|  |  |
| --- | --- |
| **Đại HỌC QUỐC GIA HÀ NỘI** | |
| **tRƯỜNG đẠI HỌC CÔNG NGHỆ** | |
| **----------------------------------------** | |
|  | |
| BÁO CÁO THỰC TẬP | |
| NGÀNH: CÔNG NGHỆ THÔNG TIN | |
|  | |
| ĐỀ TÀI: NGHIÊN CỨU VÀ XÂY DỰNG MÔ HÌNH PHÂN LỚP CÂU HỎI TRONG LĨNH VỰC  PHÁP LUẬT VỀ CƯ TRÚ | |
|  | |
|  | |
| Giảng viên hướng dẫn: Lê Hoàng Quỳnh | |
|  | |
|  |  |
|  | **Sinh viên:** Lương Thế Quyền |
|  | **Mã sinh viên:** 22021156 |
|  | **Lớp:** K67I-IT1 |
|  |  |
|  |  |
| **Hà Nội, tháng 10 năm 2025** | |

# mục lục

[MỤC LỤC 1](#_Toc211353153)

[LỜI CẢM ƠN 2](#_Toc211353154)

[CHƯƠNG I: TỔNG QUAN 3](#_Toc211353155)

[I. Đặt vấn đề 3](#_Toc211353156)

[II. Bối cảnh thực tập 3](#_Toc211353157)

[CHƯƠNG II: PHÁT BIỂU BÀI TOÁN 4](#_Toc211353158)

[I. Kiến trúc bài toán tổng thể 4](#_Toc211353159)

[II. Nhiệm vụ nghiên cứu cụ thể: Xây dựng mô hình phân lớp "Đối tượng" 5](#_Toc211353160)

[CHƯƠNG III: CƠ SỞ LÝ THUYẾT VÀ PHƯƠNG PHÁP THỰC NGHIỆM 5](#_Toc211353161)

[I. Các lý thuyết, giải pháp và thuật toán liên quan 5](#_Toc211353162)

[II. Phương pháp thực nghiệm 7](#_Toc211353163)

[III. Liên hệ và so sánh với cách đã có 11](#_Toc211353164)

[CHƯƠNG IV: XÂY DỰNG VÀ CÀI ĐẶT HỆ THỐNG 12](#_Toc211353165)

[I. Môi trường cài đặt 13](#_Toc211353166)

[II. Pipeline chính 13](#_Toc211353167)

[CHƯƠNG V: TỔNG KẾT VÀ NGHIỆM THU 22](#_Toc211353168)

[I. Kết quả thử nghiệm 22](#_Toc211353169)

[II. Kết luận và định hướng phát triển 25](#_Toc211353170)

[TÀI LIỆU THAM KHẢO 26](#_Toc211353171)

# lời cảm ơn

Để hoàn thành bài báo cáo thực tập này, trước hết, tôi xin chân thành cảm ơn sự giúp đỡ và tạo điều kiện của Khoa Công nghệ thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.

Đặc biệt, tôi xin gửi lời cảm ơn sâu sắc nhất tới TS. Lê Hoàng Quỳnh. Cô là người giao đề tài, trực tiếp hướng dẫn và cho tôi những góp ý chuyên môn quý báu trong suốt quá trình thực tập và hoàn thiện báo cáo.

Tôi cũng xin gửi lời cảm ơn tới các thầy cô trong Khoa và trong Trường đã truyền đạt cho tôi những kiến thức nền tảng vững chắc, là hành trang quan trọng cho tôi trên con đường sự nghiệp sau này.

Do thời gian và kinh nghiệm còn hạn chế, bài báo cáo không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự đóng góp của quý thầy cô để bài báo cáo được hoàn thiện hơn.

# chương i: tổng quan

## Đặt vấn đề

Trong bối cảnh xã hội phát triển, nhu cầu tìm hiểu pháp luật, đặc biệt về các vấn đề thường xuyên như cư trú, tạm trú, tạm vắng ngày càng tăng cao nhưng lại gặp rào cản lớn do sự phức tạp của hệ thống văn bản pháp luật. Trợ lý ảo thông minh là một giải pháp tiềm năng cho vấn đề này, tuy nhiên, năng lực cốt lõi của chúng là phải hiểu được chính xác ý định của người dùng. Do đó, việc tự động phân loại một câu hỏi – dù là về thủ tục, đối tượng hay điều kiện – vào các danh mục định sẵn là yêu cầu nền tảng để hệ thống có thể cung cấp câu trả lời đúng trọng tâm và hữu ích.

Xuất phát từ yêu cầu thực tiễn đó, đề tài **“Nghiên cứu và Xây dựng Mô hình Phân lớp Câu hỏi trong Lĩnh vực Pháp luật về Cư trú”** được thực hiện. Đề tài tập trung vào việc ứng dụng các mô hình học máy để xây dựng một hệ thống có khả năng tự động phân tích và phân loại các câu hỏi trong lĩnh vực chuyên biệt này. Báo cáo sẽ trình bày toàn bộ quy trình từ xây dựng bộ dữ liệu cho đến thiết kế và đánh giá mô hình, qua đó chứng minh tính khả thi của việc tự động hóa phân tích yêu cầu người dùng, mở đường cho việc xây dựng một hệ thống hỏi đáp pháp luật hoàn chỉnh trong tương lai.

## Bối cảnh thực tập

Trong khuôn khổ kỳ thực tập chuyên ngành tại Khoa Công nghệ thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội, dưới sự hướng dẫn của giảng viên Lê Hoàng Quỳnh, tôi đã được tham gia vào dự án nghiên cứu hướng tới việc xây dựng một trợ lý pháp lý thông minh cho người Việt. Mục tiêu chính của dự án là ứng dụng các kỹ thuật của *Trí tuệ Nhân tạo (AI)* và *Xử lý Ngôn ngữ Tự nhiên (NLP)* để tự động hóa việc giải đáp các thắc mắc pháp luật, giúp người dân dễ dàng tiếp cận thông tin pháp lý một cách chính xác và nhanh chóng.

Nhiệm vụ cụ thể của tôi trong kỳ thực tập là tập trung vào giai đoạn đầu tiên và cũng là nền tảng quan trọng nhất của dự án: xây dựng, xử lý dữ liệu. Các công việc chính tôi đã thực hiện bao gồm:

* **Gán nhãn dữ liệu:** Dựa trên bộ dữ liệu có sẵn và các tiêu chí đã được xác định, tiến hành phân loại và gán nhãn cho từng cặp câu hỏi – câu trả lời để xác định các đặc trưng.
* **Tiền xử lý và chuẩn hóa dữ liệu:** Thực hiện các bước tách câu ghép thành các câu đơn, kiểm tra chính tả, loại bỏ các dữ liệu nhiễu,… nhằm đảm bảo chất lượng của bộ dữ liệu đầu vào.
* **Xây dựng mô hình phân lớp cơ bản:** Từ bộ dữ liệu câu hỏi đã chuẩn bị, huấn luyện một mô hình cho tác vụ phân loại câu hỏi theo bộ nhãn có sẵn.

Kết quả của quá trình này là một bộ dữ liệu đã được tinh chỉnh và một mô hình phân lớp bước đầu, tạo tiền đề vững chắc cho việc phát triển các tính năng phức tạp hơn của trợ lý pháp lý trong tương lai.

# chương ii: phát biểu bài toán

## Kiến trúc bài toán tổng thể

Bài toán tổng quan là phát triển một hệ thống hoàn chỉnh có khả năng tự động phân tích và gán các thuộc tính ngữ nghĩa cho một câu hỏi pháp luật. Quá trình này được thực hiện thông qua hai giai đoạn chính và độc lập: **Xây dựng bộ dữ liệu gán nhãn** và **Xây dựng hệ thống mô hình phân lớp**.

1. **Giai đoạn 1: Xây dựng Bộ dữ liệu Gán nhãn**

Đây là giai đoạn nền tảng, tập trung vào việc chuyển đổi dữ liệu thô từ một tập lớn các câu hỏi pháp luật về cư trú đã được thu thập thành một bộ dữ liệu có cấu trúc để huấn luyện mô hình. Quy trình chính bao gồm việc đọc, hiểu và gán nhãn thủ công cho từng câu hỏi dựa trên một bộ lược đồ các thuộc tính đã được định nghĩa trước. Các bộ nhãn này bao gồm:

* **Tính liên quan:** Yes, No
* **Dạng câu hỏi:** What, How, When, Yes/No, …
* **Đối tượng:** Người VN dưới 14 tuổi, Người VN trên 14 tuổi, Nước ngoài, Khác
* **Lĩnh vực:** Tạm trú, Tạm vắng, Thường trú, Lưu trú
* **Thủ tục:** Đăng ký, Khai báo, Gia hạn, Xóa đăng ký, ...
* **Hoàn cảnh:** Nội địa, Xuất cảnh
* **Điều luật:** Trích xuất các thực thể là điều luật nếu có

Thông qua quá trình này, đầu ra thu được là một bộ dữ liệu chất lượng cao, trong đó mỗi câu hỏi được liên kết với một vector các nhãn tương ứng, tạo thành nguyên liệu cốt lõi cho giai đoạn sau.

1. **Giai đoạn 2: Xây dựng Hệ thống Mô hình Phân lớp**

Giai đoạn tiếp theo là xây dựng **Hệ thống Mô hình Phân lớp**, dựa trên bộ dữ liệu đã được gán nhãn từ **Giai đoạn 1**. Mục tiêu của giai đoạn này là tạo ra các mô hình học máy có khả năng tự động hóa hoàn toàn quy trình phân tích. Quy trình thực hiện bao gồm việc phát triển một hệ thống với nhiều mô hình con, trong đó mỗi mô hình được huấn luyện chuyên biệt để nhận dạng và gán giá trị cho một trong các trường mục tiêu đã xác định, bao gồm: *Tính liên quan, Dạng câu hỏi, Đối tượng, Lĩnh vực, Thủ tục, Hoàn cảnh,* và *Điều luật*.

Sản phẩm cuối cùng của giai đoạn này là một pipeline xử lý tự động và hoàn chỉnh. Khi hệ thống tiếp nhận một câu hỏi mới, pipeline sẽ đưa câu hỏi này qua đồng thời tất cả các mô hình đã được huấn luyện. Ngay sau đó, hệ thống sẽ trả về một cấu trúc dữ liệu đầu ra duy nhất, tổng hợp đầy đủ các thuộc tính ngữ nghĩa mà các mô hình đã dự đoán.

## Nhiệm vụ nghiên cứu cụ thể: Xây dựng mô hình phân lớp "Đối tượng"

Trên cơ sở của bài toán lớn, nhiệm vụ mà tôi thực hiện là tập trung giải quyết một trong các bài toán phân lớp chuyên biệt đó. Cụ thể, tôi sẽ xây dựng mô hình phân lớp cho thuộc tính *Đối tượng.*

Đây là một bài toán phân loại văn bản đa lớp được định nghĩa như sau:

* **Đầu vào (Input):** Một câu hỏi pháp luật dưới dạng văn bản.
* **Đầu ra (Output):** Một nhãn duy nhất được dự đoán từ tập nhãn đã định nghĩa: {Người VN dưới 14 tuổi, Người VN trên 14 tuổi, Người nước ngoài, Không có thông tin}.

Ví dụ, với câu hỏi *"Người nước ngoài có được cấp thẻ tạm trú tại Việt Nam không?"*, mô hình mà tôi xây dựng cần có khả năng phân tích và dự đoán chính xác nhãn đầu ra là Nước ngoài.

Việc giải quyết bài toán này sẽ tạo ra một module có khả năng tự động xác định nhóm chủ thể mà người dùng đang quan tâm, từ đó đóng góp trực tiếp vào mục tiêu chung của bài toán lớn là xây dựng một hệ thống phân tích câu hỏi pháp luật hoàn chỉnh và thông minh.

# chương iii: cơ sở lý thuyết và phương pháp thực nghiệm

Chương này trình bày chi tiết về phương pháp luận được áp dụng để giải quyết bài toán phân loại văn bản, bao gồm các lý thuyết nền tảng, quy trình thực nghiệm và so sánh với các hướng tiếp cận khác.

## Các lý thuyết, giải pháp và thuật toán liên quan

### Tăng cường dữ liệu (Data Augmentation)

Để giải quyết vấn đề dữ liệu trong thực tế thường bị thiếu hụt hoặc mất cân bằng, kỹ thuật tăng cường dữ liệu đã được sử dụng. Phương pháp này cho phép tạo ra các mẫu dữ liệu mới một cách tự động mà không đòi hỏi quá trình gán nhãn thủ công tốn kém thời gian, qua đó làm giàu thêm cho tập dữ liệu ban đầu.

Hai phương pháp chính được lựa chọn để thực hiện việc này. Thứ nhất là **Dịch ngược (Back-translation)**, bao gồm việc dịch một câu từ ngôn ngữ nguồn sang ngôn ngữ đích rồi dịch ngược lại, giúp tạo ra câu mới có ngữ nghĩa tương tự nhưng cấu trúc và từ vựng đa dạng hơn. Thứ hai là **Diễn giải (Paraphrasing)**, tận dụng sức mạnh của các mô hình ngôn ngữ để viết lại một câu theo cách khác trong khi vẫn đảm bảo giữ nguyên ý nghĩa gốc.

### Lọc dữ liệu bằng tương đồng Embedding (Embedding Similarity)

Để đảm bảo chất lượng dữ liệu sau khi tăng cường, tôi sử dụng kỹ thuật so sánh vector embedding của câu gốc và câu mới được tạo ra. Chỉ những cặp câu có độ tương đồng ngữ nghĩa cao (vượt một ngưỡng nhất định) mới được giữ lại, giúp loại bỏ các mẫu dữ liệu nhiễu, sai lệch.

### Mô hình Transformer (PhoBERT & ViLegalBERT):

* 1. **PhoBERT**

Là mô hình ngôn ngữ dựa trên kiến trúc **Transformer**, được huấn luyện trước (pre-trained) trên một tập dữ liệu tiếng Việt khổng lồ. Việc sử dụng PhoBERT cho phép mô hình "thừa hưởng" kiến thức sâu rộng về ngôn ngữ và ngữ cảnh, giúp đạt hiệu quả cao khi tinh chỉnh (fine-tune) cho các tác vụ cụ thể.

* 1. **ViLegalBERT**

Là một biến thể của BERT, được tiếp tục huấn luyện trước (continue-pretraining) trên tập dữ liệu chuyên ngành pháp lý của Việt Nam. Điều này giúp mô hình nắm bắt tốt hơn các thuật ngữ và ngữ cảnh đặc thù của ngành, có thể mang lại hiệu suất vượt trội cho các bài toán trong lĩnh vực này.

### Các kiến trúc bổ trợ (CNN & Attention):

Để nâng cao hiệu suất mô hình, các kiến trúc bổ trợ đã được tích hợp. Cụ thể, **Mạng tích chập (CNN)** được kết hợp sau lớp BERT nhằm trích xuất các đặc trưng cục bộ (local features) một cách hiệu quả, giúp mô hình có khả năng nhận diện các cụm từ quan trọng. Tiếp theo, một lớp **Cơ chế tập trung (Attention)** được thêm vào sau CNN, cho phép mô hình "tập trung" vào những phần thông tin quan trọng nhất đã được trích xuất, từ đó nâng cao đáng kể khả năng phân loại chính xác.

### Giải quyết mất cân bằng dữ liệu (Class Imbalance):

Đồng thời, nhằm giải quyết tình trạng mất cân bằng dữ liệu, hai kỹ thuật đã được áp dụng trong quá trình huấn luyện. Phương pháp **Trọng số lớp (Class-weight)** được sử dụng để gán trọng số cao hơn cho lớp thiểu số khi tính toán hàm mất mát, buộc mô hình phải "quan tâm" hơn đến việc học và phân loại đúng các mẫu thuộc lớp này. Ngoài ra, **Hàm mất mát** **Focal Loss**, một phiên bản cải tiến của Cross-Entropy Loss, cũng được triển khai để giảm bớt sự ảnh hưởng của các mẫu dễ phân loại, giúp mô hình tập trung nguồn lực vào việc học các mẫu khó, vốn thường thuộc về các lớp thiểu số.

### Các kỹ thuật tối ưu hóa huấn luyện

Để đảm bảo mô hình hội tụ tốt và tránh các vấn đề phổ biến như overfitting, một loạt kỹ thuật tối ưu hóa đã được áp dụng. **Early Stopping** được sử dụng để theo dõi hiệu suất và dừng quá trình huấn luyện sớm nếu không có sự cải thiện, giúp tiết kiệm thời gian tính toán. Đồng thời, **Learning Rate Scheduler** kết hợp với **Warmup** được triển khai để tự động điều chỉnh tốc độ học, giúp mô hình khởi đầu ổn định và hội tụ chính xác hơn vào điểm tối ưu. Cuối cùng, để có được một đánh giá khách quan và đáng tin cậy về khả năng tổng quát hóa của mô hình, phương pháp **K-Fold Cross-Validation** đã được sử dụng, thông qua việc huấn luyện và kiểm định mô hình trên nhiều tập dữ liệu con khác nhau rồi lấy kết quả trung bình.

## Phương pháp thực nghiệm

### Xử lý dữ liệu

* 1. ***Dữ liệu thô***

Gồm 1000+ cặp câu hỏi và câu trả lời trong lĩnh vực pháp luật Việt Nam. Dữ liệu được thu thập và tổng hợp thành một tệp duy nhất, bao gồm các trường thông tin chính như *question\_text* (nội dung câu hỏi), *answer* (nội dung câu trả lời) cùng với một số thông tin bổ sung như *post\_time* (thời gian đăng), *detail\_url* (đường dẫn đến câu hỏi), *related\_tag* (Tag liên quan).

* 1. ***Chuẩn hoá và gắn nhãn***

Trong quá trình chuẩn hoá và gắn nhãn, các câu hỏi và câu trả lời phức tạp chứa nhiều ý trước tiên được xử lý bằng cách tách thành các cặp câu hỏi – trả lời đơn, độc lập nhưng vẫn giữ nguyên ngữ cảnh. Sau khi tách và chuẩn hoá, dữ liệu này được gán nhãn thủ công cho các trường mục tiêu như Subject, Action, Type,... nhằm phục vụ cho việc huấn luyện và đánh giá.

Kết quả của quá trình này là một bộ dữ liệu mới với hơn 3600 cặp câu hỏi – trả lời đã được tách thành từng câu đơn và gắn nhãn đầy đủ.

* 1. ***Làm sạch và kiểm tra chính tả***

Qua quan sát ban đầu, dữ liệu còn tồn tại nhiều lỗi như câu rỗng, nội dung không liên quan, các trường thông tin dư thừa, từ bị dính liền (*“tạmtrú”*, *“báotạm”*), dấu câu gắn liền (*“?ai”,* *“!Ai”*) và một số lỗi chính tả phổ biến.

Quá trình làm sạch được tiến hành tuần tự theo các bước.

Trước hết, loại bỏ các dòng trống, câu không mang nội dung pháp lý và những cột không phục vụ trực tiếp cho tác vụ phân lớp lớp *Đối tượng*. Văn bản còn lại được chuẩn hóa bằng cách chuyển toàn bộ về chữ thường, chuẩn hóa Unicode, loại bỏ ký tự đặc biệt và chèn khoảng trắng hợp lý giữa dấu câu và từ liền kề.

Tiếp đó, các câu được tách từ bằng *underthesea* và kiểm tra chính tả dựa trên bộ từ điển tiếng Việt mở rộng (kết hợp *tsdocode/vietnamese-dictionary*, từ điển của *underthesea* và các file *.txt* tự xây dựng). Những từ không nằm trong từ điển *(OOV)* được xem xét là từ sai hoặc dính liền. Sau đó thực hiện chỉnh sửa một cách thủ công, ví dụ: *“tạmtrú”* → *“tạm trú”*, *“thủtục”* → *“thủ tục”*, *“cưtrú”* → *“cư trú”*, *“đawng”* → *“đăng”*.

Sau quá trình làm sạch, tỷ lệ từ ngoài từ điển giảm đi đáng kể, cho thấy mức độ chuẩn hóa cao. Một số ví dụ minh họa:

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  |   **Trước** | **Sau** |
| |  | | --- | | *“Tôi muốn báotạm vắng ở nowi cưtrú.”* |  |  | | --- | |  | | |  | | --- | | *“tôi muốn báo tạm vắng ở nơi cư trú.”* |  |  | | --- | |  | |
| |  | | --- | | *“Thủtục đăgn ký tạmtrú ra sao?”* |  |  | | --- | |  | | *“thủ tục đăng ký tạm trú ra sao?”* |

Kết quả cho thấy dữ liệu sau xử lý có chất lượng tốt hơn, giảm nhiễu và đảm bảo tính nhất quán cho các bước huấn luyện và truy xuất thông tin pháp luật.

* 1. ***Tăng cường dữ liệu***

Áp dụng hai phương pháp **Back-translation** và **Paraphrasing** để sinh dữ liệu mới. Sau đó, **lọc bằng Embedding Similarity** để chọn lọc những mẫu dữ liệu được sinh ra có chất lượng tốt, giữ được ngữ nghĩa gốc.

* + - * 1. **Back-tranlating:**

Phần mở rộng dữ liệu được thực hiện bằng mô hình **VietAI/envit5-translation**, một biến thể của **T5 (Text-to-Text Transfer Transformer)** được huấn luyện song ngữ Anh–Việt. Mô hình có khả năng **tự động nhận diện ngôn ngữ đầu vào** và **dịch sang ngôn ngữ còn lại**, giúp việc dịch xuôi–ngược trở nên nhất quán mà không cần xác định hướng dịch.

Ngoài ra, để tạo ra các câu dịch đa dạng, mô hình EnViT5 được cấu hình sinh văn bản theo cơ chế **sampling kết hợp top-*k* và top-*p***. Cơ chế này giúp sinh ra **nhiều biến thể hợp ngữ cảnh** cho cùng một câu gốc, phục vụ tốt cho mục tiêu tăng cường dữ liệu.

Bảng dưới đây trích từ tập dữ liệu được sinh ra bằng kỹ thuật trên:

|  |  |
| --- | --- |
| **Original** | **Back translated** |
| Tôi hiên đang tạm trú tại một phường này, nay tôi muốn làm tạm trú sang phương khác cho con tôi 12 tuổi. Nay tôi đã nhờ được địa chị của nhà người bạn để con tôi tạm trú ở đó có được không? | Tôi xin giới thiệu một cô bé đang tạm trú tại một khu vực và bây giờ muốn tạm thời chuyển đứa con trai 12 tuổi của tôi sang chỗ khác. Bây giờ tôi có thể hỏi địa chỉ của một người bạn để con tôi ở đó được không? |
| Trẻ mồ côi được nuôi dưỡng tại chùa có được đăng ký thường trú tại chùa hay không? | Có phải trẻ em mồ côi được nuôi tại chùa có quyền đăng ký thường trú tại ngôi chùa? |

Các câu kết quả thể hiện sự khác biệt về cấu trúc và từ vựng, phản ánh hiệu quả của cơ chế sampling trong quá trình sinh dữ liệu.

* + - * 1. **Paraphrasing**

Trong nghiên cứu này, quá trình paraphrasing được thực hiện bằng mô hình **mT5** (checkpoint chieunq/vietnamese-sentence-paraphase), vốn dựa trên kiến trúc **Text-to-Text Transfer Transformer (T5)**. Mọi tác vụ ngôn ngữ đều được chuyển về dạng ánh xạ *văn bản → văn bản*, nên mô hình có thể tự động học cách sinh câu tương đương ngữ nghĩa từ một đầu vào bất kỳ.

Mô hình được cấu hình sinh theo **cơ chế sampling kết hợp top-*k* và top-*p*** giống với **Back-translating** với các tham số đã được điều chỉnh có kiểm soát để vừa giữ được tính mạch lạc và đồng nghĩa ngữ nghĩa với câu gốc.

Ví dụ trích từ tập dữ liệu sinh ra:

|  |  |
| --- | --- |
| **Original** | **Paraphrased** |
| Cho tôi hỏi tôi là người đăng ký tạm trú cho Peter có đúng không? | Tôi có thể xác nhận rằng tôi là người đăng ký tạm trú cho Peter không? |
| Thời hạn đăng ký tạm trú cho người nước ngoài là bao lâu? | Người nước ngoài phải đăng ký tạm trú trong bao lâu? |

* + - * 1. **Lọc bằng Embedding Similarity**

Trong quá trình mở rộng dữ liệu bằng kỹ thuật **Back-Translation** và **Paraphrasing**, mô hình **Embedding Similarity** được sử dụng để đánh giá mức độ tương đồng ngữ nghĩa giữa câu gốc và câu sinh ra, giúp loại bỏ các câu trùng lặp hoặc sai lệch ý. Hệ thống sử dụng mô hình *keepitreal/vietnamese-sbert*, một biến thể của **Sentence-BERT** cho tiếng Việt, để mã hoá câu thành vector ngữ nghĩa và tính **độ tương đồng cosine** giữa chúng.

Trong thực nghiệm, các ngưỡng tương đồng được xác định cụ thể cho từng kỹ thuật:

* **Back-Translation:** giữ lại các câu có độ tương đồng trong khoảng **0.80 – 0.92**, loại bỏ những câu quá gần (thiếu đa dạng) hoặc quá xa (sai nghĩa).
* **Paraphrasing:** áp dụng ngưỡng **0.85 – 0.92** để đảm bảo các câu được sinh vẫn giữ ý nghĩa nhưng thể hiện bằng cấu trúc diễn đạt khác.

Phương pháp này cho hiệu quả cao trong việc lọc dữ liệu sinh tự động, giúp tập dữ liệu thu được vừa đa dạng ngôn ngữ, vừa đồng nhất ngữ nghĩa, từ đó cải thiện đáng kể chất lượng huấn luyện của các mô hình xử lý ngôn ngữ tự nhiên sau này.

### Huấn luyện và tinh chỉnh mô hình

Dựa trên bộ dữ liệu đã chuẩn bị, tôi tiến hành huấn luyện và so sánh hiệu suất của các mô hình khác nhau.

* 1. ***Mô hình 1: PhoBERT kết hợp Class-weight và FocalLoss***

Mô hình được xây dựng trên PhoBERT-base và được huấn luyện trên bộ dữ liệu gốc chưa được tăng cường khiến cho phân bố nhãn không cân bằng. Để xử lý vấn đề này, hai kỹ thuật chính được áp dụng:

* **Focal Loss kết hợp Class-weight:** giúp mô hình tập trung hơn vào các mẫu khó và giảm ảnh hưởng của lớp chiếm ưu thế. Hàm loss có dạng:

Trong đó, trọng số lớp (class-weight) được thêm vào theo tần suất nghịch đảo của từng nhãn để giảm ảnh hưởng của lớp chiếm ưu thế.

* **Stratified 10-Fold Cross Validation:** Dữ liệu được chia thành 10 phần theo tỷ lệ lớp, đảm bảo mỗi fold phản ánh đúng phân bố nhãn, giúp đánh giá mô hình khách quan và giảm phương sai kết quả.

Kết quả huấn luyện sẽ dùng để đánh giá hiệu suất của mô hình trên tập dữ liệu kém chất lượng, đồng thời cho thấy tính hiệu quả của việc áp dụng các kỹ thuật sử dụng **Trọng số lớp** và **FocalLoss** để giải quyết mất cân bằng dữ liệu.

* 1. ***Mô hình 2: PhoBERT sử dụng CrossEntropyLoss***

So với mô hình đầu tiên, mô hình PhoBERT-base lần này được huấn luyện trên bộ dữ liệu sau khi tăng cường và đạt mức độ cân bằng nhất định. Do vậy, mô hình sẽ chỉ cần sử dụng **CrossEntropyLoss** thông thường kết hợp các kỹ thuật tối ưu hoá tốc độ học và giảm thiểu overfitting như **Linear Scheduler**, **EarlyStopping**.

Kết quả huấn luyện sẽ dùng để đánh giá hiệu suất của mô hình trên tập dữ liệu đã cân bằng. Qua đó phản ảnh được tác động của việc tăng cường dữ liệu trong huấn luyện.

* 1. ***Mô hình 3: ViLegalBERT fine-tune***

Trong giai đoạn này, mô hình **htdung167/ViLegalBERT-v0**được sử dụng thay thế cho PhoBERT nhằm cải thiện hiệu quả trên dữ liệu pháp luật.

Quy trình fine-tune gần như được giữ nguyên như với PhoBERT, gồm các kỹ thuật:

* **Tối ưu hóa:** sử dụng **AdamW** kết hợp d**scheduler tuyến tính** có *warm-up ratio = 0.1* để điều chỉnh tốc độ học ổn định và EalyStopping để tránh overfitting.
* **Hàm mất mát:** áp dụng **CrossEntropyLoss** cơ bản.
* **Tham số huấn luyện:** Tăng nhẹ *learning\_rate* giúp hội tụ nhanh và ổn định.

Kết quả fine-tune sẽ phản ánh hiệu suất phân loại văn bản pháp luật trên mô hình được pre-trained trên dữ liệu pháp lý so với mô hình được pre-trained trên ngôn ngữ tổng quát như PhoBERT.

* 1. ***Mô hình 4: Kiến trúc phức hợp ViLegalBERT + CNN + Attention***

Mô hình được xây dựng dựa trên kiến trúc **ViLegalBERT** tương tự như **Mô hình 3**. Tuy nhiên, hay vì chỉ fine-tune trực tiếp như mô hình cơ sở, cấu trúc được mở rộng thêm hai thành phần xử lý đặc trưng song song nhằm tăng khả năng biểu diễn ngữ nghĩa và trích chọn thông tin cục bộ.

Các kiến trúc, kỹ thuật chính được áp dụng:

* **Bộ mã hóa ngữ cảnh (Contextual Encoder):** Sử dụng ViLegalBERT để thu được biểu diễn ngữ nghĩa theo ngữ cảnh cho từng token trong câu đầu vào. Đầu ra là tensor kích thước [*batch\_size*, *sequence\_length*, *hidden\_size*], trong đó *hidden\_size = 768*.
* **Khối CNN (Convolutional Feature Extractor):** Ba lớp **1D Convolution** được áp dụng trên trục chiều dài chuỗi (sequence length) với các *kernel size = 2, 3, 4*, giúp mô hình phát hiện các cụm từ và mẫu ngữ pháp ngắn trong văn bản.  
  Sau mỗi lớp, hàm kích hoạt **ReLU** và phép **max-pooling** được dùng để chọn đặc trưng nổi bật nhất theo từng bộ lọc. Các đầu ra được **nối (concatenate)** để tạo thành vector đặc trưng tổng hợp *[batch\_size, 192].*
* **Cơ chế Attention (Global Context Integration):** Một tầng **Multi-Head Attention (4 heads)** được thêm vào để tái gán trọng số cho từng token, giúp mô hình tập trung vào các từ khóa pháp lý quan trọng có ảnh hưởng lớn đến phân loại. Kết quả trung bình theo chiều chuỗi *(mean pooling)* cho ra vector toàn cục *[batch\_size, 768].*
* **Tầng kết hợp và phân lớp (Fusion & Classification):** Hai nhánh đặc trưng từ CNN và Attention được nối lại *(torch.cat)* và đi qua **fully connected layer** để đưa ra xác suất phân loại cuối cùng.
* Hàm mất mát sử dụng là **CrossEntropyLoss** có **Class-weight** nhẹ để tăng khả năng dự đoán cho các nhãn không phổ biến, sau đó tối ưu bằng **AdamW**, với **learning rate cao (5e-5)** nhằm tăng tốc độ hội tụ và ổn định trên miền dữ liệu quen thuộc.

Mô hình **ViLegalBERT + CNN + Attention** được kỳ vọng sẽ vượt trội hơn mô hình **ViLegalBERT fine-tune** thông thường nhờ khả năng mở rộng trong việc trích chọn và tổng hợp đặc trưng ở nhiều cấp độ khác nhau.

## Liên hệ và so sánh với cách đã có

Phương pháp của tôi không chỉ là việc áp dụng một mô hình có sẵn, mà là một quy trình có hệ thống từ việc làm giàu dữ liệu một cách thông minh cho đến việc lựa chọn và cải tiến mô hình chuyên ngành để đạt được hiệu suất tối ưu nhất cho bài toán đặc thù. Các điểm nổi bật chính có thể nói đến như:

### Sự đổi mới trong xử lý dữ liệu

Thay vì chỉ áp dụng các kỹ thuật tiền xử lý cơ bản, tôi tập trung giải quyết tận gốc vấn đề thiếu hụt và mất cân bằng dữ liệu thông qua một chiến lược tăng cường dữ liệu tiên tiến. Việc kết hợp **Back-translation** và **Paraphrasing** được lựa chọn một cách có chủ đích:

* **Back-translation** có xu hướng tạo ra sự đa dạng về mặt cú pháp, thay đổi cấu trúc và trật tự câu một cách tự nhiên.
* **Paraphrasing** lại mạnh về việc tạo ra sự đa dạng về mặt từ vựng, sử dụng các từ và cụm từ đồng nghĩa để diễn đạt lại ý tưởng.

Bằng cách kết hợp cả hai, tôi tạo ra một tập dữ liệu tăng cường vừa đa dạng về cấu trúc câu, vừa phong phú về cách dùng từ. Chiến lược này giúp mô hình trở nên mạnh mẽ, có khả năng khái quát hóa tốt hơn và không bị overfitting theo những mẫu câu cố định. Quan trọng hơn, bước lọc bằng **Embedding Similarity** hoạt động như một cơ chế kiểm soát chất lượng, đảm bảo rằng chỉ những dữ liệu tổng hợp thực sự phù hợp mới được đưa vào huấn luyện, vượt trội hơn hẳn so với việc chỉ nhân bản dữ liệu một cách cơ học.

### Nâng cao kiến thức chuyên môn với ViLegalBERT

Trong khi việc tinh chỉnh PhoBERT đã cho thấy hiệu quả mạnh mẽ, mô hình này vẫn là mô hình ngôn ngữ tổng quát. Để vượt qua giới hạn này, nghiên cứu đã áp dụng **ViLegalBERT**, một mô hình được huấn luyện chuyên sâu trên kho dữ liệu pháp lý tiếng Việt. Sự huấn luyện chuyên biệt này giúp ViLegalBERT có "am hiểu" kiến thức luật pháp vượt trội, từ đó nhận diện các thực thể, mối quan hệ và ý định trong câu hỏi pháp lý một cách chính xác hơn đáng kể so với PhoBERT.

### Tối ưu hóa bằng kiến trúc chuyên biệt

Nghiên cứu mở rộng ViLegalBERT bằng cách kết hợp **CNN** và **Attention** nhằm khai thác đồng thời hai đặc trưng: **CNN** trích xuất các mẫu ngữ nghĩa cục bộ (cụm từ pháp lý), trong khi **Attention** giúp mô hình tập trung vào các thông tin quan trọng toàn cục. Kiến trúc này được thiết kế riêng cho bài toán pháp luật, kỳ vọng mang lại độ chính xác và khả năng tổng quát hóa cao hơn so với mô hình ViLegalBERT fine-tune thông thường.

# chương iv: xây dựng và cài đặt hệ thống

Chương này trình bày chi tiết về môi trường cài đặt, thư viện sử dụng và ứng dụng các kỹ thuật, mô hình đã được đề cập bên trên trong dự án.

## Môi trường cài đặt

Hệ thống được thiết lập trong môi trường Python (phiên bản 3.12.11) với các thư viện phục vụ cho xử lý dữ liệu, huấn luyện mô hình học sâu và trực quan hóa. Bộ thư viện bao gồm ba nhóm chính:

1. **Xử lý dữ liệu và tiền xử lý:** pandas, numpy, re, json, os, underthesea hỗ trợ đọc, làm sạch và chuẩn hóa dữ liệu tiếng Việt.
2. **Học sâu và mô hình ngôn ngữ:** torch, torch.nn, torch.utils.data, transformers (gồm các mô-đun như AutoModel, AutoTokenizer, MT5ForConditionalGeneration) dùng để xây dựng, huấn luyện và đánh giá mô hình học sâu dựa trên kiến trúc Transformer.
3. **Đánh giá và trực quan hóa:** scikit-learn cung cấp các công cụ tính toán chỉ số đánh giá (accuracy, F1-score, confusion matrix); seaborn, matplotlib dùng để biểu diễn kết quả trực quan; tqdm theo dõi tiến trình huấn luyện.

Môi trường được khuyến nghị chạy trên GPU hỗ trợ CUDA để tăng tốc huấn luyện. Việc cài đặt các thư viện được thực hiện thông qua *pip* hoặc *conda*.

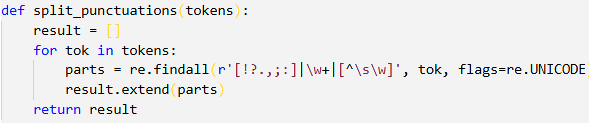
## Pipeline chính

### Tiền xử lý dữ liệu

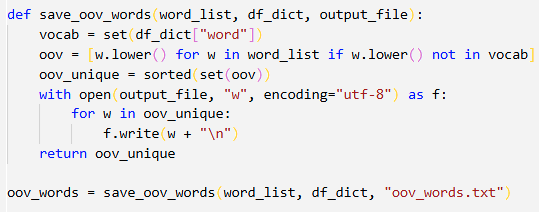
Đầu tiên, tôi sử dụng hàm *word\_tokenize* của thư viện *underthesea* để tách văn bản thành các từ Tiếng Việt có nghĩa.



Sau đó thực hiện tách các dấu câu (như `.` `,` `?`) và các ký tự đặc biệt ra khỏi các từ trong một danh sách các token bằng hàm *split\_punctuations*.



Sau khi thu được các token sạch, tôi tiến hành tra cứu các token trong bộ từ điển và trả về file *oov\_words.txt* chứa các từ không nằm trong từ điển.

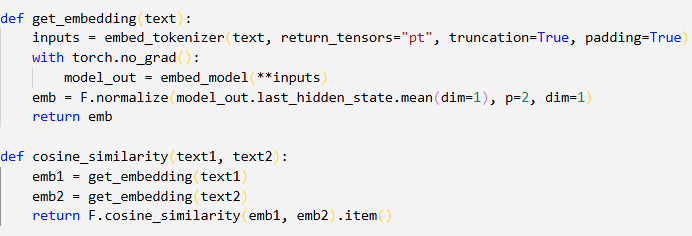


Cuối cùng, dựa theo danh sách các từ được lọc tôi thực hiện chỉnh sửa thủ công trên bộ dữ liệu gốc và thu được bộ dữ liệu mới không còn lỗi sai chính tả.

### Tăng cường dữ liệu

* 1. ***Embedding Similarity***

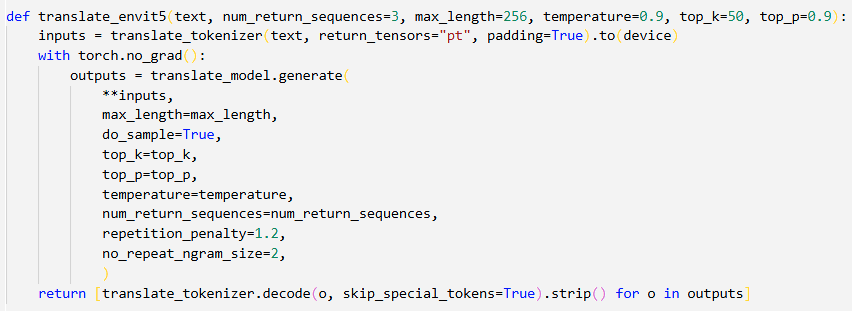
Hàm tính cosine\_similarity sử dụng model embedding *keepitreal/vietnamese-sbert* để mã hoá văn bản đầu vào



* 1. ***Back-traslating***

Sử dụng mô hình *VietAI/envit5-translation* để sinh các văn bản dịch ngược. Thiết lập các tham số để mô hình sinh ra các văn bản tự nhiên và không lặp lại với văn bản gốc:

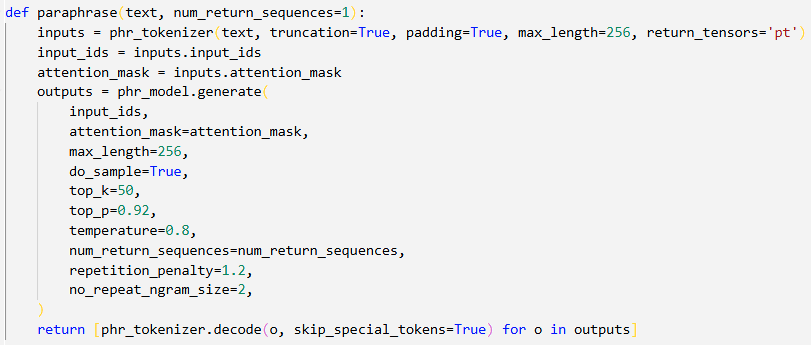
* *do\_sample=True:* đặt cơ chế sampling
* *top\_k=50:* giới hạn dự đoán trong 50 từ có xác suất cao nhất
* *top\_p=0.9:* chọn các từ sao cho tổng xác suất tích lũy đạt 90%
* *temperature=0.9:* điều chỉnh độ ngẫu nhiên
* *no\_repeat\_ngram=2:* tránh lặp cụm từ liên tiếp
* *repetition\_penalty=1.2:* hệ số phạt khi bị lặp lại



Sau đó, sử dụng hàm *translate\_envit5* thực hiện dịch ngược và sinh ra một số lượng dữ liệu mới được chỉ định từ 1 dữ liệu gốc và kết hợp kiểm tra *cosine similarity* để giữ lại các câu phù hợp (trong khoảng 0.8 – 0.92).

* 1. ***Paraphrasing***

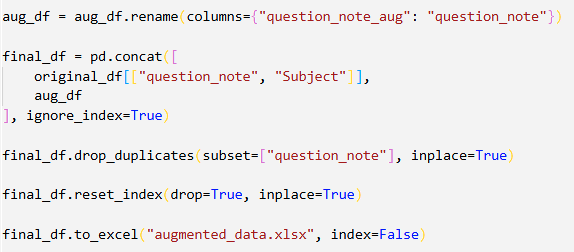
Với **Paraphrasing**, tôi có hàm *paraphrase* với các tham số được thiết lập tương tự như **Back-translating**. Kết quả trả về là 1 danh sách *n* các văn bản được sinh ra từ dữ liệu gốc.



Dữ liệu sinh ra sau đó sẽ được lọc bằng *cosine similarity* chỉ giữ lại các giá trị trong khoảng 0.85 – 0.92.

* 1. ***Gộp dữ liệu***

Sau khi dữ liệu được sinh ra từ 2 phương pháp **Dịch ngược** và **Diễn giải** sẽ được gộp vào với dữ liệu gốc để tạo ra bộ dữ liệu mới đã được cân bằng.

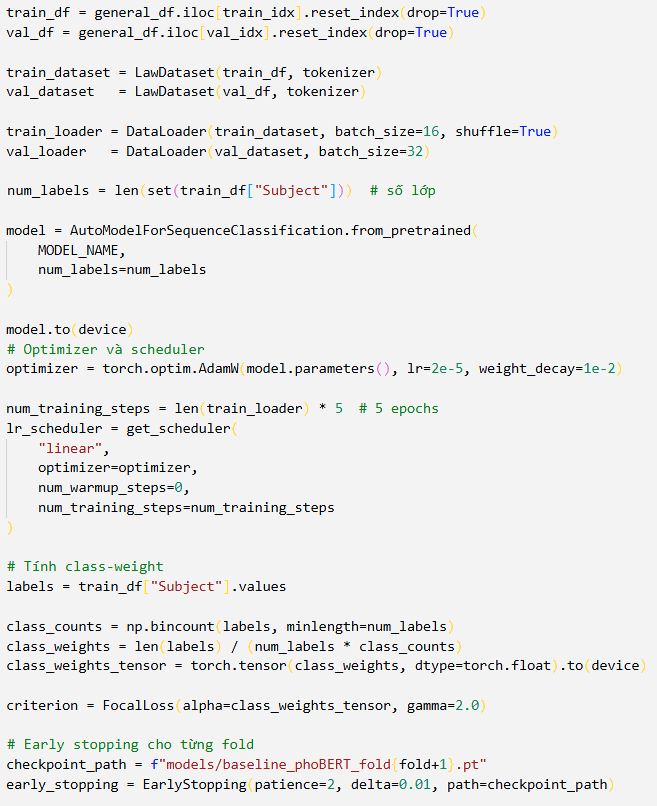


### Huấn luyện và tinh chỉnh mô hình

* 1. ***Mô hình 1: PhoBERT-base trên tập dữ liệu gốc***

Trước tiên, tôi thiết lập vòng lặp huấn luyện với 5 epochs và 10 folds. Tại mỗi fold tôi thực hiện bước tạo DataLoader; khai báo model, Optimizer, Scheduler và EarlyStopping; tính lại class-weight cho tập train. Các tham số huấn luyện được cài đặt như sau:

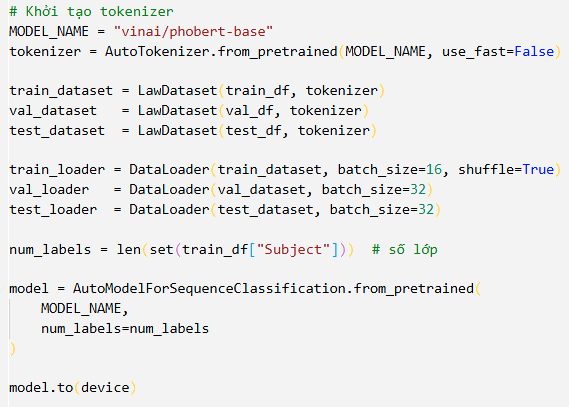
* *batch\_size:* 16 với tập train và 32 với tập val
* *learning\_rate=2e-5:* tốc độ học thấp để tránh phá vỡ kiến thức mô hình gốc
* *weight\_decay=1e-2:* ngăn chặn overfitting
* *num\_training\_steps=len(train\_loader)\*5:* tính toán việc giảm *learning-rate*

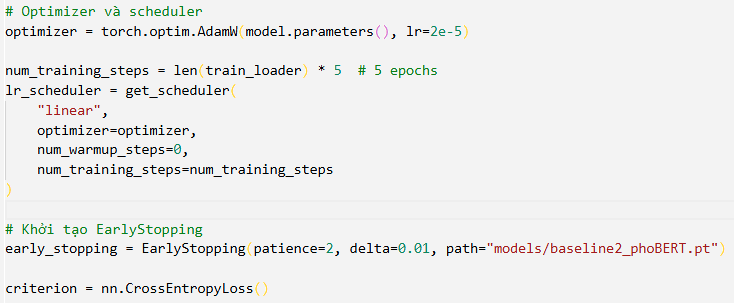


Sau khi thiết lập các mô hình và điều kiện cần thiết, tôi sẽ thực hiện vòng lặp huấn luyện trên 5 epochs và lưu mô hình tốt nhất vào *checkpoint\_path.*

* 1. ***Mô hình 2: PhoBERT-base trên tập dữ liệu đã cân bằng***

Khai báo mô hình và thiết lập tham số tương tự như **Mô hình 1**. Hàm mất mát được thiết lập về lại CrossEntropyLoss.





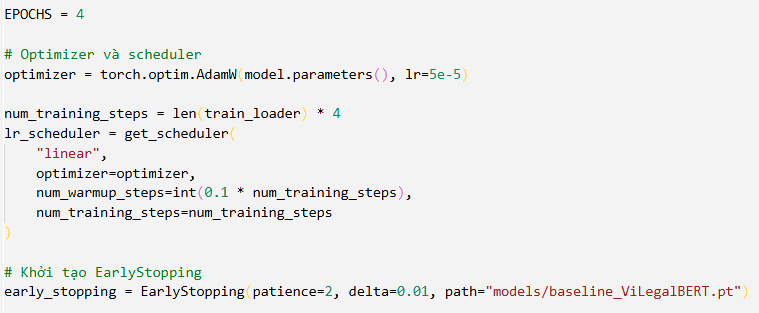
Mô hình tốt nhất trong quá trình huấn luyện được lưu vào đường dẫn *models/baseline2\_phoBERT.pt.*

* 1. ***Mô hình 3: ViLegalBERT fine-tune***

Sử dụng mô hình base là *htdung167/ViLegalBERT-v0* từ thư viện huggingface thay cho mô hình PhoBERT ban đầu.

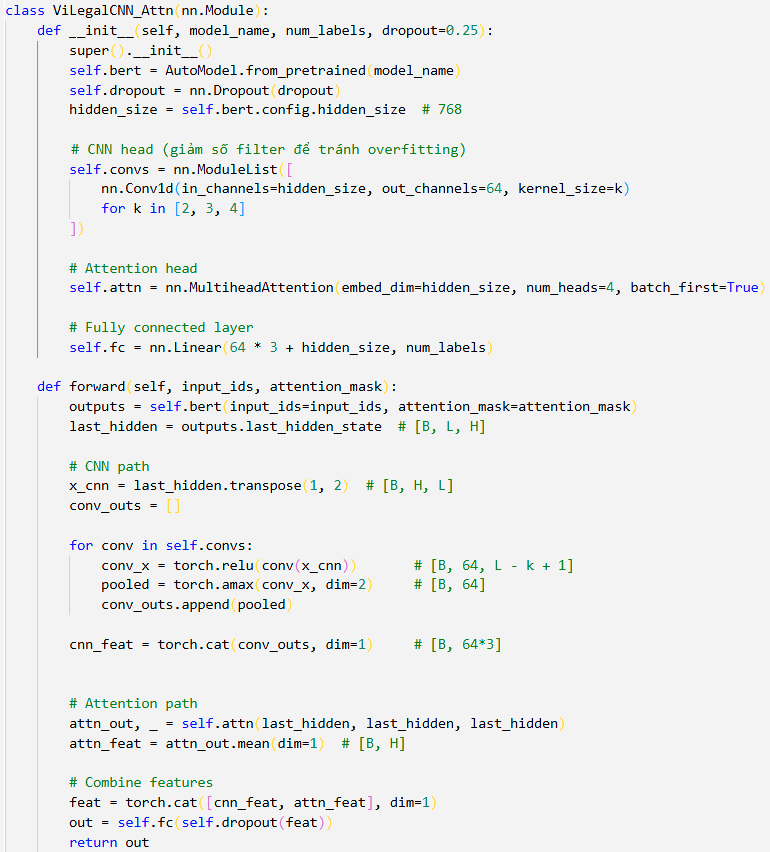


Giảm số lượng epoch và đặt *learning\_rate=5e-5* để tăng tốc độ hội tụ. Ngoài ra đặt *num\_warmup\_steps=10%* để kiểm soát tốc độ học giúp cho mô hình ổn định.



Sau đó lưu mô hình tốt nhất vào *models/baseline\_ViLegalBERT.pt*

* 1. ***Mô hình 4: ViLegalBERT + CNN + Attention***



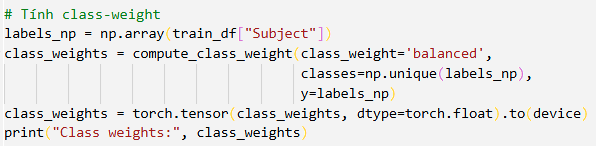
Mô hình ViLegalBERT được trích lấy phần encoder và thêm vào 3 bộ phận:

* Khối CNN:
* *in\_channels=hidden\_size*: đầu vào mỗi từ là vector 768 chiều.
* *out\_channels=64:* mỗi kernel tạo 64 đặc trưng*.*
* *kernel\_size = 2, 3, 4:* tương ứng với n-gram 2, 3, 4 từ.

→ Có tổng cộng 64 \* 3 = 192 đặc trưng được sinh ra từ khối CNN.

* Khối Attention:
* *embed\_dim=768:* kích thước embedding từ BERT.
* *num\_heads=4:* chia không gian đặc trưng thành 4 phần song song để học các mối quan hệ khác nhau giữa các từ.
* *batch\_first=True:* cho phép batch nằm ở trục đầu tiên của tensor [B, L, H].
* Lớp Fully Connected: Kết hợp 2 vector đặc trưng cục bộ của CNN và vector ngữ cảnh toàn cục của Attention để dự đoán xác suất cho *n\_labels* lớp.

Sau đó tôi tính lại **Class-weight** để đảm bảo mô hình chú ý tới các nhãn có ít dữ liệu hơn.



Cuối cùng thực hiện huấn luyện kiến trúc mới trên vòng lặp huấn luyện với các tham số điều kiện tương tự như **mô hình 3** và lưu mô hình tốt nhất vào checkpoint.

# chương v: tổng kết và nghiệm thu

## Kết quả thử nghiệm

Báo cáo đánh giá hệ thống trên cả 2 quá trình: **Xử lý dữ liệu** và **Huấn luyện mô hình**

### Xử lý dữ liệu

Quá trình xử lý đã loại bỏ đi các dữ liệu rỗng, không liên quan, sửa lại 118 từ Tiếng Viết sai chính tả và làm thay đổi phân bổ dữ liệu giữa các nhãn. Sự thay đổi này được thể hiện rõ rệt trong bảng và biểu đồ cột dưới đây:

Table : Thống kê dữ liệu trước & sau xử lý

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **VN\_lớn\_hơn\_14** | **VN\_bé\_hơn\_14** | **NN** | **Không có thông tin** |
| **Trước** | 837 | 281 | 32 | 11 |
| **Sau** | 816 | 446 | 419 | 249 |

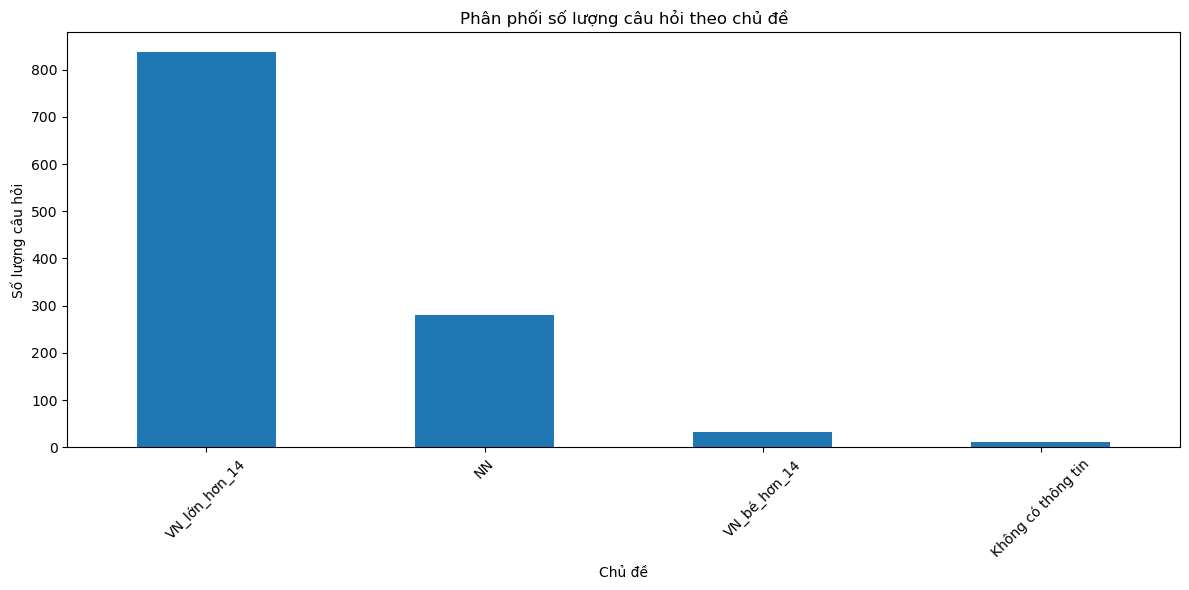


Figure : Phân bổ dữ liệu gốc

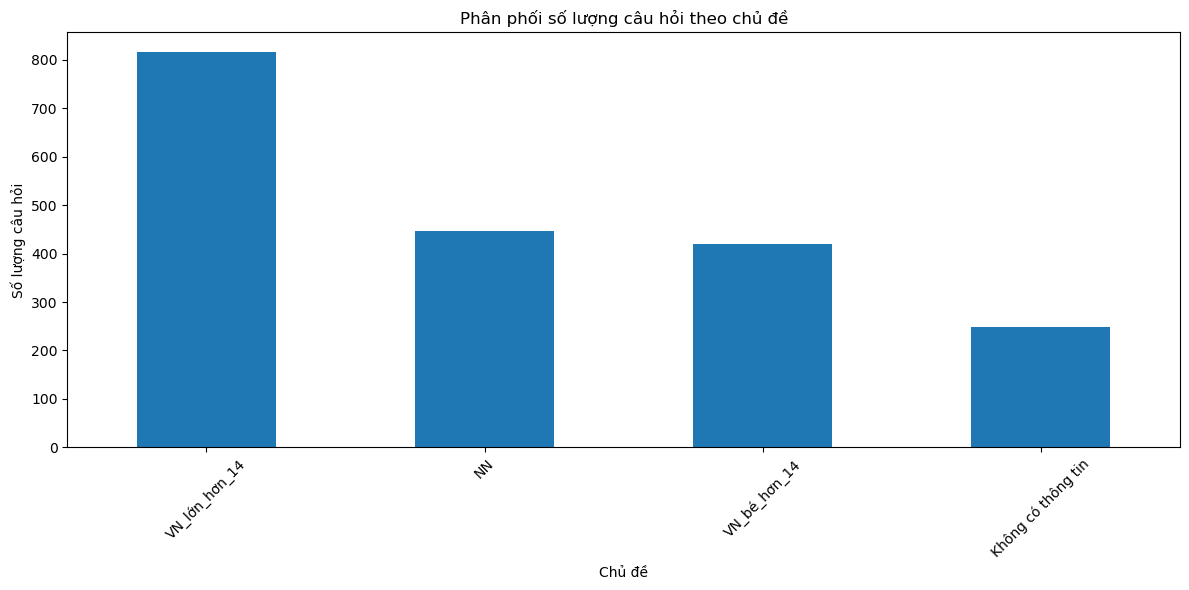


Figure : Phân bổ dữ liệu tăng cường

### Huấn luyện mô hình

Tôi tiến hành đánh giá 4 mô hình bằng cách so sánh kết quả huấn luyện của các mô hình với nhau dựa trên một số tiêu chí nhất định:

* Chất lượng bộ dữ liệu huấn luyện (Mô hình 1, 2)
* Chọn mô hình huấn luyện trước (Mô hình 2, 3, 4)
  1. ***Đánh giá trên chất lượng bộ dữ liệu huấn luyện***

Kết quả huấn luyện của **Mô hình 1** trên 10 folds và **Mô hình 2** được đánh giá bằng các chỉ số *accuracy, precision, recall, f1-score* chi tiết trong bảng sau:

Table : So sánh kết quả huấn luyện Mô hình 1 và 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***precision*** | ***Recall*** | ***F1-score*** | ***Accuracy*** |
| **Mô hình 1 (tính trung bình)** | 0.68 | 0.85 | 0.72 | 0.87 |
| **Fold 1** | 0.9335 | 0.8009 | 0.8226 | 0.906 |
| **Fold 2** | 0.6228 | 0.935 | 0.7 | 0.8632 |
| **Fold 3** | 0.6334 | 0.8486 | 0.6888 | 0.8632 |
| **Fold 4** | 0.6275 | 0.9290 | 0.7072 | 0.8632 |
| **Fold 5** | 0.4880 | 0.4913 | 0.4797 | 0.8462 |
| **Fold 6** | 0.6737 | 0.9410 | 0.7421 | 0.8632 |
| **Fold 7** | 0.7499 | 0.8694 | 0.7432 | 0.8889 |
| **Fold 8** | 0.6873 | 0.9349 | 0.7498 | 0.8803 |
| **Fold 9** | 0.6512 | 0.8635 | 0.6878 | 0.8718 |
| **Fold 10** | 0.7897 | 0.9556 | 0.8308 | 0.9231 |
| **Mô hình 2** | 0.9690 | 0.9726 | 0.9707 | 0.9655 |

Qua quan sát có thể thấy **Mô hình 1** chứng minh hiệu quả của **Focal Loss + Class Weight** trong việc giảm bias lớp lớn, song hiệu suất vẫn bị giới hạn bởi sự mất cân bằng dữ liệu. Trong khi đó, **Mô hình 2**, nhờ tăng cường và cân bằng dữ liệu, đạt được độ chính xác cao, ổn định và tổng quát hơn, dù chỉ dùng CrossEntropyLoss đơn giản.

Như vậy có thể kết luận rằng trong điều kiện dữ liệu pháp luật, việc xử lý cân bằng dữ liệu đầu vào quan trọng hơn việc tinh chỉnh hàm mất mát. Ngoài ra, **Focal Loss** chỉ nên dùng khi không thể cân bằng dữ liệu; còn nếu dữ liệu được tăng cường hợp lý, **CrossEntropyLoss** là lựa chọn hiệu quả và ổn định hơn.

* 1. ***Đánh giá trên mô hình huấn luyện trước***

Kết quả huấn luyện của Mô hình 2, 3, 4 được thu thập trên các chỉ số như: accuracy, macro-f1 (m-f1), weighted-f1 (w-f1), params, throughput (sample/sec) để so sánh hiệu suất tổng thể cả về độ chính xác lẫn tốc độ xử lý và kích thước mô hình.

Table : So sánh hiệu suất mô hình 2, 3, 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mô hình | acc | m-f1 | w-f1 | params | sample/sec |
| 2 | 0,9655 | 0,9707 | 0,9655 | 135M | 36,46 |
| 3 | 0,9690 | 0,9736 | 0,9690 | 278M | 35,42 |
| 4 | 0,9724 | 0,9766 | 0,9724 | 280M | 32,92 |

Ba mô hình được huấn luyện trên cùng tập dữ liệu câu hỏi pháp luật cho kết quả đều rất cao, nhưng có sự khác biệt rõ về hiệu năng và chi phí tính toán. PhoBERT fine-tune đạt **accuracy 0.9655**, **macro-F1 0.9707**, với **135M tham số** và tốc độ **36.46 mẫu/giây**, cho thấy hiệu quả tốt so với kích thước nhỏ. ViLegalBERT fine-tune cải thiện nhẹ lên **0.9690 / 0.9736**, nhưng số tham số tăng gấp đôi (**278M**) và tốc độ giảm nhẹ. Mô hình ViLegalBERT kết hợp **CNN + Attention** tiếp tục đạt mức cao nhất (**accuracy 0.9724**, **macro-F1 0.9766**), song tốc độ giảm còn **32.92 mẫu/giây** do kiến trúc phức tạp hơn.

Kết luận, khi áp dụng vào tác vụ phân loại văn bản pháp luật, cả ba mô hình đều đạt hiệu suất cao, cho thấy khả năng hiểu và xử lý ngữ nghĩa pháp lý tốt. **PhoBERT** có ưu thế về tốc độ và tính gọn nhẹ, phù hợp triển khai thực tế trên tập dữ liệu nhỏ. **ViLegalBERT** và **ViLegalBERT + CNN + Attention** cho độ chính xác cao hơn, phù hợp cho giai đoạn mở rộng hoặc các hệ thống chuyên sâu yêu cầu mức hiểu ngữ nghĩa pháp lý cao.

## Kết luận và định hướng phát triển

### 1. Kiến thức và kỹ năng đạt được

Quá trình triển khai và đánh giá các mô hình giúp nắm vững nhiều kỹ năng cốt lõi trong xử lý ngôn ngữ tự nhiên (NLP) cho tiếng Việt, đặc biệt là trong miền pháp luật. Cụ thể:

* Hiểu rõ quy trình **fine-tuning** các mô hình ngôn ngữ tiền huấn luyện như *PhoBERT* và *ViLegalBERT*.
* Vận dụng thành thạo các kỹ thuật **tối ưu tham số**, **cross-validation**, **focal loss** và **class-weight** để cân bằng dữ liệu và tăng độ ổn định.
* Biết cách **phân tích, so sánh hiệu năng mô hình** dựa trên các chỉ số như *accuracy*, *macro-F1*, *weighted-F1*, *params*, *tốc độ huấn luyện,..*.
* Nâng cao kỹ năng **kết hợp kiến trúc nâng cao** như *CNN* và *Attention* để cải thiện khả năng trích xuất đặc trưng ngữ nghĩa.
* Tích lũy kinh nghiệm **đánh giá hiệu quả mô hình trên dữ liệu nhỏ** và nhận diện vấn đề *overfitting* trong bài toán chuyên ngành.

### 2. Hạn chế và bài học kinh nghiệm

* Bộ dữ liệu hiện tại còn **quy mô nhỏ**, dẫn đến giới hạn khả năng tổng quát hóa.
* Mô hình có xu hướng **phụ thuộc ngữ cảnh huấn luyện**, khó mở rộng sang các lĩnh vực pháp lý khác.
* Việc tăng độ phức tạp mô hình (như ViLegalBERT + CNN + Attention) giúp cải thiện F1 nhưng làm giảm tốc độ suy luận, cho thấy cần cân nhắc giữa hiệu năng và tính triển khai thực tế.

### 3. Phương hướng phát triển và hoàn thiện giải pháp

* **Mở rộng và làm giàu dữ liệu** bằng cách thu thập, gán nhãn thêm các văn bản pháp luật thuộc nhiều lĩnh vực khác nhau.
* Áp dụng **DAPT (Domain-Adaptive Pretraining)** để tinh chỉnh mô hình ngôn ngữ theo miền pháp lý.
* Kết hợp **Retrieval-Augmented Generation (RAG)** hoặc cơ chế truy xuất điều luật để tăng tính chính xác ngữ cảnh.
* **Tối ưu hóa mô hình nhẹ** (quantization, distillation) để dễ triển khai nội bộ.
* Đánh giá trên **bài toán thực tế**, hướng tới xây dựng **trợ lý pháp lý tự động** có khả năng phân loại và gợi ý căn cứ pháp luật chính xác.

# tài liệu tham khảo

[1] C. T. Ngo, T. T. H. Nguyen, T. D. Tran, and T. T. Do, "A Text Classification for Vietnamese Feedback via PhoBERT-Based Deep Learning" trong *Hội nghị Quốc tế lần thứ 12 về Tri thức và Kỹ thuật Hệ thống (12th International Conference on Knowledge and Systems Engineering - KSE)*, 2020, trang 1-6, doi: 10.1109/KSE50997.2020.9287847.

[2] D. Q. Nguyen, D. Q. Tran, and D. Q. Nguyen, "Empirical Study of Text Augmentation on Social Media Text in Vietnamese" trong *Kỷ yếu Hội nghị Chương Bắc Mỹ của Hiệp hội Ngôn ngữ học Tính toán: Công nghệ Ngôn ngữ Con người (Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - NAACL-HLT)*, 2021, trang 4310-4316.

|  |  |
| --- | --- |
|  | |
| Ý kiến đánh giá | |
| ……………………………………….…………………………  ……………………………………….…………………………  ……………………………………….…………………………  ……………………………………….…………………………  ……………………………………….…………………………  ……………………………………….…………………………  ……………………………………….…………………………  ……………………………………….…………………………  ……………………………………….…………………………  ……………………………………….………………………… | |
| Điểm số: ……. Điểm chữ: ………… | |
|  | Hà Nội, ngày tháng năm 20 .  Giảng viên đánh giá  (Ký, ghi rõ họ tên) |