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.SĨ NGUYỄN THỊ HUỆ** |
| **SINH VIÊN THỰC HIỆN:** | **HOÀNG TUẤN VŨ** |
| **MÃ SINH VIÊN :**  **LỚP:** | **2055010293**  **20CN5** |

**Hà Nội, tháng 10 năm 2024**

**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.

Em muốn bày tỏ lòng biết ơn chân thành đến cô Nguyễn Thị Huệ đã hỗ trợ và hướng dẫn em suốt quá trình thực hiện đề tài này. Sự đóng góp của cô đã giúp em hiểu rõ hơn về các vấn đề gặp phải và hoàn thiện đề tài của mình.

Mục lục

[Chương 1: Cơ sở lý thuyết 5](#_heading=h.30j0zll)

[1.1. Tổng quan về trí tuệ nhân tạo 5](#_heading=h.1fob9te)

[1.1.1. Khái niệm 5](#_heading=h.3znysh7)

[1.1.2. Các hướng nghiên cứu 6](#_heading=h.2et92p0)

[1.1.3. Phân loại trí tuệ nhân tạo 7](#_heading=h.tyjcwt)

[1.1.4. Ứng dụng 9](#_heading=h.3dy6vkm)

[1.1.5. Mặt tích cực và hạn chế của trí tuệ nhân tạo 11](#_heading=h.1t3h5sf)

[1.2. Tổng quan về đề tài 12](#_heading=h.4d34og8)

[1.2.1. Yêu cầu 12](#_heading=h.2s8eyo1)

[1.2.2. Mô tả bài toán 13](#_heading=h.17dp8vu)

[1.2.3. Mô hình 13](#_heading=h.3rdcrjn)

[1.2.4 Datasets 15](#_heading=h.26in1rg)

[1.3. Công nghệ sử dụng 15](#_heading=h.lnxbz9)

[1.3.1. Python 15](#_heading=h.35nkun2)

[1.3.2. Transformer 16](#_heading=h.1ksv4uv)

[1.3.3. PyTorch 16](#_heading=h.44sinio)

[1.3.4. Flask 17](#_heading=h.2jxsxqh)

[Chương 2: Quy trình 17](#_heading=h.z337ya)

[2.1. Chuẩn bị tập dữ liệu 17](#_heading=h.3j2qqm3)

[2.2. Tiền xử lý dữ liệu 18](#_heading=h.1y810tw)

[2.3. Phân chia dữ liệu 18](#_heading=h.4i7ojhp)

[2.4. Lựa chọn mô hình 19](#_heading=h.2xcytpi)

[2.5. Huấn luyện mô hình 19](#_heading=h.1ci93xb)

[2.6. Đánh giá kết quả 20](#_heading=h.3whwml4)

[Chương 3: Triển khai và thực nghiệm 22](#_heading=h.2bn6wsx)

[3.1 Công cụ sử dụng 22](#_heading=h.qsh70q)

[3.2 Phân tích thiết kế hệ thống 22](#_heading=h.3as4poj)

[3.3 Quy trình thực nghiệm 22](#_heading=h.1pxezwc)

[3.4 Đánh giá mô hình 22](#_heading=h.49x2ik5)

[3.5 Kết luận 22](#_heading=h.2p2csry)

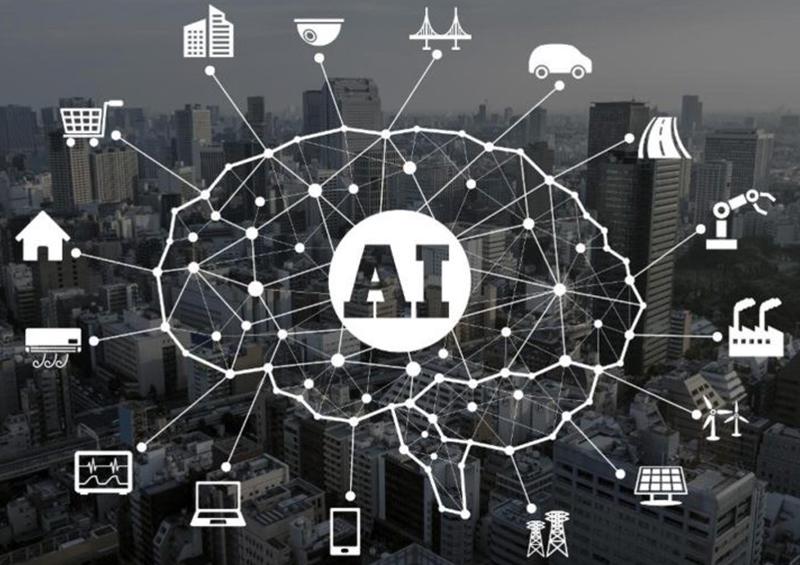
Mục lục hình ảnh và ký hiệu viết tắt

# 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 về đề tài

### 1.2.1. Yêu cầu

Ứng dụng trả lời văn bản trả lời được những câu hỏi:

* Tên của văn bản hành chính
* Nội dung của văn bản hành chính
* Ngày ban hành, cơ quan ban hành của văn bản hành chính

### 1.2.2. Mô tả bài toán

Lĩnh vực: Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP)

Xử lý ngôn ngữ tự nhiên là lĩnh vực trong trí tuệ nhân tạo (AI) nhằm giúp máy tính hiểu, xử lý và tương tác với ngôn ngữ tự nhiên của con người. Trong bài toán này, tập trung vào bài toán Hỏi - Đáp (Question Answering - QA).

Đầu vào (Input)

* Context: Văn bản chứa nội dung thông tin cần thiết, có thể là một đoạn văn, tài liệu, văn bản hành chính hoặc file chứa đoạn văn ví dụ như .docx .pdf .txt
* Question: Một câu hỏi được đặt ra liên quan đến nội dung trong đoạn văn bản (Context).

Đầu ra (Output)

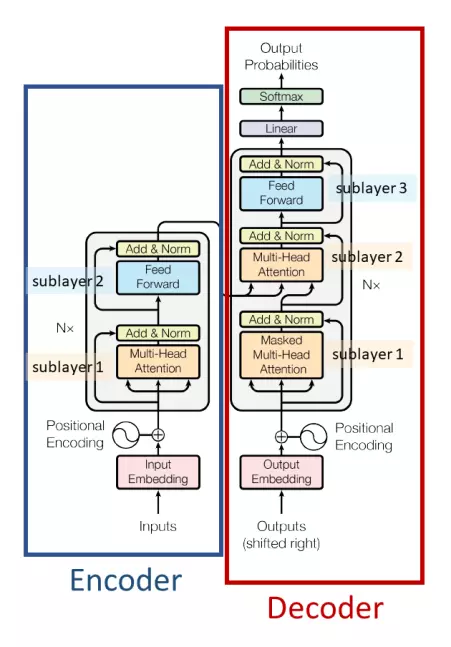
* Answer: Kết quả trả lời câu hỏi dựa trên nội dung trong đoạn văn bản (Context). Đầu ra là một đoạn văn ngắn lấy từ Context (Extractive QA).

### 1.2.3. Mô hình

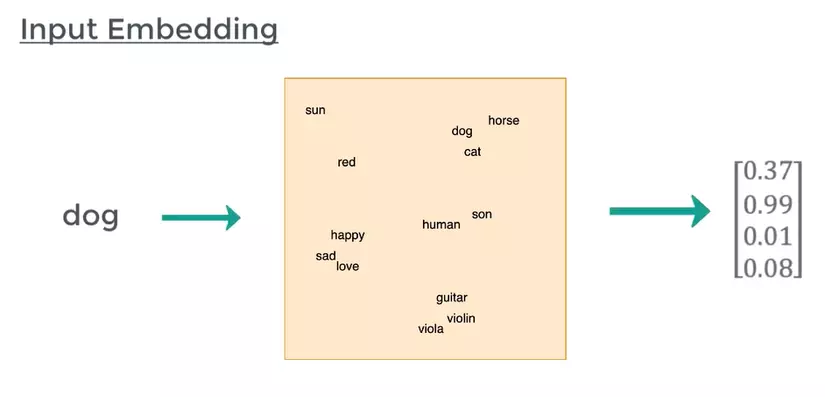
Để giải quyết bài toán Hỏi - Đáp (Question Answering), đồ án sử dụng mô hình học sâu dựa trên kiến trúc Transformer, trong đó triển khai cụ thể với phiên bản BERT được tùy chỉnh cho tiếng Việt là PhoBERT.

1.2.3.1. Kiến trúc Transformer

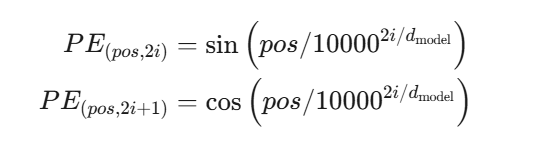
Transformer là một kiến trúc mô hình học sâu ra đời từ bài báo “Attention is All You Need” của Vaswani và cộng sự (2017). Transformer đã cách mạng hóa lĩnh vực Xử lý ngôn ngữ tự nhiên nhờ khả năng xử lý dữ liệu theo cách song song và tận dụng hiệu quả các mối quan hệ ngữ nghĩa trong văn bản.



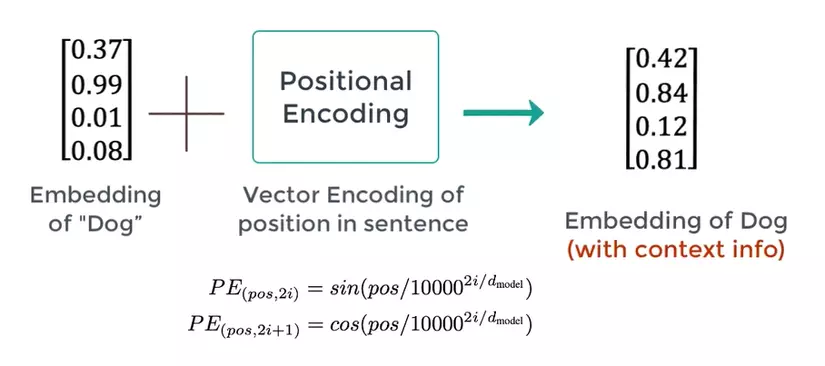
**Input Embedding:** Máy tính không hiểu câu chữ mà chỉ đọc được số, vector, ma trận; vì vậy cần phải biểu diễn câu chữ dưới dạng vector, gọi là input embedding. Điều này đảm bảo các từ gần nghĩa có vector gần giống nhau.

****

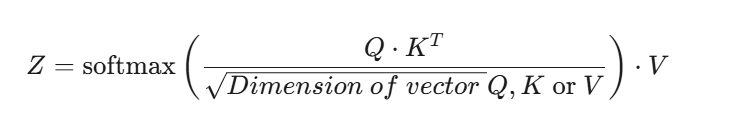
**Positional Encoding:** Word embeddings giúp biểu diễn ngữ nghĩa của một từ, tuy nhiên cùng một từ ở vị trí khác nhau của câu lại mang ý nghĩa khác nhau. Do đó Transformers có thêm một phần Positional Encoding để inject thêm thông tin về vị trí của một từ



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:



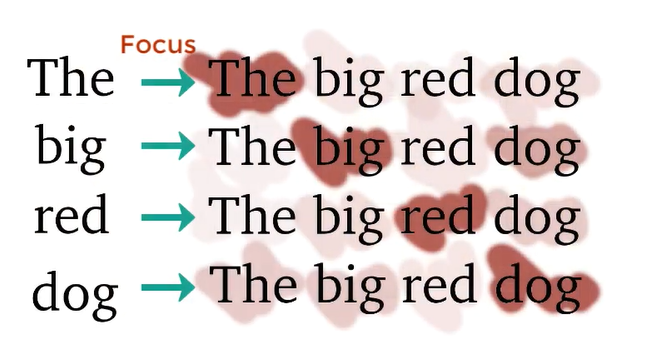
**Self-Attention:** Self-Attention là cơ chế giúp Transformers hiểu được sự liên quan giữa các từ trong một câu, bất kể khoảng cách giữa chúng.



Để có được 3 vectors Q, K, V, input embeddings được nhân với 3 ma trận trọng số tương ứng (được tune trong quá trình huấn luyện) WQ, WK, WV.

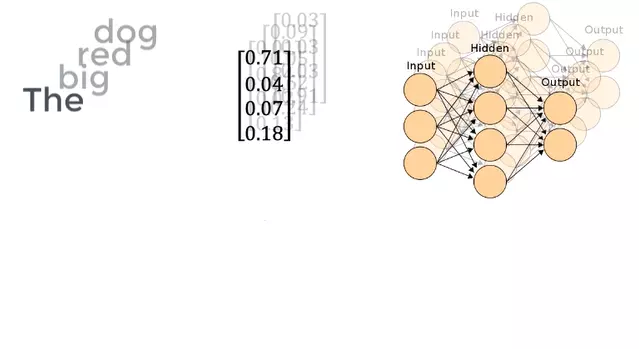
**Multi-head Attention:** Vấn đề của Self-attention là attention của một từ sẽ luôn "chú ý" vào chính nó. Thay vì sử dụng một bộ ma trận trọng số duy nhất WQ,WK,WV cơ chế Multi-Head Self-Attention sử dụng nhiều bộ ma trận (được gọi là "heads").

Ví dụ như sau:

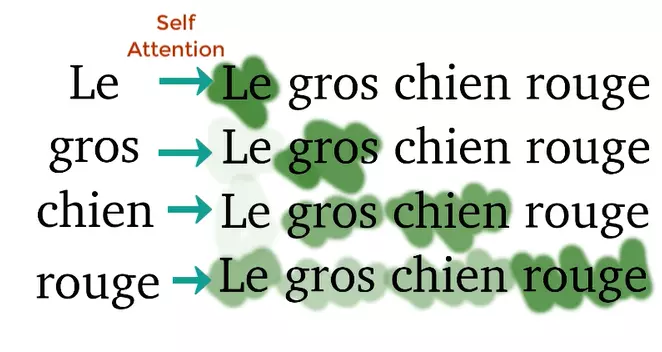


Vì mỗi "head" sẽ cho ra một ma trận attention riêng nên ta phải kết hợp (concatenate) các ma trận này và nhân với ma trận trọng số WO để ra một ma trận attention duy nhất (weighted sum).

**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.



**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.

****

**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.

1.2.3.2. 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.

Đặc điểm của BERT:

* Masked Language Modeling (MLM): BERT được huấn luyện bằng cách dự đoán các từ bị che đi (masked) trong văn bản, giúp mô hình hiểu được ngữ nghĩa của từ dựa trên ngữ cảnh xung quanh.
* Next Sentence Prediction (NSP): Một nhiệm vụ khác trong quá trình huấn luyện BERT là dự đoán câu tiếp theo có phù hợp với câu hiện tại hay không, giúp mô hình hiểu được quan hệ giữa các câu.

### 1.2.4 Datasets

Nguồn dữ liệu:

* Văn bản hành chính: Các tài liệu công khai như quyết định, quy định hoặc công văn từ các cơ quan nhà nước.
* Dữ liệu tự xây dựng: Tự tạo dữ liệu bằng cách tổng hợp và gán nhãn các cặp câu hỏi - câu trả lời phù hợp với ngữ cảnh.

Kích thước và chất lượng dữ liệu:

* Tập dữ liệu ban đầu bao gồm 1,000 mẫu dữ liệu được gán nhãn thủ công.
* Sử dụng phương pháp đánh giá chéo (cross-validation) để đảm bảo chất lượng và độ chính xác của dữ liệu.

Tiền xử lý dữ liệu:

* Loại bỏ các đoạn văn bản dư thừa hoặc chứa nhiều lỗi chính tả.
* Chuẩn hóa văn bản bằng cách loại bỏ ký tự đặc biệt và chuẩn hóa dấu câu.
* Mã hóa dữ liệu bằng tokenizer của Phobert để đưa vào mô hình.

## 1.3. Công nghệ sử dụng

### 1.3.1. Python

Python là ngôn ngữ lập trình chính được sử dụng trong đồ án nhờ vào tính đơn giản, khả năng mở rộng, và hệ sinh thái phong phú của các thư viện hỗ trợ. Python đóng vai trò quan trọng trong:

* Xử lý dữ liệu: Tiền xử lý văn bản, mã hóa đầu vào, và phân tích kết quả.
* Phát triển mô hình: Sử dụng các thư viện học sâu như PyTorch để xây dựng và huấn luyện mô hình.
* Tích hợp ứng dụng: Xây dựng giao diện lập trình ứng dụng (API) bằng Flask để triển khai mô hình trên nền tảng web.

### 1.3.2. Transformer

Transformer là kiến trúc mô hình học sâu hiện đại, đã cách mạng hóa lĩnh vực NLP. Trong đồ án, kiến trúc transformer được ứng dụng thông qua mô hình Phobert (vinai/phobert-base-v2), được tinh chỉnh để giải quyết bài toán Hỏi - Đáp.

Các đặc điểm nổi bật:

* Self-Attention Mechanism: Giúp mô hình nắm bắt mối quan hệ giữa các từ trong ngữ cảnh một cách hiệu quả, ngay cả với các đoạn văn bản dài.
* Fine-Tuning: Tinh chỉnh mô hình transformer để thích ứng với bài toán cụ thể, giúp cải thiện độ chính xác trong việc trả lời câu hỏi.
* Tokenizer: Sử dụng cơ chế Byte-Pair Encoding (BPE) được tối ưu hóa cho ngôn ngữ tiếng Việt, hỗ trợ việc xử lý văn bản hiệu quả.

### 1.3.3. PyTorch

PyTorch là thư viện mã nguồn mở mạnh mẽ cho học sâu, được sử dụng để xây dựng và huấn luyện mô hình trong đồ án.

* Giao diện thân thiện, dễ dàng kiểm tra và gỡ lỗi.
* Hỗ trợ tính toán trên GPU, tăng tốc quá trình huấn luyện mô hình.
* Tích hợp tốt với các thư viện transformer như Hugging Face Transformers.
* Ứng dụng trong đồ án:
* Định nghĩa kiến trúc mô hình dựa trên transformer.
* Xử lý và tối ưu hóa loss function trong quá trình huấn luyện.
* Đánh giá hiệu suất mô hình bằng các thước đo tiêu chuẩn.

### 1.3.4. Flask

Flask là một micro-framework Python phổ biến, được sử dụng để xây dựng giao diện lập trình ứng dụng (API) cho mô hình.

Vai trò:

* Kết nối giữa mô hình AI và người dùng qua giao diện web.
* Cung cấp các endpoint RESTful cho việc gửi câu hỏi và nhận câu trả lời.
* Đảm bảo ứng dụng hoạt động nhẹ, nhanh và dễ triển khai.

Chức năng:

* Endpoint /predict: Nhận đầu vào gồm Context và Question, sau đó trả về Answer được mô hình tạo ra.

Kiến trúc triển khai:

* Mô hình được huấn luyện và lưu dưới dạng checkpoint.
* Flask API được tích hợp để tải checkpoint và xử lý yêu cầu từ người dùng.
* Ứng dụng Flask có thể triển khai trên các nền tảng như Heroku, Vercel, hoặc máy chủ nội bộ.

# Chương 2: Quy trình

## 2.1. Chuẩn bị tập dữ liệu

Để huấn luyện mô hình cần chuẩn bị một tập dữ liệu, tập dữ liệu là yếu tố vô cùng quan trọng trong việc huấn luyện mô hình. Tập dữ liệu sạch, đầy đủ và chất lượng sẽ giúp xây dựng mô hình AI chính xác và hiệu quả.

Để xây dựng một tập dữ liệu thống nhất và hiệu quả, tất cả các dữ liệu được lưu dưới dạng SquAD 2.0 (Standford Question Answering Dataset 2.0) là một bộ dữ liệu trong lĩnh vực xử lý ngôn ngữ tự nhiên. SquAD 2.0 là phiên bản mở rộng của SquAD 1.1, bổ sung các tính năng quan trọng nhằm giải quyết nhiều khía cạnh thực tế hơn trong việc hiểu văn bản và trả lời câu hỏi.

Dữ liệu được thu thập là các văn bản hành chính được ban hành gồm nhiều nội dung như báo cáo, thông tư, quyết định, nghị quyết, ...

Ví dụ: Thông tư 01/2020/TT-BTP

Context: Thông tư 01/2020/TT-BTP quy định chi tiết thi hành một số điều của Luật Hộ tịch, có hiệu lực từ ngày 01/07/2020. Thông tư này áp dụng cho cơ quan hộ tịch, tổ chức và cá nhân liên quan đến đăng ký hộ tịch tại Việt Nam.

Question: Thông tư 01/2020/TT-BTP có hiệu lực từ ngày nào?

Answers: ngày 01/07/2020

## 2.2. Tiền xử lý dữ liệu

Sau khi thu thập dữ liệu thành công cần đánh dấu vị trí của câu trả lời bắt đầu trong nội dung văn bản

Context: Thông tư 01/2020/TT-BTP quy định chi tiết thi hành một số điều của Luật Hộ tịch, có hiệu lực từ ngày 01/07/2020. Thông tư này áp dụng cho cơ quan hộ tịch, tổ chức và cá nhân liên quan đến đăng ký hộ tịch tại Việt Nam.

Question: Thông tư 01/2020/TT-BTP có hiệu lực từ ngày nào?

Answers: ngày 01/07/2020

Answer start: 96

Để mô hình có thể hiểu được ta cần phải mã hóa dữ liệu

Input IDs là một danh sách các số nguyên, trong đó mỗi số đại diện cho một token sau khi văn bản được tokenize bởi mô hình ngôn ngữ.

Vai trò của input\_ids:

* Chuyển đổi văn bản sang dạng số để mô hình có thể xử lý
* Đảm bảo mỗi từ được biểu diễn duy nhất
* Cho phép mô hình học và hiểu mối quan hệ giữa các token

Quá trình chuyển đổi diễn ra như sau:

1. Văn bản ban đầu (text) →
2. Tokenizer phân tách thành các token →
3. Mỗi token được ánh xạ tới một ID duy nhất

Vì phải có độ dài cố định nên phải thêm padding vào câu <pad>

Việc thêm padding trong xử lý ngôn ngữ tự nhiên là vô cùng quan trọng:

Đồng nhất độ dài input

* Các mô hình học máy yêu cầu input có độ dài cố định
* Các văn bản có độ dài khác nhau được điều chỉnh về một độ dài chung

Hiệu quả tính toán

* GPU và các thiết bị tính toán hoạt động hiệu quả nhất khi xử lý các tensor có kích thước đồng nhất
* Giảm thiểu overhead trong quá trình batch processing

[CLS] và [SEP] là hai token đặc biệt trong nhiều mô hình transformer như BERT, được sử dụng cho các nhiệm vụ xử lý ngôn ngữ tự nhiên.

[CLS] (Classification Token):

* Được đặt ở đầu chuỗi token
* Viết tắt của "Classification"
* Mục đích: Biểu diễn toàn bộ câu hoặc văn bản. Được sử dụng cho các nhiệm vụ phân loại văn bản. Chứa thông tin ngữ cảnh tổng thể

[SEP] (Separator Token):

* Viết tắt của "Separator"
* Được sử dụng để: Phân tách các câu hoặc các phần khác nhau của văn bản. Đánh dấu ranh giới giữa các phần trong input

Ví dụ:

Context: Thông tư 01/2020/TT-BTP quy định chi tiết thi hành một số điều của Luật Hộ tịch, có hiệu lực từ ngày 01/07/2020. Thông tư này áp dụng cho cơ quan hộ tịch, tổ chức và cá nhân liên quan đến đăng ký hộ tịch tại Việt Nam.

Question: Thông tư 01/2020/TT-BTP có hiệu lực từ ngày nào?

Answer: ngày 01/07/2020

Input IDs: [0, 4848, 1713, 7818, 3914, 5419, 12607, 253, 10, 3256, 2792, 39, 43, 40865, 1672, 114, 2, 2, 4848, 1713, 7818, 3914, 5419, 12607, 253, 4026, 2608, 811, 3365, 201, 2537, 16, 100, 184, 7, 761, 13977, 34562, 1347, 4, 10, 3256, 2792, 39, 43, 7818, 16425, 3914, 3914, 5, 4848, 1713, 23, 2925, 8410, 13, 2209, 2665, 506, 34562, 1347, 4, 1391, 1766, 6, 569, 1362, 2657, 2665, 30, 1813, 765, 506, 26865, 35, 350, 43557, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

Tokens: ['<s>', 'Thông', 'tư', '01/@@', '20@@', '20/@@', 'TT-B@@', 'TP', 'có', 'hiệu', 'lực', 'từ', 'ngày', 'nà@@', 'o@@', '?', '</s>', '</s>', 'Thông', 'tư', '01/@@', '20@@', '20/@@', 'TT-B@@', 'TP', 'quy', 'định', 'chi', 'tiết', 'thi', 'hành', 'một', 'số', 'điều', 'của', 'Luật', 'Hộ', 'tị@@', 'ch@@', ',', 'có', 'hiệu', 'lực', 'từ', 'ngày', '01/@@', '07/@@', '20@@', '20@@', '.', 'Thông', 'tư', 'này', 'áp', 'dụng', 'cho', 'cơ', 'quan', 'hộ', 'tị@@', 'ch@@', ',', 'tổ', 'chức', 'và', 'cá', 'nhân', 'liên', 'quan', 'đến', 'đăng', 'ký', 'hộ', 'tịch', 'tại', 'Việt', 'Nam.', '</s>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>', '<pad>']

## 2.3. 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)

## 2.4. Lựa chọn mô hình

PhoBERT: Phiên bản BERT cho tiếng Việt

PhoBERT là một mô hình transformer được phát triển dựa trên kiến trúc BERT, nhưng được tối ưu hóa cho tiếng Việt bởi VinAI. Điểm nổi bật:

* Tối ưu hóa ngôn ngữ: PhoBERT được huấn luyện trên kho dữ liệu văn bản tiếng Việt lớn, giúp mô hình hiểu sâu ngữ pháp và ngữ nghĩa đặc thù của tiếng Việt.
* Tokenization: Sử dụng kỹ thuật Byte-Pair Encoding (BPE) để chia từ thành các token nhỏ hơn, phù hợp với đặc thù ngôn ngữ đơn tiết và dấu của tiếng Việt.

Ứng dụng PhoBERT trong bài toán Hỏi - Đáp:

Fine-Tuning:

PhoBERT được tinh chỉnh trên tập dữ liệu dạng SQuAD (Stanford Question Answering Dataset) bằng cách tối ưu hóa dự đoán vị trí bắt đầu và kết thúc của câu trả lời trong ngữ cảnh.

Đầu vào của PhoBERT là tổ hợp của:

[CLS] + Question + [SEP] + Context + [SEP].

Token [CLS] chứa thông tin tổng hợp của toàn bộ đầu vào.

Output:

PhoBERT dự đoán hai điểm quan trọng trong văn bản: vị trí bắt đầu (start position) và vị trí kết thúc (end position) của câu trả lời trong đoạn văn.

Quy trình ứng dụng:

* Tiền xử lý: Token hóa Context và Question bằng tokenizer của PhoBERT. Chuyển đổi dữ liệu đầu vào thành định dạng phù hợp với mô hình.
* Huấn luyện: Sử dụng tập dữ liệu huấn luyện với thông tin về câu hỏi và câu trả lời. Tối ưu hóa hàm mất mát dựa trên vị trí bắt đầu và kết thúc của câu trả lời.
* Dự đoán: Với câu hỏi mới, mô hình trả về đoạn văn chứa câu trả lời trong Context.

## 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 **2** 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 BERT-based.
* **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

Độ chính xác Accuracy:

"Accuracy" (Chính xác) 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:

Accuracy= Số lượng dự đoán đúng/Tổng số dự đoán ×100%

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 – sự mất mát

"Loss" (mất mát) là một khái niệm trong học máy và máy học để đo lường sự chênh lệch giữa giá trị dự đoán của mô hình và giá trị thực tế của dữ liệu huấn luyện. Mục tiêu là giảm thiểu giá trị Loss, tức là làm cho dự đoán của mô hình càng gần với thực tế càng tốt.

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: 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= TPFP+TP ×100%

Và khi 0< Precision <=1, Precision càng lớn có nghĩa là độ chính xác của các điểm tìm được càng cao. Đối với mô hình nhận diện biển báo có Precision khá cao

RECALL= True PositiveTrue Positive+False Negative ×100%

Công thức trên có nghĩa là tỷ lệ giữa các điểm positive thực được nhận đúng trên tổng điểm Positive thực

F-1 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= 2×Precision ×Recall/Precision+Recall

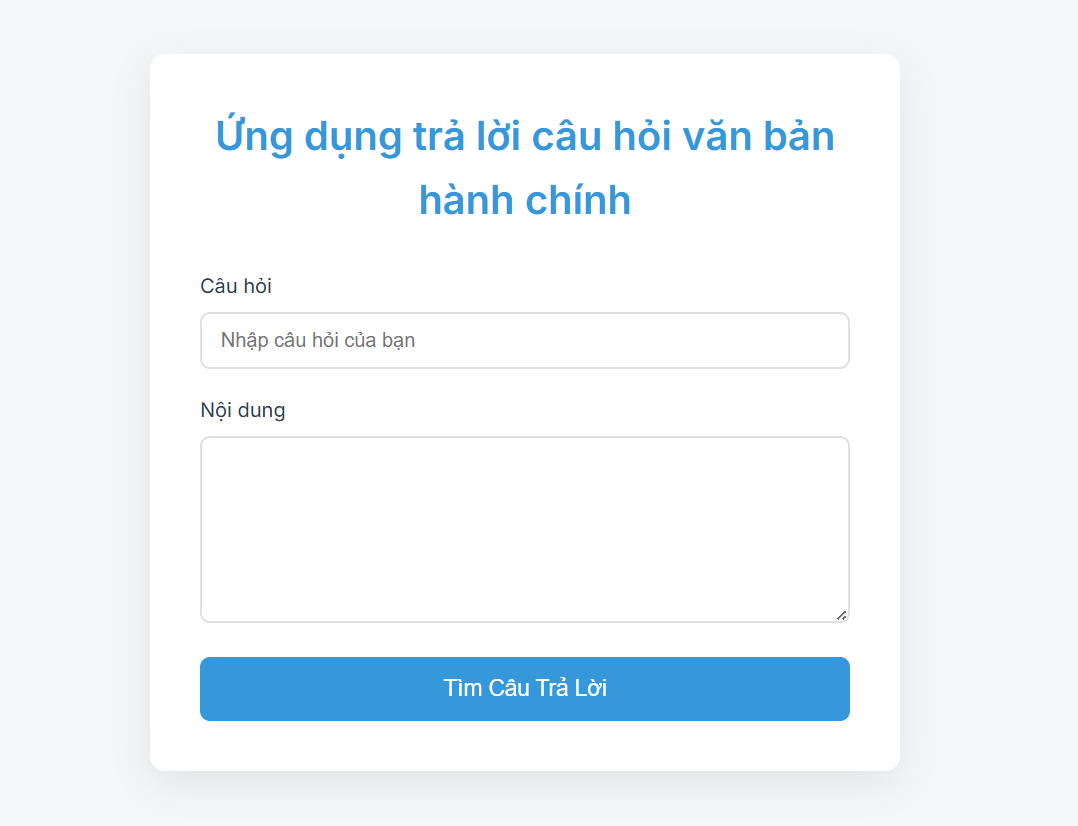
F1-score tự nhiên có giá trị trong khoảng từ 0 đến 1, với 1 là giá trị tốt nhất (đạt được khi cả Precision và Recall đều bằng 1). F1-score giúp cân bằng giữa Precision và Recall, giúp đánh giá hiệu suất của mô hình một cách tổng thể, đặc biệt trong trường hợp có sự mất cân bằng giữa các lớp.

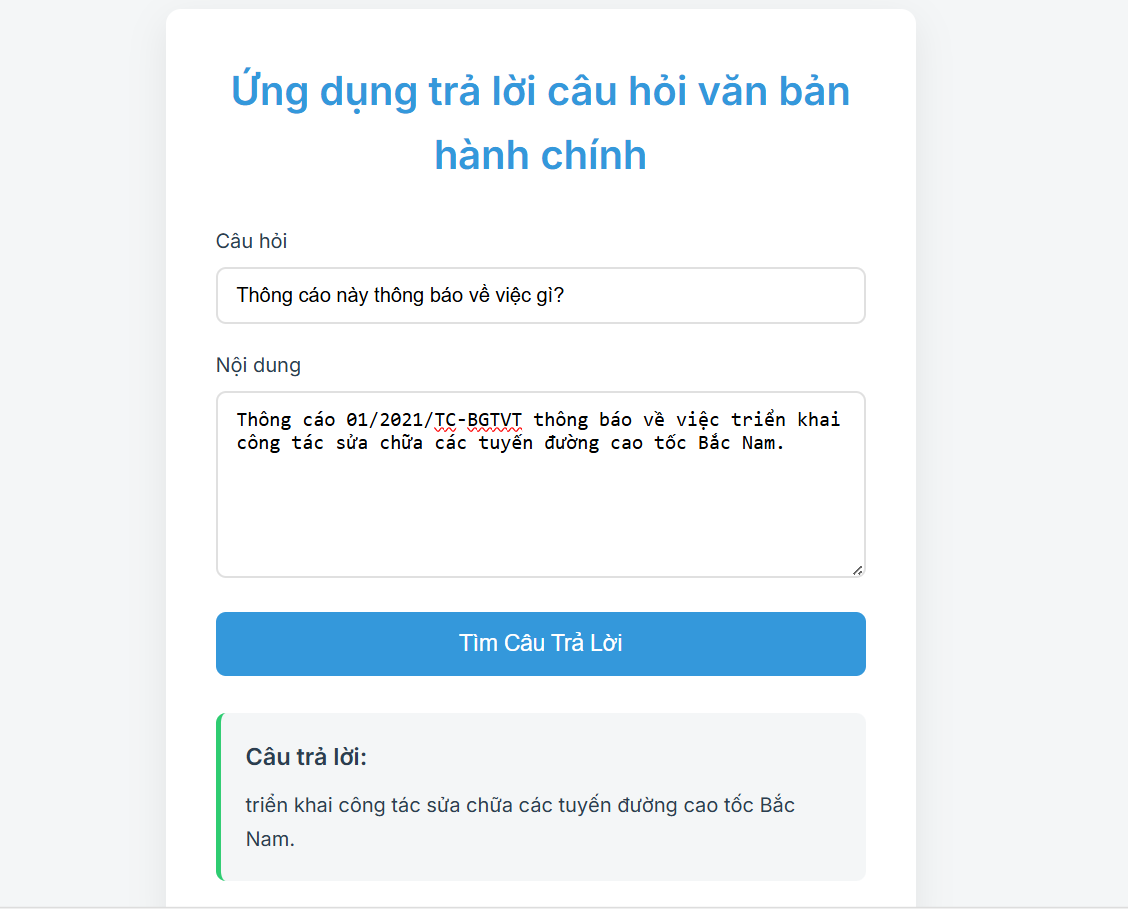
# Chương 3: Triển khai và thực nghiệm

## 3.1 Công cụ sử dụng

## 3.2 Phân tích thiết kế hệ thống

## 3.3 Quy trình thực nghiệm





## 3.4 Đánh giá mô hình

## 3.5 Kết luận

TÀI LIỆU THAM KHẢO