Ứng dụng AI kiểm duyệt nội dung văn bản tiếng Việt

# Danh mục các ký hiệu, các chữ viết tắt

| STT | Từ viết tắt | Từ gốc | Tiếng Việt |
| --- | --- | --- | --- |
| 1 | AI | Artificial Intelligence | Trí tuệ nhân tạo |
| 2 | API | Application Programing Interface | Giao diện lập trình ứng dụng |
| 3 | NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Mở đầu

Trong bối cảnh số hóa và toàn cầu hóa hiện nay, việc kiểm duyệt nội dung trên các nền tảng trực tuyến đã trở thành một nhu cầu cấp thiết đối với các tổ chức chính trị - xã hội. Với sự phát triển mạnh mẽ của công nghệ trí tuệ nhân tạo (AI), việc áp dụng các giải pháp AI vào kiểm duyệt nội dung văn bản tiếng Việt không chỉ giúp tối ưu hóa quy trình kiểm duyệt mà còn đảm bảo tính chính xác và nhanh chóng. Điều này đặc biệt quan trọng trong việc duy trì môi trường trao đổi thông tin lành mạnh, bảo vệ quyền lợi và an ninh của các hội viên.

Tính năng AI kiểm duyệt nội dung văn bản tiếng Việt được phát triển nhằm hỗ trợ các tổ chức trong việc tự động hóa quá trình kiểm duyệt, giúp nhận diện và loại bỏ những nội dung không phù hợp một cách hiệu quả. Hệ thống sẽ tích hợp nhiều chức năng tiện ích như phát hiện ngôn từ phản cảm, đánh giá tính phù hợp của nội dung, lọc bỏ các thông tin sai lệch hoặc có hại. Bên cạnh đó, AI cũng sẽ hỗ trợ phân loại và gắn nhãn các nội dung dựa trên mức độ nhạy cảm, giúp quản lý và điều hành các hoạt động truyền thông một cách chặt chẽ và hiệu quả hơn.

Tài liệu này sẽ cung cấp cái nhìn toàn diện về cách thức hệ thống AI kiểm duyệt nội dung văn bản tiếng Việt được thiết kế, triển khai và vận hành nhằm đáp ứng các yêu cầu và mục tiêu đề ra. Từ cấu trúc phần mềm, các dịch vụ API, đến thiết kế chức năng và quy trình thực hiện, tất cả đều được trình bày rõ ràng và cụ thể. Thông qua tài liệu này, đội ngũ phát triển sẽ có đầy đủ thông tin cần thiết để triển khai tính năng AI kiểm duyệt một cách hiệu quả, đồng thời các bên liên quan cũng có thể dễ dàng theo dõi và hiểu rõ tiến trình của dự án.

# 1.Cấu trúc phần mềm

## 1.1.Mô tả chung

| Tên Module | Mô tả |
| --- | --- |
| NLP Infer | NLP Infer là thành phần chịu trách nhiệm xử lý và phân loại nội dung văn bản. Cung cấp điểm cuối (endpoint) để các hệ thống bên ngoài và các ứng dụng khác có thể sử dụng tính năng phân loại văn bản. |
| NLP Training | NLP Training là thành phần thực hiện huấn luyện mô hình AI giúp cải tiến độ chính xác. Cung cấp điểm cuối (endpoint) để điều khiển (bắt đầu hoặc kết thúc) quá trình huấn luyện. |
| NLP Data | Tự động kiểm tra và khởi tạo các bucket minio, các bảng trong cơ sở dữ liệu khi khởi động hệ thống. Cung cấp api bổ sung dữ liệu đã gắn nhãn vào cơ sở dữ liệu phục vụ cho cải tiến mô hình. |
| MinIO | Quản lý model, dữ liệu huấn luyện. |
| Postgres DB | Lưu trữ dữ liệu đã gán nhãn từ người kiểm duyệt nội dung trên nền tảng MXH, được bổ sung liên tục để cải tiến mô hình. |
| Adminer | Công cụ quản trị dữ liệu trên Postgres DB. |
| Nginx | Thực hiện phân tải: phân phối yêu cầu của người dùng đến các NLP Infer. |
| Tensorboard | Giao diện theo dõi các chỉ số trong quá trình huấn luyện model. |

## 1.2.Cấu trúc NLP Infer

| Tên package | Chức năng |
| --- | --- |
| bert\_model | Xác định kiến trúc mô hình AI phân loại văn bản. |
| utils | Định nghĩa các hàm chức năng tiện ích hỗ trợ. |
| logger | Định nghĩa cấu hình ghi nhật ký. |
| constants | Đọc file cấu hình, khai báo các biến toàn cục. |
| minio\_client | Tương tác với minio server. |
| preprocess | Định nghĩa hàm tiền xử lý dữ liệu. |
| server | Tiếp nhận yêu cầu HTTP request, tương tác với mô hình AI, tiền xử lý dữ liệu, thực hiện phân loại văn bản và trả về response. |

## 1.3. Cấu trúc NLP Training

| Tên package | Chức năng |
| --- | --- |
| bert\_model | Xác định kiến trúc mô hình AI phân loại văn bản. |
| utils | Định nghĩa các hàm chức năng tiện ích hỗ trợ. |
| logger | Định nghĩa cấu hình ghi nhật ký. |
| constants | Đọc file cấu hình, khai báo các biến toàn cục. |
| minio\_client | Tương tác với minio server. |
| preprocess | Định nghĩa hàm tiền xử lý dữ liệu. |
| server | Tiếp nhận yêu cầu HTTP request, tương tác với mô hình AI, tiền xử lý dữ liệu, thực hiện huấn luyện mô hình, tải mô hình đã huấn luyện lên MinIO và trả về response. |

## 1.4. Cấu trúc NLP Data

| Tên package | Chức năng |
| --- | --- |
| constants | Đọc file cấu hình, khai báo các biến toàn cục. |
| utils | Định nghĩa các hàm chức năng tiện ích hỗ trợ |
| logger | Định nghĩa cấu hình ghi nhật ký. |
| minio\_client | Tương tác với minio server. |
| models | Định nghĩa mô hình dữ liệu bằng SQLAlchemy. |
| schemas | Định nghĩa schema cho dữ liệu đầu vào. |
| database | Thiết lập kết nối cơ sở dữ liệu. |
| server | Tiếp nhận yêu cầu HTTP request, lưu dữ liệu vào cơ sở dữ liệu và trả về response. |

# 2.Thiết kế chức năng

## 2.1.Danh sách các chức năng

| STT | Chức năng | Mô tả |
| --- | --- | --- |
| 1 | Phát hiện nội dung văn bản vi phạm tiêu chuẩn cộng đồng | Phát hiện và gắn nhãn nội dung không tuân thủ các quy tắc ứng xử do nền tảng thiết lập nhằm duy trì môi trường trực tuyến an toàn và lành mạnh. |
| 2 | Bổ sung dữ liệu huấn luyện | Bổ sung dữ liệu đã gắn nhãn vào cơ sở dữ liệu nhằm cải tiến mô hình AI. |
| 3 | Huấn luyện mô hình AI | Huấn luyện mô hình AI dựa trên dữ liệu đã được gán nhãn giúp cải tiến độ chính xác của mô hình. |
| 4 | Dừng huấn luyện mô hình AI | Chủ động dừng huấn luyện khi thấy các chỉ số không được cải thiện đáng kể. |

### 

## 2.2.Chức năng phát hiện nội dung văn bản vi phạm tiêu chuẩn cộng đồng

### 2.2.1.Thiết kế request API

TextClassifyReq

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| Data | ParagraphRequest | Dữ liệu gửi yêu cầu | Y |

ParagraphRequest

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| paragraph | String | Văn bản cần kiểm duyệt | Y |

### 2.2.2.Thiết kế response API

TextClassifyRes

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| Code | Int | Mã response | Y |
| Data | [ ] TextClassification | Dữ liệu phân loại văn bản | Y |

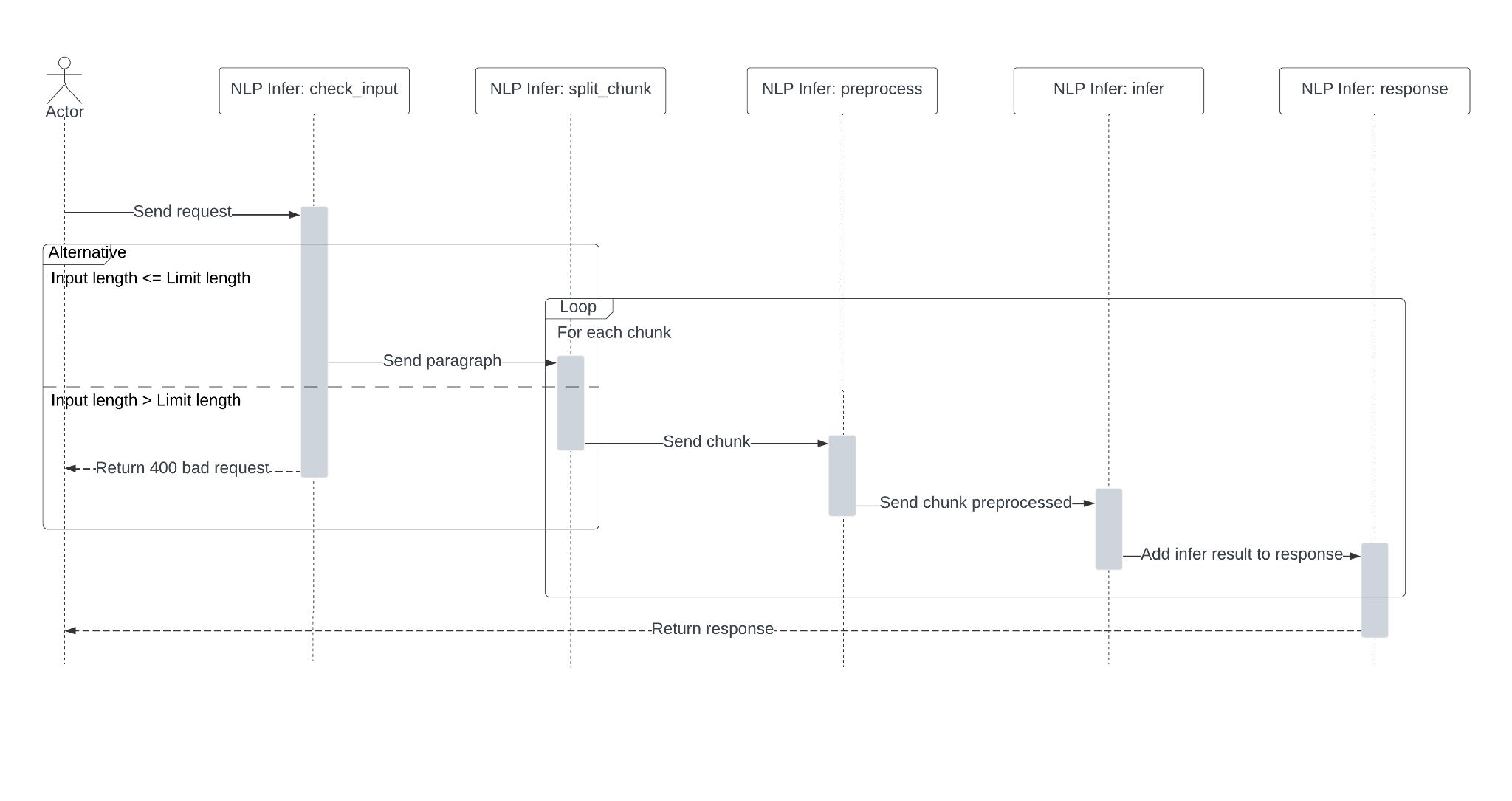
TextClassification

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| chunk | String | Một đoạn được tách ra từ văn bản đầu vào dựa theo giới hạn số từ có thể xử lý trong 1 lần chạy tiến trình phân loại | Y |
| label | String | Nhãn phân loại của đoạn văn bản | Y |
| confidence | Float | Độ tin cậy | Y |

### 2.2.3.Chu trình thực hiện

* Client thực hiện yêu cầu POST đến endpoint/text-classify với request TextClassifyReq.
* Yêu cầu được chuyến đến hàm kiểm tra văn bản đầu vào. Nếu đầu vào thỏa mãn yêu cầu về giới hạn số ký tự có thể xử lý thì chuyển tới bước tiếp theo. Nếu không thì trả về code 400 kèm thông báo về số ký tự giới hạn.
* Văn bản được chuyển tới hàm phân đoạn.
* Từng đoạn văn bản được chuyển tới hàm tiền xử lý, thực hiện phân đoạn từ và lọc bỏ ký tự không hợp lệ.
* Từng đoạn văn bản đã được tiền xử lý được chuyển tới hàm suy luận của model AI.
* Nếu thành công, trả về TextClassifyRes với code 200 cùng dữ liệu phân loại văn bản. Nếu thất bại, trả về code lỗi kèm mô tả lỗi của nó.

### 2.2.4.Biểu đồ tuần tự



## 2.3.Chức năng bổ sung dữ liệu huấn luyện

### 2.3.1.Thiết kế request API

AddDataReq

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| Data | LabeledData | Dữ liệu gửi yêu cầu | Y |

LabeledData

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| paragraph | String | Văn bản cần kiểm duyệt | Y |
| label | String | Nhãn phân loại của đoạn văn bản | Y |

### 2.3.2.Thiết kế response API

AddDataRes

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| Code | Int | Mã response | Y |
| Data | AddedData | Dữ liệu đã được bổ sung thành công | Y |

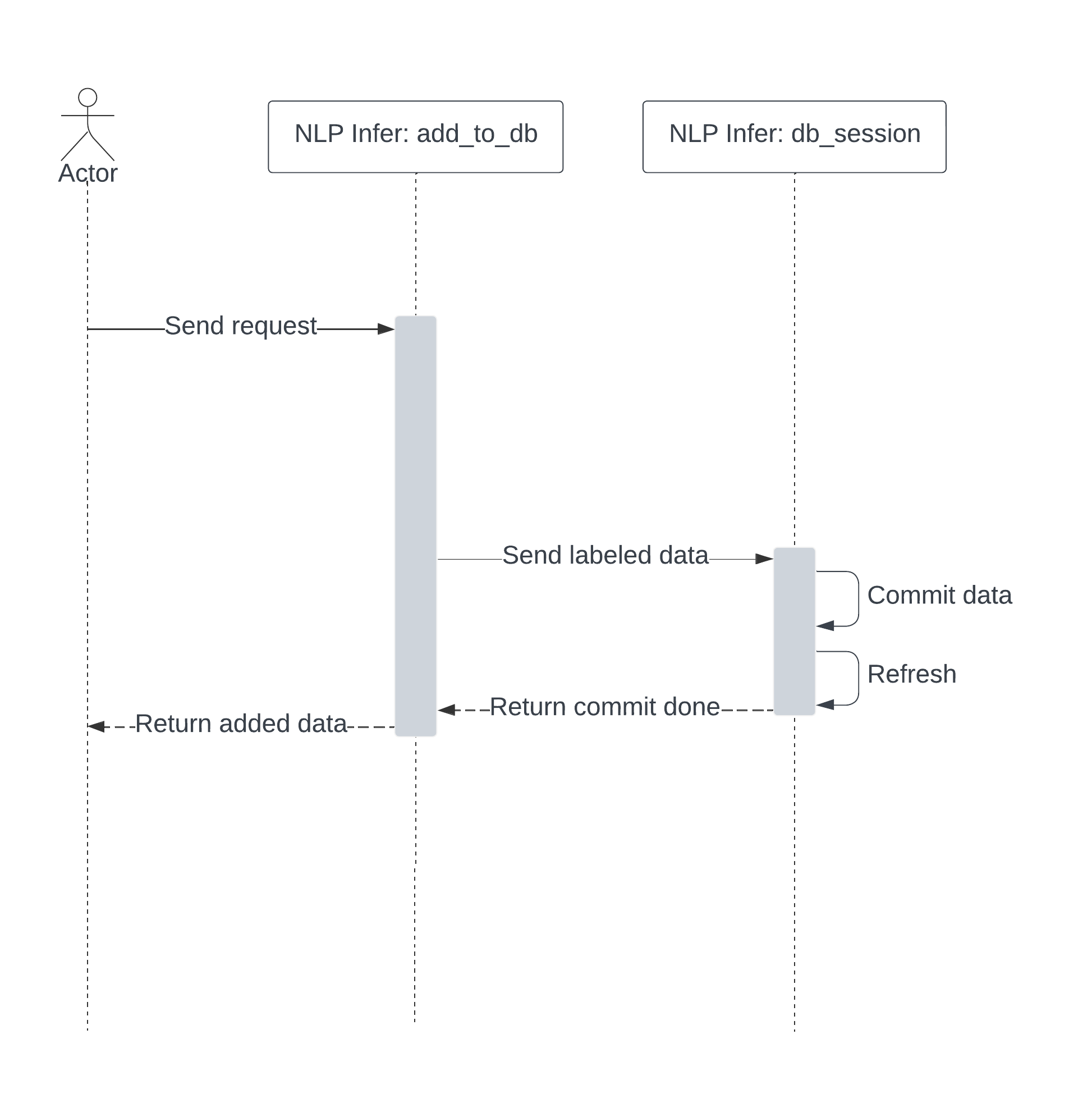
AddedData

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| paragraph | String | Văn bản cần kiểm duyệt | Y |
| label | String | Nhãn phân loại của đoạn văn bản | Y |

### 2.3.3.Chu trình thực hiện

* Client thực hiện yêu cầu POST đến endpoint/add-data với request AddDataReq.
* Dữ liệu được thêm vào cơ sở dữ liệu.
* Nếu thành công, trả về AddDataRes với code 200 cùng dữ liệu đã được bổ sung thành công. Nếu thất bại, trả về code lỗi kèm mô tả lỗi của nó.

### 2.3.4.Biểu đồ tuần tự



## 2.4. Chức năng huấn luyện mô hình AI

### 2.4.1.Thiết kế request API

StartTrainingReq

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| Data | TrainingRequest | Dữ liệu gửi yêu cầu | Y |

TrainingRequest

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| pretrain | String | Chỉ định model pretrain được sử dụng | Y |

### 2.4.2.Thiết kế response API

StartTrainingRes

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| Code | Int | Mã response | Y |
| Data | TrainingResponse | Kết quả khởi động training | Y |

TrainingResponse

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| status | String | Trạng thái tiến trình training | Y |
| message | String | Tin nhắn hệ thống gửi tới người phát triển | Y |

### 2.4.3.Chu trình thực hiện

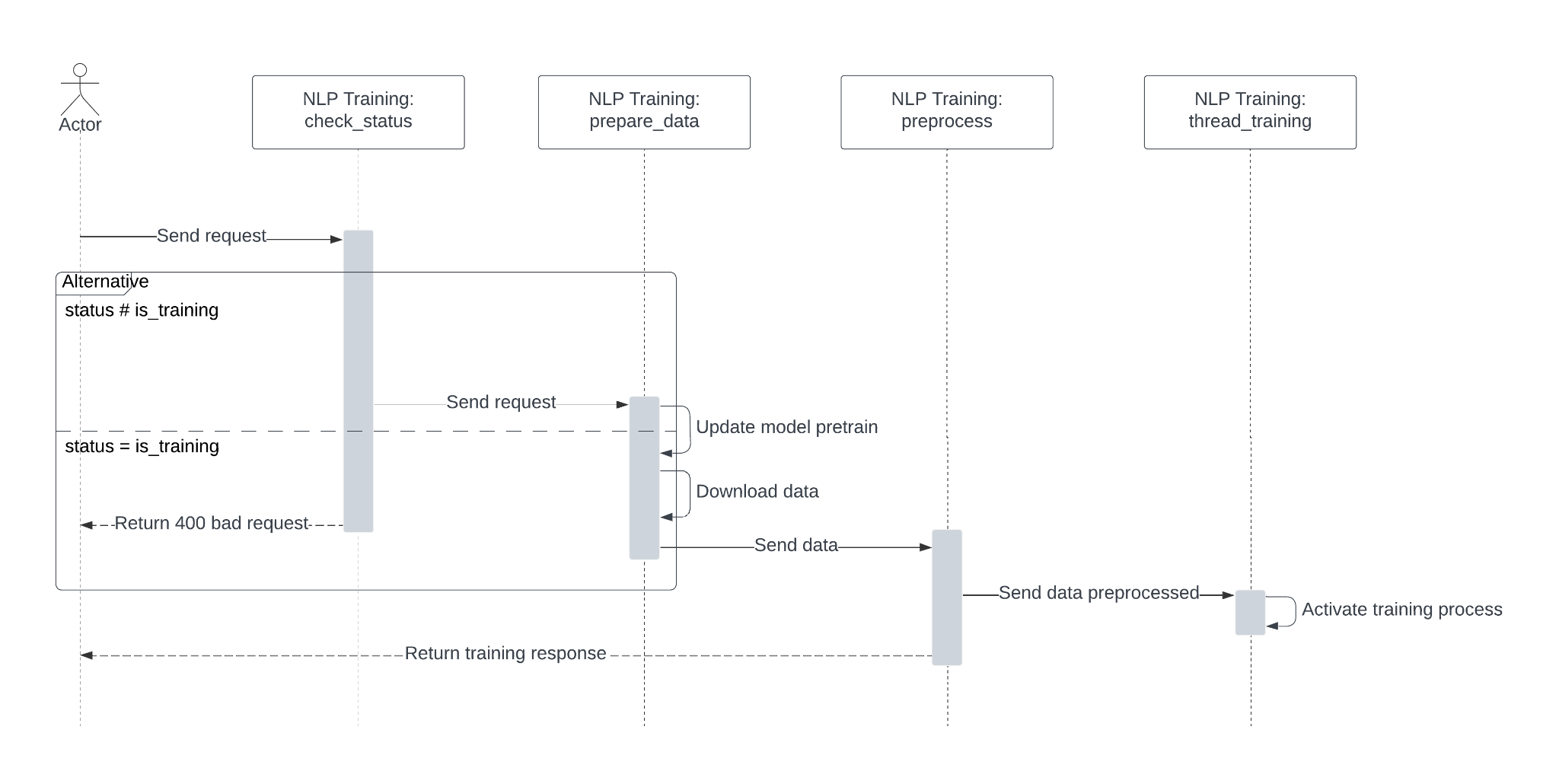
Chu trình thực hiện yêu cầu huấn luyện:

* Client thực hiện yêu cầu POST đến endpoint/start-training với request StartTrainingReq.
* Hệ thống kiểm tra xem có tiến trình training nào đang diễn ra không. Nếu có thì trả về code 400 kèm thông báo tiến trình training đang diễn ra. Nếu không thì chuyển sang bước tiếp theo.
* Cập nhật model pretrain và tải dữ liệu huấn luyện.
* Dữ liệu được chuyển sang bước tiền xử lý dữ liệu, thực hiện phân đoạn từ và loại bỏ các ký tự không hợp lệ.
* Dữ liệu đã qua tiền xử lý được gửi tới luồng huấn luyện kèm tín hiệu kích hoạt luồng huấn luyện.
* Trả về thông báo trạng thái bắt đầu training kèm tin nhắn tới người phát triển.

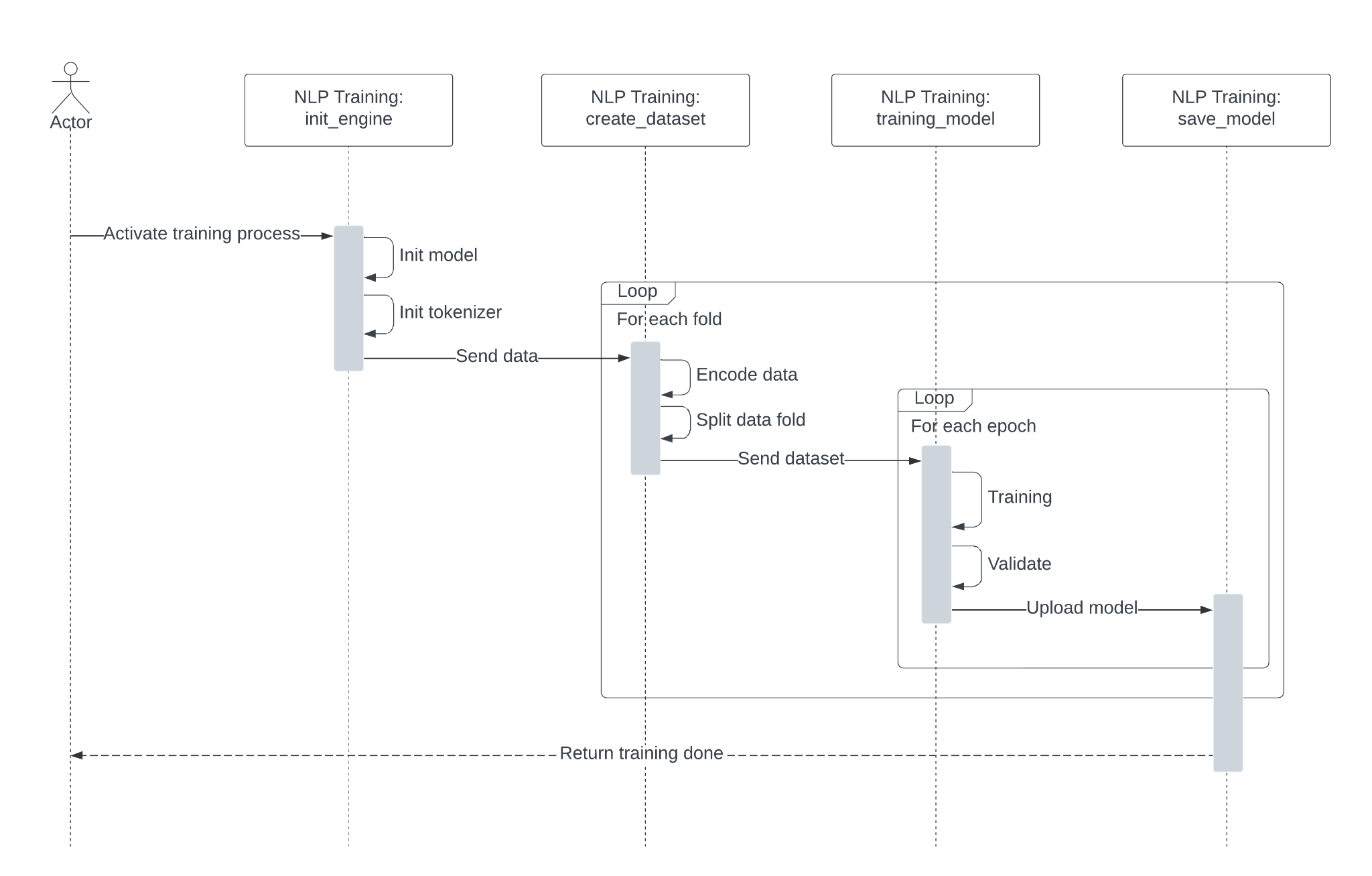
Chu trình thực hiện luồng huấn luyện:

* Nhận tín hiệu kích hoạt luồng huấn luyện
* Khởi tạo model và tokenizer
* Mã hóa dữ liệu
* Chia dataset
* Huấn luyện mô hình
* Kiểm tra kết quả huấn luyện
* Lưu mô hình

### 2.4.4.Biểu đồ tuần tự



*(Biểu đồ API training)*



*(Biểu đồ thread training)*

## 2.5. Chức năng dừng huấn luyện mô hình AI

### 2.5.1.Thiết kế request API

StopTrainingReq

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
|  |  |  |  |

### 2.5.2.Thiết kế response API

StopTrainingRes

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| Code | Int | Mã response | Y |
| Data | TrainingResponse | Kết quả khởi động training | Y |

TrainingResponse

| Tham số | Kiểu dữ liệu | Mô tả | Bắt buộc |
| --- | --- | --- | --- |
| status | String | Trạng thái tiến trình training | Y |
| message | String | Tin nhắn hệ thống gửi tới người phát triển | Y |

### 2.5.3.Chu trình thực hiện

Chu trình thực hiện yêu cầu huấn luyện:

* Client thực hiện yêu cầu POST đến endpoint/stop-training với request StopTrainingReq.
* Hệ thống kiểm tra xem có tiến trình training nào đang diễn ra không. Nếu không thì trả về code 400 kèm thông báo không có tiến trình training nào đang diễn ra. Nếu có thì chuyển sang bước tiếp theo.
* Gửi tín hiệu dừng huấn luyện tới luồng huấn luyện.
* Giải phóng VRam trên GPU.
* Trả về trạng thái đã dừng training kèm tin nhắn tới người phát triển.

### 2.5.4.Biểu đồ tuần tự

