với các lớp khác. Trong trường hợp này, một mô hình có thể có accuracy cao mà vẫn không thực sự hiệu quả.

**LOSS**

Loss (hàm mất mát) là một khái niệm cốt lõi trong học máy và deep learning, đại diện cho sự khác biệt giữa giá trị dự đoán của mô hình và giá trị thực tế. Mục tiêu của việc tối ưu hóa mô hình là giảm giá trị của hàm mất mát này để cải thiện hiệu suất dự đoán.



Các thuật ngữ khác như "Cost" hoặc "Error" cũng thường được sử dụng thay thế cho "loss". Loss Function (hàm mất mát) chịu trách nhiệm tính toán giá trị mất mát dựa trên dự đoán của mô hình và giá trị thực tế. Cụ thể, loss function đo lường sự khác biệt giữa giá trị dự đoán và giá trị thực tế của mỗi mẫu dữ liệu trong tập huấn luyện.

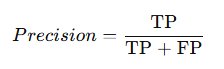
Mục tiêu của quá trình huấn luyện mô hình là tối ưu hóa hàm mất mát tức là giảm thiểu giá trị loss. Quá trình này thường được thực hiện thông qua các thuật toán tối ưu hóa Gradient Descent hoặc các biến thể của nó.

Một số hàm mất mát phổ biến bao gồm:

* Mean Squared Error (MSE): Đo lường trung bình bình phương của sự chênh lệch giữa dự đoán và giá trị thực tế.
* Cross-Entropy Loss (Log Loss): Phổ biến trong các bài toán phân loại, đo lường sự chênh lệch giữa phân phối xác suất dự đoán và phân phối xác suất thực tế.
* Hinge Loss: Thường được sử dụng trong các bài toán Support Vector Machines (SVM) và các bài toán phân loại nhị phân.

**PRECISION**

Precision là tỷ lệ giữa số sample được tính là True Positive (TP) với tổng số sample được phân loại là Positive (bằng chính TP + FP)

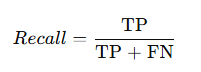


Precision đặc biệt quan trọng trong các bài toán mà FP (False Positive) gây ra hậu quả nghiêm trọng.

Và khi 0< Precision <=1, Precision càng lớn có nghĩa là độ chính xác của các điểm tìm được càng cao.

**RECALL**

Recall là chỉ số đo lường khả năng của mô hình trong việc phát hiện chính xác các mẫu dương tính trong toàn bộ dữ liệu.



Recall quan trọng trong các bài toán mà FN (False Negative) gây ra hậu quả nghiêm trọng.

**F-1 SCORE**

F1-Score là một thước đo kết hợp giữa Precision và Recall. Đặc biệt F1-Score được sử dụng trong trường hợp dữ liệu mất cân bằng, khi số lượng mẫu thuộc các lớp khác nhau không đồng đều. F1-Score đặc biệt hữu ích khi dữ liệu không cân bằng và cần đánh giá hiệu suất tổng quát.



F1-score tự nhiên có giá trị trong khoảng từ 0 đến 1, với 1 là giá trị tốt nhất (đạt được khi cả Precision và Recall đều bằng 1). F1-score giúp cân bằng giữa Precision và Recall, giúp đánh giá hiệu suất của mô hình một cách tổng thể, đặc biệt trong trường hợp có sự mất cân bằng giữa các lớp.

# **CHƯƠNG 3: TRIỂN KHAI VÀ THỰC NGHIỆM**

Trong chương này trình bày về quy trình triển khai thực nghiệm mô hình và kết quả sau khi tích hợp mô hình.

## **3.1. Mô tả bài toán**

Văn bản hành chính đóng vai trò thiết yếu trong hoạt động của các cơ quan, tổ chức nhà nước và đơn vị hành chính các cấp. Đây là công cụ chính thức được sử dụng để truyền đạt chỉ thị, quyết định, hướng dẫn và các thông tin quản lý từ cấp trên xuống cấp dưới, cũng như báo cáo và phản hồi từ cấp dưới lên cấp trên. Các văn bản này bao gồm nhiều loại như nghị định, thông tư, quyết định, công văn, và được sử dụng trong mọi lĩnh vực từ kinh tế, văn hóa đến xã hội, góp phần đảm bảo tính thống nhất và hiệu quả trong công tác quản lý nhà nước.

Tuy nhiên, việc theo dõi và nắm bắt nội dung các văn bản hành chính đang gây nhiều khó khăn đối với người dân. Thứ nhất, số lượng văn bản được ban hành ngày càng nhiều, với tần suất cập nhật và thay đổi liên tục để đáp ứng yêu cầu quản lý trong thời đại mới. Thứ hai, nội dung các văn bản thường mang tính chuyên môn cao, sử dụng nhiều thuật ngữ đặc thù và có cấu trúc phức tạp, gây khó khăn cho người dân trong việc tìm hiểu và áp dụng. Thứ ba, sự phân tán của các văn bản trên nhiều kênh thông tin khác nhau cũng làm tăng thêm độ phức tạp trong việc tra cứu và tổng hợp thông tin.

Trước những khó khăn này, việc xây dựng một ứng dụng thông minh hỗ trợ tra cứu và tìm hiểu văn bản hành chính là một giải pháp cấp thiết và có ý nghĩa thực tiễn cao. Ứng dụng được đề xuất sẽ tận dụng công nghệ trí tuệ nhân tạo để xây dựng hệ thống hỏi đáp tương tác, cho phép người dùng đặt câu hỏi bằng ngôn ngữ tự nhiên và nhận được câu trả lời chính xác, dễ hiểu về nội dung văn bản hành chính. Hệ thống này không chỉ giúp đơn giản hóa quá trình tìm kiếm thông tin mà còn có khả năng giải thích các nội dung phức tạp theo cách dễ hiểu với người dân.

Hơn nữa, việc tích hợp ứng dụng này vào hệ thống công nghệ thông tin tại các cơ quan hành chính cơ sở như Ủy ban nhân dân xã sẽ mang lại nhiều lợi ích thiết thực. Người dân có thể dễ dàng tra cứu thông tin về thủ tục hành chính, chính sách mới, quyền lợi và nghĩa vụ của mình mà không cần phải trực tiếp đến cơ quan nhà nước hoặc tìm kiếm qua nhiều nguồn khác nhau. Điều này không chỉ tiết kiệm thời gian và công sức cho người dân mà còn góp phần giảm tải cho đội ngũ cán bộ, công chức trong việc giải đáp thắc mắc và hướng dẫn người dân.

Chức năng chính của ứng dụng là trả lời các câu hỏi về nội dung văn bản hành chính. Người dùng sẽ vào ứng dụng để bắt đầu giao tiếp với mô hình chatbot. Mô hình sẽ nhận đầu vào và đưa ra kết quả hợp lý nhất cho người dùng.

## **3.2. Công nghệ sử dụng**

### **3.2.1. Python**

Python là một ngôn ngữ lập trình bậc cao, được thiết kế với mục tiêu đơn giản, dễ đọc và dễ học. Được tạo ra bởi **Guido van Rossum** vào năm 1991, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất thế giới. Python hỗ trợ nhiều mô hình lập trình khác nhau, bao gồm lập trình hướng đối tượng, lập trình thủ tục, và lập trình hàm, giúp nó linh hoạt trong việc giải quyết các vấn đề đa dạng.



Hình 1-10: Logo Python

Với cú pháp đơn giản, Python dễ đọc và gần gũi với ngôn ngữ tự nhiên, giúp các lập trình viên viết code nhanh chóng và dễ bảo trì. Một đặc điểm nổi bật khác của Python là **thư viện tiêu chuẩn phong phú**, cung cấp các công cụ hỗ trợ từ xử lý văn bản, làm việc với hệ thống tệp, đến phát triển web và học máy. Python cũng hỗ trợ rất nhiều thư viện và framework phổ biến như **NumPy**, **Pandas**, **Django**, **Flask**, và **TensorFlow**, giúp tăng cường hiệu quả và mở rộng khả năng của ngôn ngữ.

Python được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

* **Khoa học dữ liệu và học máy**: Nhờ các thư viện mạnh mẽ như Scikit-learn, PyTorch, và TensorFlow.
* **Phát triển web:** Với các framework như Django và Flask.
* **Tự động hóa:** Giúp tự động hóa các tác vụ lặp đi lặp lại.
* **Phát triển ứng dụng:** Từ ứng dụng nhỏ đến hệ thống lớn.
* **IoT và trí tuệ nhân tạo:** Python là lựa chọn hàng đầu trong việc xây dựng các giải pháp thông minh.

### **3.2.2. Nuxt**

**Nuxt.js** là một framework dựa trên **Vue.js**, được thiết kế để giúp việc xây dựng các ứng dụng web hiện đại trở nên dễ dàng hơn, đặc biệt là các ứng dụng **Universal (SSR - Server-Side Rendering)** hoặc **Single Page Applications (SPA)**. Được phát triển bởi một cộng đồng mở và các nhà phát triển từ đội ngũ Nuxt, framework này kết hợp các công nghệ mạnh mẽ, tối ưu hóa hiệu suất, và cung cấp nhiều tính năng hữu ích để tạo ra các ứng dụng web chất lượng cao.



Hình 1-11: Logo Nuxt

**Đặc điểm nổi bật của Nuxt.js:**

* **Hỗ trợ Server-Side Rendering (SSR):** SSR giúp cải thiện hiệu suất tải trang và tăng khả năng SEO (Search Engine Optimization), làm cho ứng dụng dễ dàng được tìm thấy bởi các công cụ tìm kiếm.
* **Static Site Generation (SSG):** Nuxt.js cho phép tạo các trang tĩnh (static websites) từ ứng dụng Vue, sử dụng nuxt generate. Điều này rất hữu ích cho việc xây dựng các trang web nhanh, nhẹ và thân thiện với SEO.
* **Cấu trúc thư mục rõ ràng:** Nuxt.js tự động quản lý router dựa trên cấu trúc thư mục pages/. Người dùng không cần tự định nghĩa các route, giúp code dễ đọc và quản lý hơn.
* **Hệ sinh thái phong phú:** Hỗ trợ các tính năng như quản lý trạng thái qua Vuex, tích hợp plugin, module hóa, và các tiện ích như Axios, PWA, TailwindCSS, và nhiều hơn nữa.
* **Hỗ trợ Typescript:** Nuxt.js cung cấp khả năng tích hợp Typescript để tăng tính an toàn và khả năng kiểm tra lỗi trong ứng dụng.
* **Tích hợp dễ dàng với các công cụ khác:** Nuxt.js hỗ trợ tích hợp liền mạch với các công cụ như Google Analytics, Sentry, Contentful, giúp phát triển ứng dụng nhanh chóng và thuận tiện hơn.

**Ứng dụng của Nuxt.js:**

* Phát triển các trang web cần SEO cao (như blog, trang thương mại điện tử).
* Xây dựng các hệ thống quản lý nội dung (CMS).
* Tạo các ứng dụng SPA hoặc Progressive Web Apps (PWAs).
* Làm các dự án cần khả năng mở rộng và tối ưu hiệu suất.

## **3.3. Phân tích thiết kế hệ thống**

### **3.3.1. Yêu cầu chức năng**

Ứng dụng trả lời câu hỏi văn bản hành chính phải đáp ứng các yêu cầu cơ bản sau:

* Nhận diện chủ đề và ý định của người dùng.
* Xử lý các câu hỏi khác nhau cho cùng một vấn đề.
* Nội dung văn bản hành chính phải chính xác, cập nhật liên tục.
* Giao diện thân thiện, dễ sử dụng.
* Thời gian phản hồi nhanh, đảm bảo yếu tố giao tiếp thời gian thực.

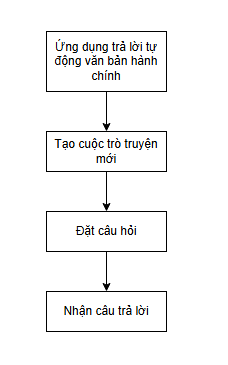
### **3.3.2. Kiến trúc tổng quan**

Hệ thống được thiết kế theo mô hình **client-server**, trong đó frontend (Nuxt.js) chịu trách nhiệm hiển thị giao diện và tương tác với người dùng, còn backend (Flask) đảm nhận xử lý câu hỏi và trả lời. Frontend giao tiếp với backend thông qua API REST, khi nhận được câu hỏi mô hình sẽ xử lý và gửi phản hồi lại câu trả lời về frontend để hiện thị cho người dùng

Hệ thống hoạt động theo quy trình sau:

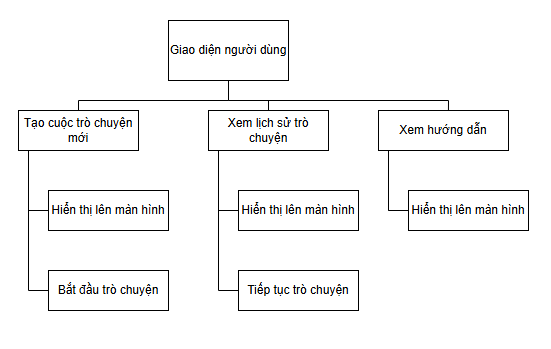
* Người dùng nhập câu hỏi trên giao diện.
* Gửi yêu cầu chứa câu hỏi đến API của Flask thông qua phương thức HTTP (RESTful API).
* Flask tiếp nhận yêu cầu và thực hiện xử lý.
* Sau khi xử lý hoàn tất, Flask gửi câu trả lời về frontend.
* Frontend hiển thị kết quả trên giao diện.

### **3.3.3. Mô hình tổng quan chức năng ứng dụng**

****

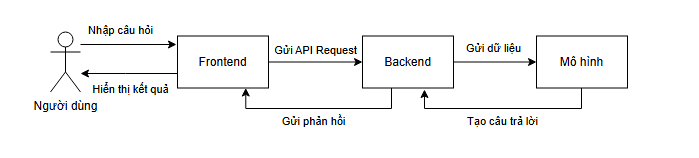
Hình 3-1: Mô hình tổng quan chức năng ứng dụng

### **3.3.4. Mô hình phân rã chức năng**

****

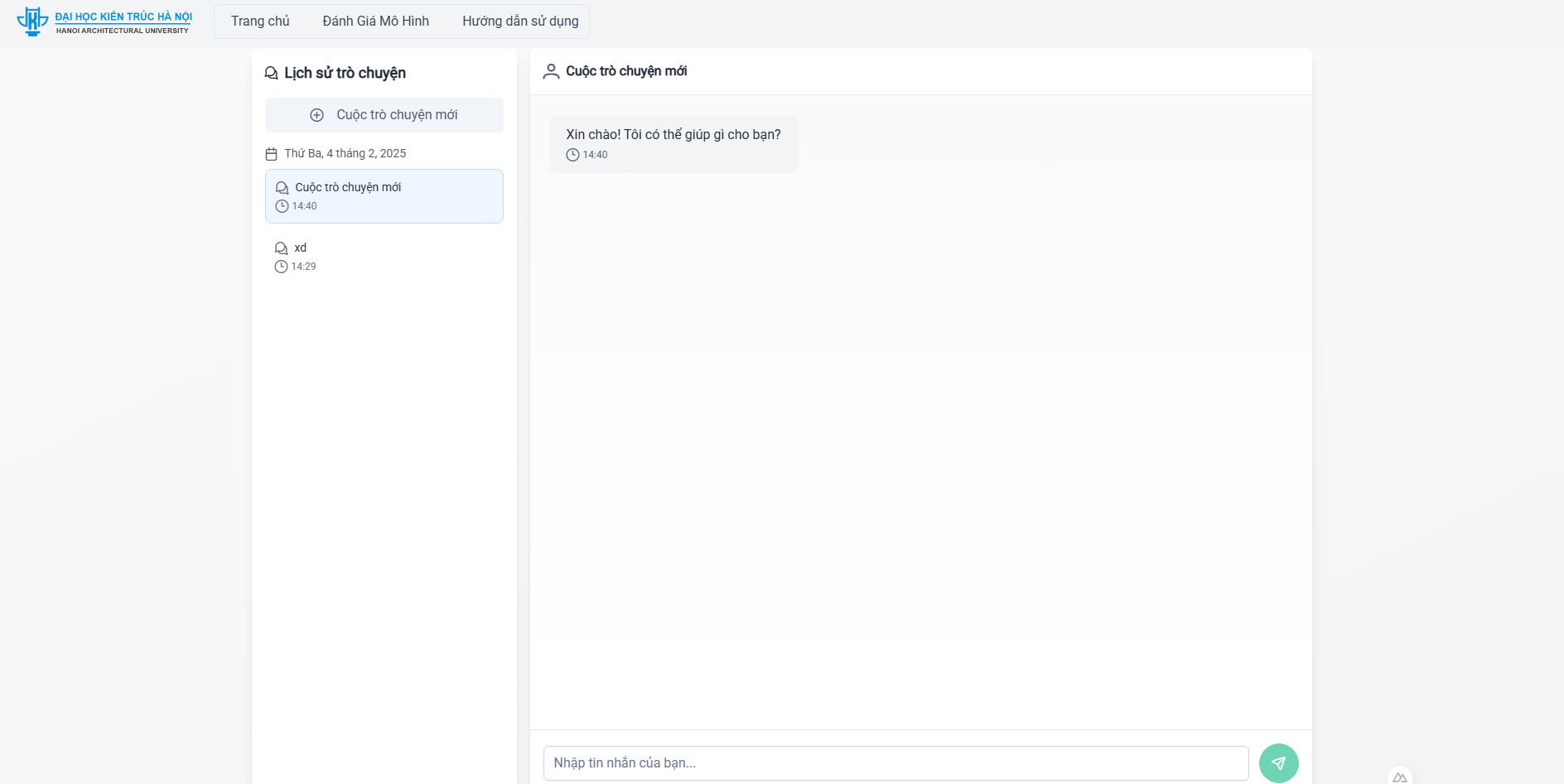
Hình 3-2: Mô hình chức năng ứng dụng

**3.3.5. Biểu đồ luồng dữ liệu**

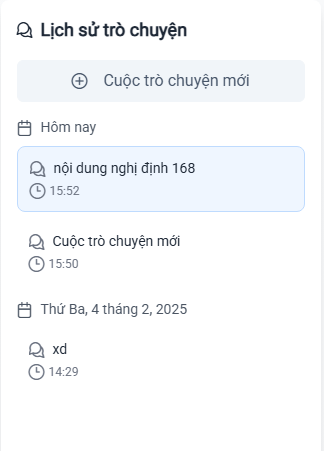
****

Hình 3-3: Biểu đồ luồng dữ liệu

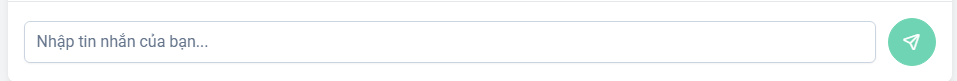
### **3.3.6. Giao diện ứng dụng**

****

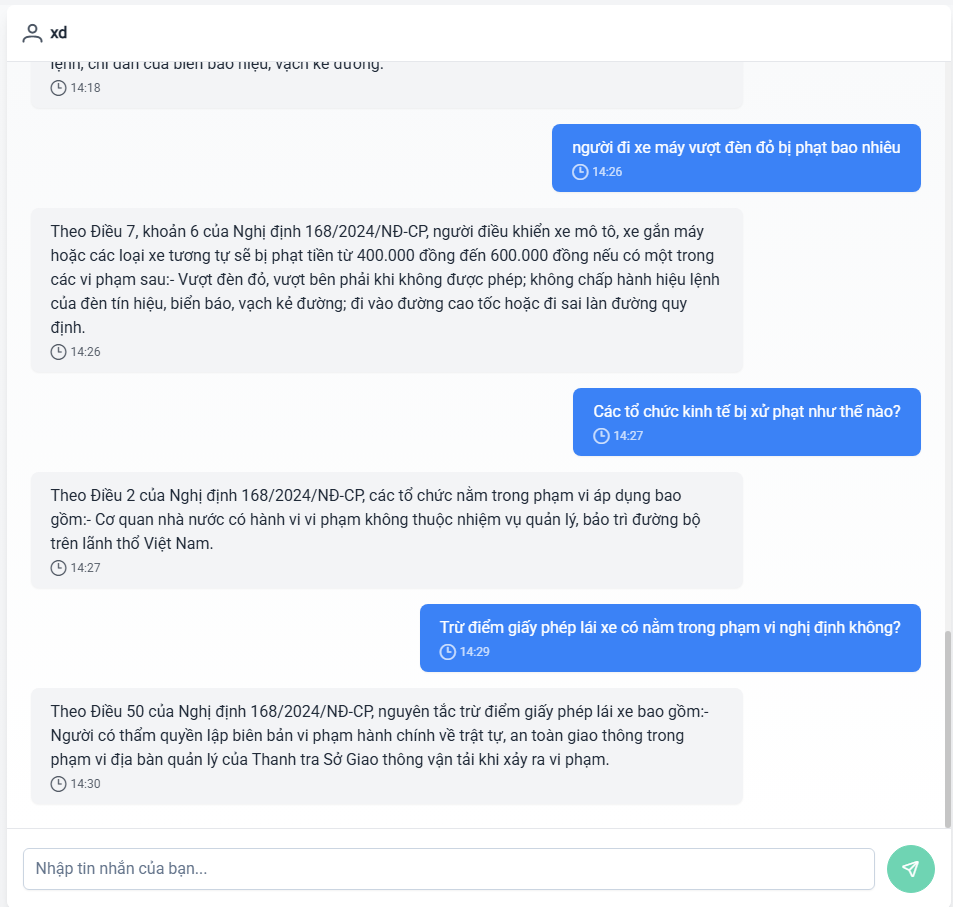
Hình 3-3: Giao diện ban đầu của ứng dụng

****

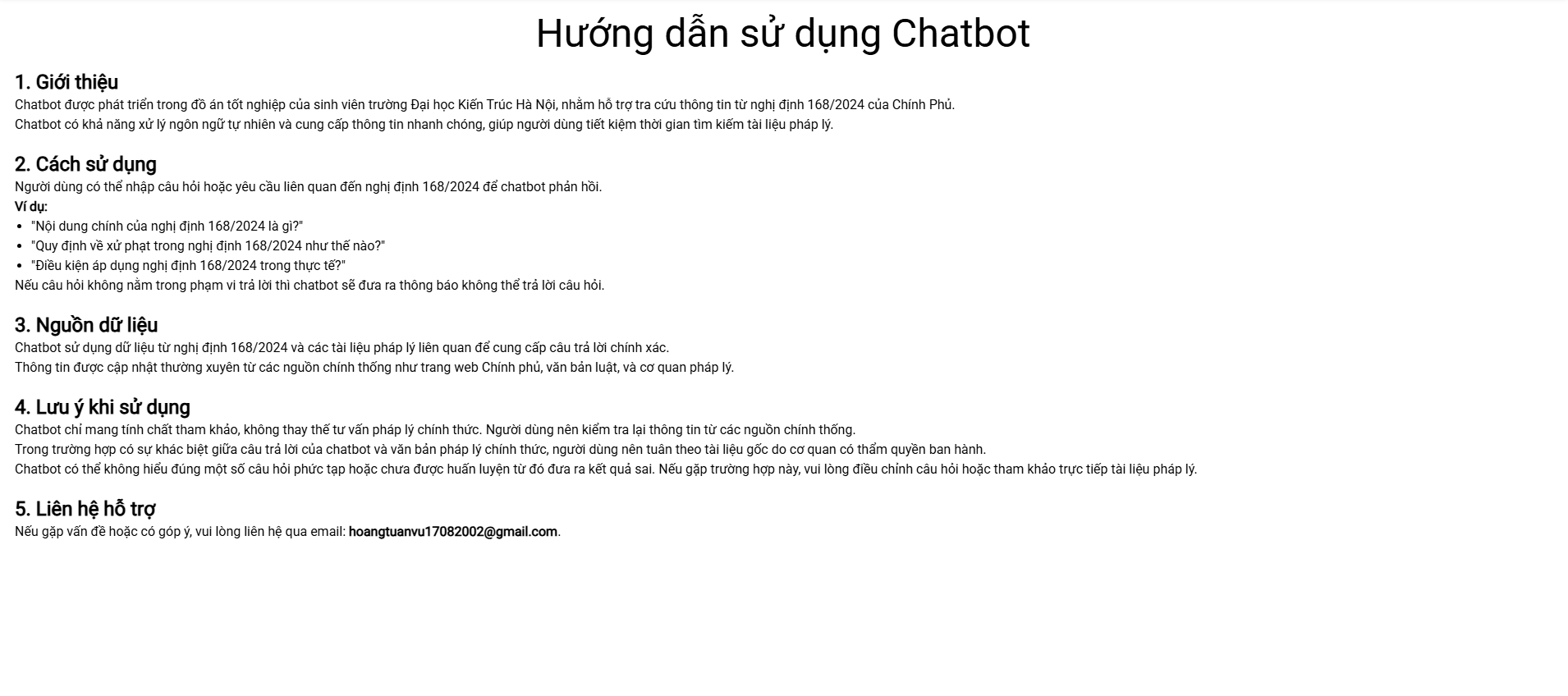
Hình 3-4: Thanh lịch sử trò chuyện

****

Hình 3-5: Thanh trò chuyện dùng để người dùng đặt câu hỏi

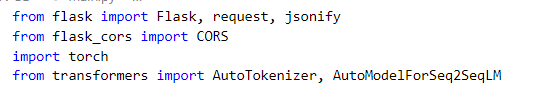
****

Hình 3-6: Giao diện cuộc trò chuyện của ứng dụng



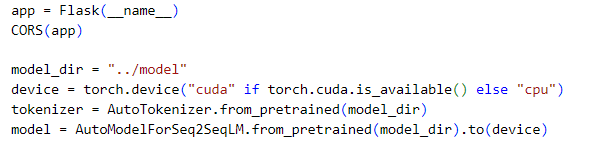
Hình 3-7: Giao diện hướng dẫn sử dụng

## **3.4. Quy trình thực nghiệm**



Import các thư viện cần thiết

* Flask: Framework giúp xây dựng ứng dụng web nhanh chóng trong Python.
* request: Dùng để lấy dữ liệu từ yêu cầu HTTP.
* jsonify: Chuyển đổi dữ liệu Python thành JSON để trả về từ API.
* CORS: Bật chính sách CORS (Cross-Origin Resource Sharing) cho phép ứng dụng web khác domain truy cập API.
* torch: Thư viện dành cho tính toán tensor.
* transformers: Thư viện từ Hugging Face để làm việc với mô hình.



**app = Flask(name):** Tạo một ứng dụng Flask mới.

**CORS(app):** Bật CORS cho toàn bộ các route, cho phép ứng dụng từ các domain khác truy cập vào API này.

**torch.device():** Kiểm tra xem GPU có khả dụng không; nếu có, sẽ sử dụng GPU, nếu không sẽ dùng CPU.

**tokenizer:** Tải tokenizer từ thư mục chứa mô hình. Tokenizer chuyển đổi văn bản thành các token mà mô hình hiểu được.

**model:** Tải mô hình đã fine-tune và đưa nó vào thiết bị (CPU hoặc GPU).



**@app.route('/predict', methods=['POST']):** Định nghĩa route /predict, chấp nhận phương thức HTTP POST.

**def predict():** Hàm xử lý yêu cầu tại route này.

**request.get\_json():** Lấy dữ liệu JSON từ yêu cầu.

**data.get('input'):** Lấy giá trị của trường input từ JSON.

**if not input\_text:** Kiểm tra xem input\_text có tồn tại không; nếu không, trả về lỗi với mã trạng thái HTTP 400.

**tokenizer(...):** Chuyển văn bản đầu vào thành tensor (dạng số).

* return\_tensors="pt": Yêu cầu trả về dữ liệu dưới dạng tensor của PyTorch.
* padding=True: Tự động thêm padding nếu cần để các chuỗi có cùng độ dài.
* truncation=True: Cắt ngắn chuỗi nếu dài hơn max\_length.
* max\_length=128: Giới hạn số token tối đa là 128.

**to(device):** Chuyển dữ liệu đầu vào lên thiết bị (CPU hoặc GPU).

**model.generate(...)**: Sinh câu trả lời từ mô hình dựa trên dữ liệu đầu vào.

* input\_ids: Token ID của chuỗi đầu vào.
* attention\_mask: Gắn nhãn để mô hình biết token nào cần chú ý.
* num\_beams=4: Sử dụng beam search với 4 beams để tìm câu trả lời tốt nhất.
* length\_penalty=1.0: Hình phạt dựa trên độ dài (1.0 là không thay đổi).
* early\_stopping=True: Dừng sớm khi tìm được câu trả lời hoàn chỉnh.
* no\_repeat\_ngram\_size=2: Ngăn không cho lặp lại các n-gram có độ dài 2.

## **3.5. Đánh giá kết quả**

Sau quá trình phát triển và triển khai thử nghiệm, hệ thống đã được đánh giá trên nhiều khía cạnh khác nhau để xác định hiệu quả hoạt động và khả năng đáp ứng mục tiêu đề ra. Kết quả đánh giá chi tiết được phân tích trên các tiêu chí cụ thể như sau:

Về giao diện người dùng, hệ thống được thiết kế trực quan và dễ sử dụng. Giao diện được thiết kế với các thành phần được bố trí hợp lý và có tính nhất quán cao. Khung chat được thiết kế rõ ràng, dễ nhìn, với phần hiển thị lịch sử hội thoại giúp người dùng dễ dàng theo dõi và tham khảo lại các thông tin đã trao đổi. Tuy nhiên, vẫn còn một số hạn chế như chưa hỗ trợ đầy đủ chế độ hiển thị trên các thiết bị di động và cần tiếp tục nghiên cứu và phát triển để tối ưu hóa giao diện, làm cho trải nghiệm người dùng trở nên ngày càng thuận tiện và thân thiện hơn nữa.

Về độ chính xác, hệ thống đã trả lời tốt được các câu hỏi liên quan đến luật giao thông. Mô hình đã hoạt động đúng khi tích hợp lên web và trả lời các câu hỏi đúng như đã huấn luyện. Tuy nhiên, với những câu hỏi phức tạp hoặc liên quan đến nhiều văn bản khác nhau, độ chính xác còn cần được cải thiện thêm.

Về tốc độ phản hồi, hệ thống đạt được thời gian trả lời trung bình từ 2-3 giây cho mỗi câu hỏi, đáp ứng được yêu cầu về tương tác thời gian thực với người dùng. Thời gian xử lý này khá ổn định và không bị ảnh hưởng nhiều bởi độ dài của câu hỏi hay số lượng người dùng truy cập đồng thời.

Đánh giá tổng quan, hệ thống đã đạt được mục tiêu cơ bản là xây dựng một công cụ hỗ trợ tra cứu thông tin văn bản hành chính hiệu quả và thân thiện với người dùng. Tuy nhiên, vẫn cần tiếp tục cải thiện về độ chính xác, tốc độ xử lý, và bổ sung thêm các tính năng mới để đáp ứng tốt hơn nhu cầu của người dùng trong tương lai.

# **CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **4.1. Kết luận về đề tài**

Trong bối cảnh chuyển đổi số ngày càng mạnh mẽ, việc ứng dụng công nghệ trí tuệ nhân tạo vào lĩnh vực văn bản hành chính không chỉ là một xu hướng mà còn là một nhu cầu cấp thiết. Ứng dụng trả lời văn bản hành chính là một giải pháp sáng tạo, mang lại nhiều lợi ích thiết thực cho cơ quan, tổ chức và người dân.

Sau quá trình tìm hiểu và thực hiện, đồ án đã hoàn thành và giao tiếp và trả lời về văn bản hành chính nói chung và văn bản giao thông hành chính nói riêng. Biết được quá trình thu thập và xử lý dữ liệu để có thể tiến hành huấn luyện trí tuệ nhân tạo. Hoàn thành mục tiêu và kết quả đề ra. Tập dữ liệu vẫn chưa hoàn thiện nên cần phải cập nhật thêm nhiều dữ liệu về các văn bản hành chính khác, đồng thời cũng có thể nâng cấp phần cứng để tăng tốc độ xử lý và hiệu suất của sản phẩm.

**Kết quả đạt được:**

* Hiểu về mô hình Transformer, các dạng kiến trúc trí tuệ nhân tạo như Bert, Bard và áp dụng vào xây dựng mô hình trả lời văn bản hành chính.
* Hiểu cách và các phương pháp thu thập, xử lý dữ liệu trong mô hình.
* Mô hình có khả năng cung cấp thông tin nhanh chóng, chính xác và liên tục , giúp giảm tải công việc cho nhân viên hành chính.
* Người dân và tổ chức có thể tra cứu thông tin hành chính dễ dàng, không cần chờ đợi hoặc thực hiện các thủ tục phức tạp.
* Tiết kiệm về nguồn lực giúp tự động hóa các công việc hành chính cơ bản, giảm thiểu chi phí vận hành và nguồn lực nhân sự.

**Hạn chế:**

* Chương trình hiện tại chưa có khả năng xử lý các câu hỏi phức tạp hoặc các ngữ cảnh đa nghĩa trong văn bản hành chính.
* Vì mang tính nghiên cứu nên hệ thống chỉ mới làm việc trên tập dữ liệu nhỏ với số lượng văn bản có hạn.
* Chưa tích hợp được lên hệ thống hoàn chỉnh để sử dụng trong thực tế tại các cơ quan, tổ chức.

## **4.2. Hướng phát triển của đề tài**

Đề tài có tính thực tiễn cao khi ứng dụng trí tuệ nhân tạo (AI) vào việc hỗ trợ người dân và cơ quan chức năng trong việc tra cứu, xử lý thông tin. Trong bối cảnh các công cụ AI ngày càng phổ biến, ứng dụng này không chỉ giúp nâng cao hiệu suất làm việc mà còn góp phần tạo ra một hệ thống thông minh, thân thiện với người dùng.

Trong quá trình phát triển, ứng dụng có thể được mở rộng bằng cách bổ sung thêm dữ liệu về các văn bản hành chính thuộc nhiều lĩnh vực khác nhau như xây dựng, y tế, giáo dục, giao thông, pháp luật,... Việc mở rộng phạm vi dữ liệu sẽ giúp ứng dụng phục vụ đa dạng nhu cầu của người dùng hơn, không chỉ dừng lại ở một nhóm đối tượng cụ thể mà có thể hỗ trợ cả doanh nghiệp, tổ chức và cá nhân trong nhiều lĩnh vực khác nhau. Khi dữ liệu ngày càng phong phú, hệ thống có thể cung cấp những câu trả lời chính xác hơn, đáp ứng được nhiều tình huống thực tế hơn.

Bên cạnh đó, việc tích hợp chức năng giao tiếp bằng giọng nói sẽ giúp ứng dụng trở nên thân thiện hơn với người dùng. Điều này đặc biệt hữu ích đối với những đối tượng như người cao tuổi, người khuyết tật hoặc những người gặp khó khăn trong việc nhập liệu bằng văn bản. Thay vì chỉ trả lời câu hỏi dựa trên văn bản, ứng dụng có thể phát triển thành một trợ lý ảo thực sự, hỗ trợ người dùng thực hiện các thao tác tìm kiếm, tra cứu và hướng dẫn sử dụng thông qua hội thoại. Ngoài ra, nâng cấp hệ thống nhận diện giọng nói để hỗ trợ nhiều ngữ điệu, vùng miền khác nhau cũng là một hướng phát triển quan trọng, giúp tăng tính cá nhân hóa và độ chính xác trong quá trình giao tiếp.

Không chỉ dừng lại ở nền tảng web, ứng dụng có thể được triển khai trên nhiều nền tảng khác nhau như mobile (iOS, Android) hay chatbot trên các ứng dụng nhắn tin phổ biến như Messenger, Zalo, Telegram,... Điều này không chỉ giúp mở rộng tệp người dùng mà còn tạo ra nhiều cơ hội ứng dụng trong thực tế. Trên nền tảng mobile, ứng dụng có thể tích hợp vào lĩnh vực giáo dục, giúp học sinh, sinh viên dễ dàng tra cứu thông tin, tài liệu học tập hoặc thậm chí phát triển thành một công cụ học tập thông minh. Đối với doanh nghiệp, hệ thống có thể hỗ trợ đào tạo nội bộ, giúp nhân viên dễ dàng tiếp cận và tra cứu các quy định, văn bản liên quan đến công việc của họ.

Xử lý ngôn ngữ tự nhiên (NLP) là một lĩnh vực đang phát triển mạnh mẽ và ngày càng có nhiều cải tiến đáng kể. Trong tương lai, ứng dụng có thể thay đổi mô hình AI tốt hơn, hiệu quả hơn, giúp tăng cường khả năng hiểu và phản hồi chính xác hơn.

Bên cạnh việc mở rộng tính năng và nâng cao chất lượng phản hồi, đảm bảo tính bảo mật và cá nhân hóa dữ liệu người dùng cũng là một khía cạnh quan trọng. Khi ứng dụng được triển khai rộng rãi và tiếp cận nhiều người dùng, việc bảo mật thông tin cá nhân, dữ liệu tìm kiếm trở thành một yếu tố cần được ưu tiên hàng đầu. Hệ thống cũng có thể phát triển các tính năng cá nhân hóa, giúp ghi nhớ sở thích, lịch sử tra cứu của từng cá nhân để tối ưu hóa trải nghiệm và mang lại câu trả lời phù hợp hơn với nhu cầu cụ thể của từng người.

Với sự phát triển không ngừng của công nghệ AI, đề tài có tiềm năng lớn để mở rộng và ứng dụng vào nhiều lĩnh vực khác nhau. Việc tích hợp các tính năng mới như hỗ trợ giọng nói, triển khai đa nền tảng và nâng cao khả năng xử lý ngôn ngữ tự nhiên sẽ giúp ứng dụng trở thành một công cụ hữu ích và mang lại nhiều giá trị thực tiễn cho người dùng.

# **TÀI LIỆU THAM KHẢO**

[1] <https://arxiv.org/pdf/1910.13461v1>

[2] John Paul Mueller, Luca Massaron – Artificial Intelligence for Dummies-Wiley (2018)

[3] An Essential Guide to Machine Learning for Beginners By Herbert Jones

[4] https://huggingface.co/

[5] https://github.com/VinAIResearch/BARTpho

[6] https://www.miai.vn/2020/12/01/mi-transformer-tim-hieu-transformer-theo-cach-de-hieu-de-nho/

[7] https://www.miai.vn/2020/12/14/bert-series-chuong-1-bert-la-cai-chi-chi/

# **PHỤ LỤC**

## **5.1. Hướng dẫn cài đặt ứng dụng**

Backend

* Vào thư mục backend
* Cài đặt python và các thư viện cần thiết: pip install flask flask-cors transformers torch
* Chạy flask: python main.py

Frontend

* Vào thư mục frontend
* Cài đặt node và các indepencies: yarn install
* Chạy Nuxtjs: yarn dev

## **5.2. Hướng dẫn sử dụng ứng dụng**