BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

NGÀNH KHOA HỌC MÁY TÍNH

ĐỀ TÀI: XÂY DỰNG HỆ THỐNG TRÍCH XUẤT THÔNG TIN DANH THIẾP BẰNG PADDLEOCR KẾT HỢP VỚI NER VÀ TINH CHỈNH KẾT QUẢ BẰNG LLM

CBHD: TS. Nguyễn Mạnh Cường

Sinh viên: Đức Minh Hoàng

Mã sinh viên: 2021605732

Hà Nội, 2025

MỤC LỤC

LỜI CẢM ƠN

Em xin gửi lời tri ân chân thành và sâu sắc nhất đến thầy cô và các anh chị đã đồng hành và hỗ trợ em trong quá trình thực hiện đồ án tốt nghiệp "Xây dựng hệ thống trích xuất thông tin danh thiếp bằng PaddleOCR kết hợp với NER và tinh chỉnh kết quả bằng LLM".

Đây không chỉ là một đồ án nghiên cứu, mà còn là cơ hội để em vận dụng những kiến thức đã tích lũy, thử thách bản thân và từng bước tiến gần hơn đến con đường chuyên môn. Trong suốt quá trình đó, em may mắn nhận được sự hướng dẫn tận tình từ TS. Nguyễn Mạnh Cường. Thầy không chỉ giúp em định hướng rõ ràng mà còn truyền đạt những kiến thức quan trọng, khuyến khích em suy nghĩ sáng tạo và tiếp cận vấn đề một cách khoa học. Sự tận tâm của thầy là động lực lớn giúp em hoàn thành đồ án này.

Hành trình đại học khép lại với nhiều bài học ý nghĩa, giúp em trưởng thành cả về tri thức lẫn tư duy. Dù đã nỗ lực hết mình, nhưng chắc chắn báo cáo vẫn còn thiếu sót. Em mong nhận được góp ý từ thầy và các thầy cô để hoàn thiện hơn.

Xin trân trọng cảm ơn!

Đức Minh Hoàng

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG BIỂU

DANH MỤC KÝ TỰ, TỪ VIẾT TẮT

Từ viết tắt	Nghĩa
AI	Artificial Intelligence – Trí tuệ nhân tạo
ML	Machine Learning – Học máy
DL	Deep Learning – Học sâu
OCR	Optical Character Recognition – Nhận diện ký tự quang học
NER	Named Entity Recognition – Nhận diện thực thể có tên
LLM	Large Language Model – Mô hình ngôn ngữ lớn

LỜI NÓI ĐẦU

Trong thời đại công nghệ số, trí tuệ nhân tạo (AI) và học máy (Machine Learning) ngày càng đóng vai trò quan trọng trong nhiều lĩnh vực khác nhau, từ tự động hóa quy trình đến xử lý và phân tích dữ liệu. Đặc biệt, nhận dạng ký tự quang học (OCR - Optical Character Recognition) đã và đang trở thành một công nghệ thiết yếu trong việc số hóa tài liệu, hỗ trợ tìm kiếm và trích xuất thông tin một cách nhanh chóng và chính xác. Với sự phát triển mạnh mẽ của các mô hình AI, nhiều giải pháp OCR đã ra đời, giúp nâng cao hiệu suất và độ chính xác trong việc nhận dạng văn bản từ hình ảnh.

Đề tài "Xây dựng hệ thống trích xuất thông tin danh thiếp bằng PaddleOCR kết hợp với NER và tinh chỉnh kết quả bằng LLM" hướng đến phát triển một hệ thống tự động nhận diện và trích xuất thông tin từ danh thiếp. Tận dụng PaddleOCR giúp nhận dạng văn bản từ hình ảnh, trong khi huấn luyện mô hình NER phân loại các thực thể quan trọng như tên, số điện thoại, địa chỉ email, công ty. Cuối cùng, LLM được sử dụng để tinh chỉnh và cải thiện độ chính xác của thông tin trích xuất. Hệ thống này giúp doanh nghiệp tối ưu hóa quản lý danh bạ khách hàng, giảm sai sót nhập liệu và nâng cao hiệu quả xử lý thông tin.

Báo cáo được chia thành bốn chương như sau:

- **Chương 1: Khảo sát và phát biểu bài toán**. Chương này cung cấp cái nhìn tổng quan về bài toán, xác định các yêu cầu cần thiết và những thách thức gặp phải trong quá trình xây dựng hệ thống.
- **Chương 2: Một số phương pháp hiện có**. Chương này được dành để giới thiệu các phương pháp phổ biến trong nhận dạng ký tự quang học, các mô hình NER và LLM, đồng thời phân tích ưu nhược điểm của từng phương pháp.

- **Chương 3: Thực nghiệm và đánh giá**. Chương này mô tả quá trình triển khai hệ thống, bao gồm các bước tiền xử lý dữ liệu, huấn luyện và đánh giá mô hình, cũng như phân tích kết quả đạt được.
- **Chương 4: Xây dựng ứng dụng**. Chương này trình bày quá trình xây dựng ứng dụng thực tế, mô tả cách tích hợp các thành phần OCR, NER và LLM vào hệ thống, đồng thời đưa ra các kịch bản thử nghiệm thực tế.

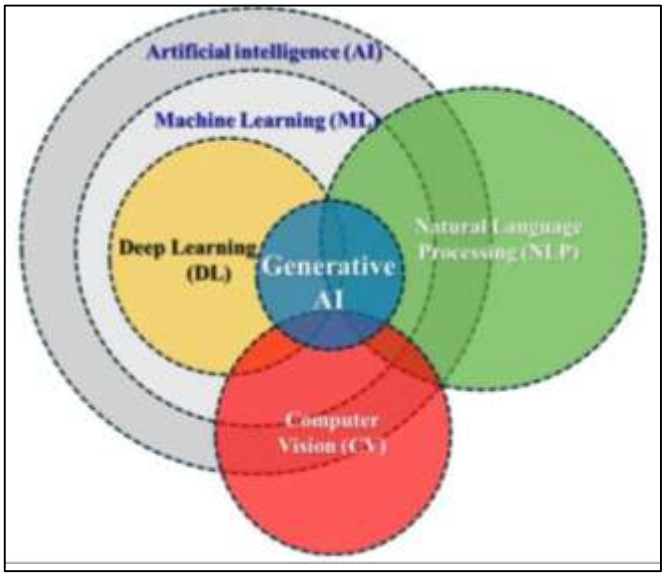
Qua đề tài này, em đã có cơ hội tìm hiểu sâu về các công nghệ OCR, NER và LLM, cũng như cách kết hợp chúng để xây dựng một hệ thống hoàn chỉnh. Em hy vọng rằng báo cáo này sẽ cung cấp những thông tin hữu ích cho những ai quan tâm đến lĩnh vực nhận dạng ký tự và trích xuất thông tin. Ngoài ra, hệ thống này có thể được ứng dụng trong thực tế để giúp doanh nghiệp và cá nhân quản lý thông tin hiệu quả hơn, góp phần thúc đẩy quá trình số hóa và tự động hóa trong công việc hàng ngày.

CHƯƠNG 1: KHẢO SÁT VÀ PHÁT BIỂU BÀI TOÁN

1.1. Tổng quan về trí tuệ nhân tạo

Trí tuệ nhân tạo (Artificial Intelligence - AI) là một lĩnh vực của khoa học máy tính tập trung vào việc phát triển các hệ thống và máy móc có khả năng thực hiện các nhiệm vụ đòi hỏi trí thông minh của con người. Các nhiệm vụ này bao gồm học tập, lập luận, giải quyết vấn đề, nhận thức và tương tác với môi trường. AI được định nghĩa rộng rãi như là khả năng của một hệ thống máy tính trong việc mô phỏng hoặc tái tạo các quá trình nhận thức của con người, từ việc xử lý thông tin đơn giản đến việc đưa ra các quyết định phức tạp trong các tình huống không chắc chắn [1].

Lịch sử của AI bắt đầu từ những năm 1950, khi Alan Turing đề xuất khái niệm về "máy có thể suy nghĩ" thông qua bài kiểm tra Turing nổi tiếng. Hội nghị Dartmouth năm 1956 được xem là cột mốc chính thức đánh dấu sự ra đời của AI như một lĩnh vực nghiên cứu độc lập. Trong những thập kỷ tiếp theo, AI trải qua nhiều giai đoạn thăng trầm, từ những kỳ vọng lớn vào các hệ thống dựa trên luật (rule-based systems) trong những năm 1980 đến sự bùng nổ của các phương pháp học máy (machine learning) vào đầu thế kỷ 21. Sự phát triển vượt bậc về năng lực tính toán, sự gia tăng của dữ liệu lớn (big data) và những tiến bộ trong các thuật toán đã thúc đẩy AI đạt được những thành tựu đáng kể trong thập kỷ vừa qua, đưa nó trở thành một trong những lĩnh vực có tác động lớn nhất đến xã hội hiện đại.



Hình 1.1: Sơ đồ Venn cho thấy mối quan hệ giữa các lĩnh vực AI [2]

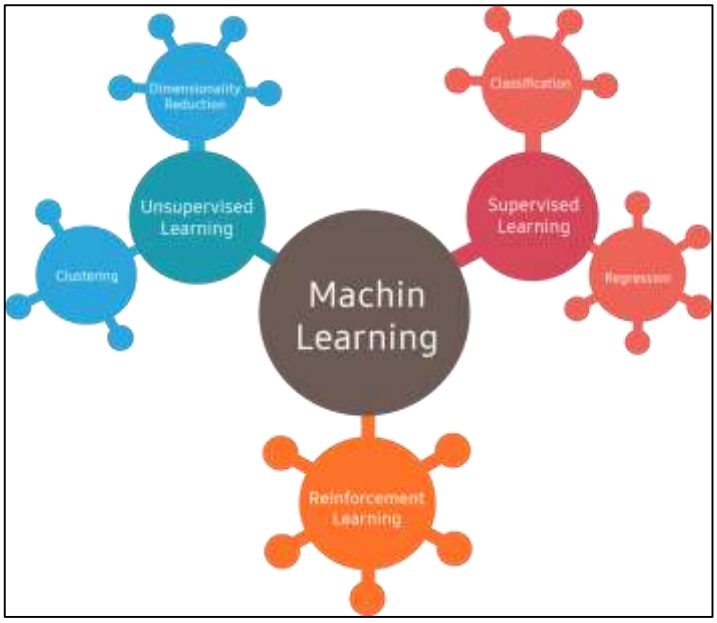
AI là một lĩnh vực đa dạng, bao gồm nhiều nhánh và phương pháp tiếp cận khác nhau, mỗi nhánh tập trung vào một khía cạnh cụ thể của trí thông minh như học máy (machine learning - ML), học sâu (deep learning – DL), xử lý ngôn ngữ tự nhiên (natural language processing - NLP) và thị giác máy tính (computer vision - CV).

1.1.1. Học máy

Học máy là một nhánh cốt lõi của AI, tập trung vào việc phát triển các thuật toán cho phép máy tính học hỏi từ dữ liệu mà không cần lập trình rõ ràng. Các hệ thống học máy sử dụng các mô hình thống kê để nhận diện các mẫu (patterns) trong dữ liệu và đưa ra dự đoán hoặc quyết định. Học máy được chia thành ba loại chính:

- Học có giám sát (Supervised Learning): Mô hình được huấn luyện trên dữ liệu đã được gán nhãn để dự đoán kết quả từ các đặc trưng đầu vào. Ví dụ, phân loại hoặc hồi quy.

- Học không giám sát (Unsupervised Learning): Mô hình xử lý dữ liệu không có nhãn để tìm ra các cấu trúc ẩn, như phân cụm hoặc giảm chiều dữ liệu.
- Học tăng cường (Reinforcement Learning): Mô hình học thông qua việc tương tác với môi trường, nhận phản hồi dưới dạng phần thưởng hoặc hình phạt để tối ưu hóa hành động.



Hình 1.2: Các nhánh của học máy [1]

Quy trình học máy gồm các bước chính sau: (1) Thu thập dữ liệu: tập hợp dữ liệu phù hợp với bài toán; (2) Tiền xử lý dữ liệu: làm sạch, chuẩn hóa và biến đổi dữ liệu để phù hợp với mô hình; (3) Chọn mô hình: lựa chọn thuật toán học máy phù hợp (ví dụ: hồi quy, cây quyết định, mạng nơ-ron); (4) Huấn luyện mô hình: sử dụng dữ liệu huấn luyện để học các tham số của mô hình; (5) Đánh giá mô hình: kiểm tra hiệu suất trên dữ liệu kiểm thử bằng các chỉ số như độ chính xác, F1-score; (6) Triển khai và giám sát: áp dụng mô hình vào thực tế và theo dõi để cải thiện khi cần thiết.

Học máy đã được ứng dụng rộng rãi trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận diện hình ảnh, và hệ thống khuyến nghị. Trong những năm

gần đây, học máy đã phát triển mạnh mẽ với nhiều xu hướng nổi bật như học sâu (deep learning) và học tăng cường (reinforcement learning) [2]. Các mô hình ngày càng phức tạp như mạng Transformer và mô hình ngôn ngữ lớn (LLMs) đang được ứng dụng rộng rãi trong nhiều lĩnh vực, từ y tế đến công nghiệp. Đồng thời, các vấn đề về bảo mật dữ liệu, khả năng giải thích của mô hình (XAI), và học máy bền vững cũng trở thành trọng tâm nghiên cứu, nhằm đảm bảo tính minh bạch, hiệu quả và thân thiện với môi trường trong việc triển khai các hệ thống học máy hiện đại.

1.1.2. Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) là một lĩnh vực nghiên cứu liên ngành giữa khoa học máy tính, trí tuệ nhân tạo và ngôn ngữ học, tập trung vào việc xây dựng các hệ thống có khả năng hiểu và tạo sinh ngôn ngữ tự nhiên của con người [1]. NLP đóng vai trò như một cầu nối giữa ngôn ngữ của con người và ngôn ngữ của máy tính, cho phép máy có thể xử lý, phân tích và phản hồi lại thông tin bằng văn bản hoặc lời nói một cách hiệu quả. Đây là nền tảng cho nhiều ứng dụng thiết thực như trợ lý ảo, dịch máy, hệ thống hỏi – đáp và phân tích cảm xúc.

Trong NLP, có nhiều bài toán chủ đạo được nghiên cứu và phát triển. Các bước đầu tiên trong quy trình xử lý bao gồm:

- Tiền xử lý văn bản: là bước nền tảng bao gồm các kỹ thuật như tách từ (tokenization), chuẩn hóa (normalization), loại bỏ từ dừng (stopword removal), và gán nhãn từ loại (part-of-speech tagging).
- Phân loại văn bản: xác định loại nội dung của văn bản, chẳng hạn như phân loại email rác, phân tích cảm xúc, hay xác định chủ đề.
- Rút trích thông tin (Information Extraction): nhận diện thực thể có tên (Named Entity Recognition – NER), trích xuất quan hệ và sự kiện trong văn bản.

- Gán nhãn thực thể và phân giải đồng tham chiếu: giúp xác định các thực thể (người, địa điểm, tổ chức,...) và mối liên hệ giữa các cụm từ đại diện cho cùng một thực thể.
- Dịch máy (Machine Translation): tự động dịch nội dung từ ngôn ngữ này sang ngôn ngữ khác.
- Tạo sinh ngôn ngữ tự nhiên (Natural Language Generation – NLG): tạo ra văn bản mới từ dữ liệu có cấu trúc, điển hình như tóm tắt văn bản hoặc sinh câu trả lời.
- Hệ thống hỏi – đáp (Question Answering – QA) và đối thoại thông minh: xây dựng các hệ thống có khả năng trả lời câu hỏi hoặc tương tác ngôn ngữ với người dùng trong thời gian thực.

Trong những năm gần đây, NLP chứng kiến sự chuyển mình mạnh mẽ với sự áp dụng rộng rãi của các mô hình học sâu (deep learning). Đặc biệt, các mô hình ngôn ngữ lớn (Large Language Model – LLM) như BERT, GPT, RoBERTa, và T5 đã tạo ra bước đột phá đáng kể nhờ khả năng học biểu diễn ngữ nghĩa phức tạp và sinh ngôn ngữ tự nhiên với độ chính xác cao [2]. Các mô hình này được huấn luyện trên tập dữ liệu văn bản lớn, cho phép chúng hiểu được ngữ cảnh dài, mối quan hệ ngôn ngữ, và thích nghi với nhiều tác vụ NLP khác nhau chỉ thông qua tinh chỉnh (fine-tuning) hoặc thiết kế gợi nhắc (prompzt engineering).

Bên cạnh đó, các xu hướng hiện nay trong NLP cũng đang tập trung vào việc cải thiện hiệu quả mô hình, mở rộng khả năng đa ngôn ngữ (multilingual NLP), học không giám sát và học chuyển giao (zero-shot, few-shot learning). Đồng thời, các vấn đề về tính công bằng, minh bạch, và khả năng giải thích của mô hình (explainable AI) cũng được chú trọng nhằm giảm thiểu rủi ro thiên lệch (bias) và tăng tính đáng tin cậy trong các ứng dụng thực tế. Ngoài ra, học máy bền vững (sustainable ML) cũng đang trở thành một chủ đề nóng, hướng

đến việc xây dựng các mô hình NLP có chi phí tính toán thấp hơn nhưng vẫn đảm bảo hiệu suất cao.

Tóm lại, xử lý ngôn ngữ tự nhiên là một trong những lĩnh vực trọng tâm của trí tuệ nhân tạo hiện đại, với tiềm năng ứng dụng sâu rộng trong khoa học, công nghệ và đời sống. Sự kết hợp giữa tiến bộ trong học sâu và hiểu biết ngôn ngữ học đang giúp các hệ thống NLP ngày càng thông minh, linh hoạt và gần gũi hơn với cách con người sử dụng ngôn ngữ trong thực tế.

1.1.3. Thị giác máy tính

Thị giác máy tính (Computer Vision – CV) là một lĩnh vực quan trọng của trí tuệ nhân tạo, nhằm phát triển các hệ thống có khả năng "nhìn" và hiểu được hình ảnh hoặc video tương tự như con người [1]. Mục tiêu chính của thị giác máy tính là trích xuất thông tin có ý nghĩa từ dữ liệu thị giác để hỗ trợ việc ra quyết định tự động. Đây là công nghệ nền tảng cho nhiều ứng dụng thực tiễn như xe tự hành, nhận diện khuôn mặt, giám sát thông minh, phân tích y tế, và robot công nghiệp.

Các bài toán chủ đạo trong thị giác máy tính rất đa dạng, bao gồm:

- Phân loại hình ảnh (Image Classification): xác định đối tượng chính trong ảnh thuộc loại nào, ví dụ: chó, mèo, xe hơi, v.v.
- Phát hiện đối tượng (Object Detection): không chỉ phân loại mà còn xác định vị trí các đối tượng trong ảnh thông qua các hộp giới hạn (bounding boxes).
- Phân đoạn ảnh (Image Segmentation): chia ảnh thành các vùng tương ứng với từng đối tượng hoặc bối cảnh, có hai dạng phổ biến là phân đoạn ngữ nghĩa (semantic segmentation) và phân đoạn thể hiện (instance segmentation).

- Nhận diện khuôn mặt (Face Recognition) và phân tích cảm xúc: xác định danh tính hoặc trạng thái cảm xúc của con người thông qua biểu cảm khuôn mặt.
- Theo dõi đối tượng (Object Tracking): theo dõi chuyển động của các đối tượng trong chuỗi video theo thời gian thực.
- Tái tạo 3D và nhận diện tư thế (3D Reconstruction, Pose Estimation): tái hiện lại hình dạng và vị trí không gian của vật thể hoặc con người từ hình ảnh 2D.
- Hiểu cảnh và hoạt động (Scene Understanding & Action Recognition): phân tích toàn bộ bối cảnh trong ảnh hoặc video và nhận diện hành vi, sự kiện đang diễn ra.

Thị giác máy tính đang phát triển mạnh mẽ nhờ sự tiến bộ của học sâu, đặc biệt là các mạng nơ-ron tích chập (CNNs) và các kiến trúc hiện đại như Vision Transformer. Các mô hình như ResNet, YOLO, Mask R-CNN hay ViT đã giúp nâng cao độ chính xác trong các bài toán như phân loại hình ảnh, phát hiện và phân đoạn đối tượng. Xu hướng hiện nay tập trung vào học tự giám sát, học không giám sát và thị giác đa phương thức, cho phép mô hình học từ dữ liệu không gán nhãn hoặc kết hợp giữa hình ảnh và văn bản – như mô hình CLIP [2]. Song song đó, việc tối ưu mô hình để triển khai trên thiết bị biên (edge computing) cũng được quan tâm, nhằm phục vụ các ứng dụng thời gian thực trong môi trường tài nguyên hạn chế. Trong đó, một hướng đi nổi bật là nhận dạng ký tự quang học (OCR), với mục tiêu chuyển đổi hình ảnh chứa văn bản thành dữ liệu có thể xử lý được. Các hệ thống OCR hiện đại kết hợp CNN với các mô hình tuần tự như LSTM hoặc transformer để tăng khả năng nhận dạng văn bản in, viết tay, và các tài liệu có bố cục phức tạp như hóa đơn hoặc biểu mẫu. Ngoài ra, thị giác máy tính hiện đại còn chú trọng đến các yếu tố đạo đức như tính công bằng, khả năng giải thích và tránh thiên lệch – đặc biệt trong

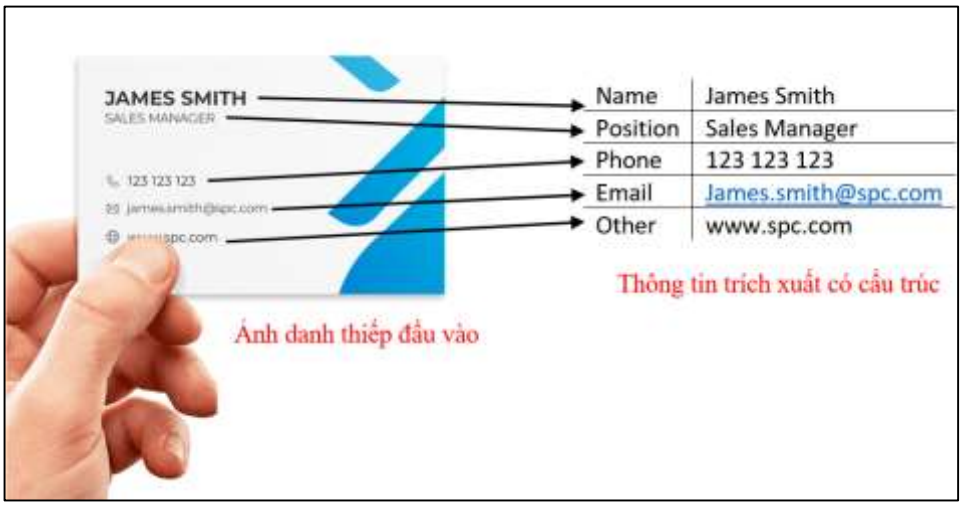
các ứng dụng nhạy cảm như giám sát hay nhận diện khuôn mặt. Với tốc độ phát triển như hiện nay, thị giác máy tính sẽ tiếp tục là một trong những công nghệ chủ lực trong kỷ nguyên trí tuệ nhân tạo.

Tóm lại, thị giác máy tính đang là một trong những lĩnh vực dẫn đầu trong làn sóng đổi mới công nghệ, góp phần xây dựng các hệ thống thông minh có khả năng cảm nhận và tương tác hiệu quả với thế giới thị giác. Với đà phát triển hiện nay, thị giác máy tính sẽ tiếp tục đóng vai trò quan trọng trong các ứng dụng công nghiệp, y tế, giao thông và cuộc sống hàng ngày trong kỷ nguyên trí tuệ nhân tạo.

1.2. Bài toán trích xuất thông tin danh thiếp

1.2.4. Mô tả bài toán

Bài toán trích xuất thông tin danh thiếp nhằm mục đích tự động hóa quá trình phân tích và trích xuất các thông tin quan trọng từ hình ảnh danh thiếp (business card) được cung cấp dưới dạng dữ liệu ảnh số. Các thông tin cần trích xuất thường bao gồm họ tên, chức danh, số điện thoại, địa chỉ email, tên công ty, địa chỉ công ty và các thông tin liên quan khác nếu có. Bài toán này thuộc lĩnh vực xử lý ảnh và nhận diện ký tự quang học (Optical Character Recognition - OCR), kết hợp với các kỹ thuật xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) để phân loại và định dạng thông tin một cách chính xác.



Hình 1.3: Minh họa đầu vào - đầu ra của bài toán trích xuất thông tin danh thiếp

Đầu vào của bài toán là một hình ảnh danh thiếp, được biểu diễn dưới dạng ma trận số học hoặc tệp ảnh định dạng phổ biến (ví dụ: JPEG, PNG). Cụ thể:

Hình ảnh danh thiếp: Một tệp ảnh I có kích thước $H \times W \times C$ trong đó:

- H là chiều cao (số pixel)
- W là chiều rộng (số pixel)
- C là số kênh màu (thường là 3 cho ảnh RGB hoặc 1 cho ảnh grayscale).

Giả định: Hình ảnh danh thiếp được chụp rõ nét, chứa các thông tin cần thiết dưới dạng văn bản có thể đọc được bằng mắt thường. Tuy nhiên, hình ảnh có thể bị ảnh hưởng bởi nhiễu, góc chụp không thẳng, hoặc ánh sáng không đồng đều.

Đầu ra của bài toán là một tập hợp thông tin được trích xuất từ danh thiếp, được biểu diễn dưới dạng cấu trúc dữ liệu có tổ chức. Cụ thể:

Danh sách thông tin: Một tập hợp $S = \{s_1, s_2, \dots, s_n\}$, trong đó mỗi s_i là một cặp (key, value) đại diện cho một trường thông tin. Ví dụ:

- $s_1 =$ ("Họ tên", "$Nguyễn\ Văn\ A$")

- $s_2 =$ ("Số điện thoại", "012345678")
- $s_3 =$ ("Email","nguyenvana@example.com")
- $s_4 =$ (Công ty", $Công\ ty\ XYZ$")

Định dạng: Có thể là một từ điển (dictionary) trong lập trình hoặc một bảng (Table) dữ liệu có cấu trúc thuận tiện cho việc lưu trữ và truy xuất thông tin.

1.2.5. Ràng buộc của bài toán

Để đảm bảo tính khả thi và hiệu quả của hệ thống, bài toán cần tuân thủ một số ràng buộc liên quan đến dữ liệu đầu vào, hiệu suất, và chất lượng đầu ra. Bài toán chịu một số ràng buộc sau:

- Chất lượng ảnh: Hình ảnh đầu vào phải có độ phân giải tối thiểu để văn bản có thể nhận diện được (ví dụ: ít nhất 300 DPI nếu là ảnh quét).
- Ngôn ngữ: Văn bản trên danh thiếp chủ yếu sử dụng một ngôn ngữ (ví dụ: tiếng Việt hoặc tiếng Anh). Trường hợp hỗn hợp ngôn ngữ có thể làm giảm độ chính xác.
- Bố cục: Danh thiếp có bố cục không cố định, nhưng các trường thông tin cần được phân biệt rõ ràng dựa trên ngữ cảnh hoặc định dạng (ví dụ: số điện thoại thường có định dạng chuẩn như "+84xxx" hoặc "xxx-xxx-xxxx").
- Thời gian xử lý: Hệ thống cần hoàn thành quá trình trích xuất trong thời gian hợp lý (ví dụ: dưới 5 giây cho một hình ảnh trên phần cứng tiêu chuẩn).
- Xử lý trường hợp đa thông tin trong cùng một danh thiếp: Đây là trường hợp mỗi key có thể nhiều hơn một value. Ví dụ key là số điện thoại có thể có 2 cái xuất hiện trong một danh thiếp.

- Độ chính xác: Độ chính xác của thông tin trích xuất phải đạt ít nhất 90% trên tập dữ liệu thử nghiệm tiêu chuẩn.

Các ràng buộc này định hình phạm vi và yêu cầu kỹ thuật của bài toán, đồng thời là cơ sở để đánh giá hiệu quả của các giải pháp được đề xuất.

1.2.6. Thuận lợi

Bài toán được hỗ trợ bởi nhiều yếu tố thuận lợi, tạo điều kiện cho việc phát triển và triển khai hệ thống trích xuất thông tin danh thiếp. Việc giải quyết bài toán "trích xuất thông tin danh thiếp" được hỗ trợ bởi một số thuận lợi sau:

- Công nghệ hiện đại sẵn có: Sự phát triển của các kỹ thuật xử lý ảnh, nhận diện ký tự quang học (OCR) và học sâu cung cấp các công cụ mạnh mẽ như Tesseract, OpenCV, hoặc các mô hình học máy tiên tiến (ví dụ: Transformer, CNN).
- Ứng dụng công nghệ di động: Sự phổ biến của điện thoại thông minh với camera chất lượng cao giúp dễ dàng chụp và xử lý hình ảnh danh thiếp trong thời gian thực.
- Cộng đồng nghiên cứu và phát triển: Bài toán được hỗ trợ từ cộng đồng nghiên cứu rộng lớn, với nhiều thư viện mã nguồn mở và tài liệu khoa học sẵn có.

Các thuận lợi này không chỉ giảm bớt khó khăn trong quá trình phát triển mà còn mở ra cơ hội để xây dựng một hệ thống hiệu quả và dễ tiếp cận.

1.2.7. Thách thức

Bài toán "trích xuất thông tin danh thiếp" đối mặt với một số khó khăn và thách thức đáng kể, bao gồm:

- Chất lượng hình ảnh không đồng đều: Hình ảnh danh thiếp có thể bị nhiễu (noise), mờ (blur), hoặc bị biến dạng do góc chụp không chuẩn (perspective distortion), dẫn đến việc nhận diện văn bản trở nên khó

khăn. Ví dụ, ánh sáng không đồng đều có thể làm mất thông tin ở một số vùng ảnh.

- Bố cục đa dạng: Danh thiếp không có bố cục chuẩn hóa, với vị trí và cách sắp xếp thông tin thay đổi tùy theo thiết kế. Điều này đòi hỏi hệ thống phải có khả năng thích nghi với các mẫu bố cục khác nhau mà không dựa vào giả định cố định.
- Nhầm lẫn giữa các trường thông tin: Một số trường thông tin có định dạng tương tự nhau (ví dụ: số điện thoại và số fax) hoặc văn bản không rõ ràng (ví dụ: "Nguyễn Văn A - Giám đốc" có thể bị nhầm lẫn giữa họ tên và chức danh), gây khó khăn trong việc phân loại chính xác.
- Hỗn hợp ngôn ngữ và ký tự đặc biệt: Danh thiếp có thể chứa nhiều ngôn ngữ (ví dụ: tiếng Việt có dấu và tiếng Anh) hoặc ký tự đặc biệt (ví dụ: "@", "#"), làm phức tạp hóa quá trình nhận diện và phân tích ngữ nghĩa.
- Hiệu suất và tài nguyên: Việc xử lý ảnh và phân tích văn bản đòi hỏi tài nguyên tính toán lớn, đặc biệt khi triển khai trên các thiết bị di động hoặc hệ thống thời gian thực, dẫn đến thách thức trong việc cân bằng giữa độ chính xác và tốc độ.
- Thiếu dữ liệu huấn luyện đa dạng: Để đạt được độ chính xác cao, mô hình học máy cần được huấn luyện trên tập dữ liệu lớn và phong phú, nhưng việc thu thập và gắn nhãn thủ công các danh thiếp đa dạng là một quá trình tốn kém và mất thời gian.

1.2.8. Ứng dụng

Giải quyết thành công bài toán trích xuất thông tin danh thiếp mang lại nhiều giá trị thực tiễn, với tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực từ kinh doanh đến quản lý. Việc giải quyết bài toán mang lại các lợi ích sau:

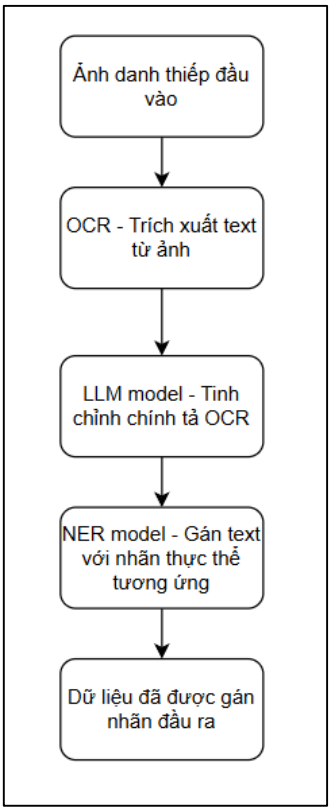
- Tự động hóa quản lý liên hệ: Hệ thống cho phép số hóa và lưu trữ thông tin liên hệ vào cơ sở dữ liệu hoặc ứng dụng CRM mà không cần nhập liệu thủ công, tiết kiệm thời gian và giảm sai sót.
- Tăng cường hiệu quả kinh doanh: Doanh nhân có thể nhanh chóng thu thập và tổ chức thông tin đối tác trong các sự kiện kết nối, thúc đẩy cơ hội hợp tác.
- Ứng dụng trong phân tích dữ liệu: Thông tin trích xuất có thể được sử dụng để phân tích xu hướng kinh doanh hoặc xây dựng cơ sở dữ liệu khách hàng.
- Hỗ trợ đa ngành: Bài toán có thể được mở rộng để trích xuất thông tin từ các tài liệu khác, như thẻ hội viên hoặc giấy tờ tùy thân.
- Tiết kiệm tài nguyên: Tự động hóa quy trình trích xuất giúp doanh nghiệp tối ưu hóa chi phí vận hành và nâng cao hiệu suất.

Những lợi ích này khẳng định tính thực tiễn của bài toán và thúc đẩy việc nghiên cứu các giải pháp tối ưu. Mục tiêu là xây dựng hệ thống tự động với độ chính xác cao, xử lý nhanh, dễ mở rộng cho nhiều loại danh thiếp và ngôn ngữ, bằng cách ứng dụng kỹ thuật xử lý ảnh, học máy và NLP hiện đại.

CHƯƠNG 2: MỘT SỐ PHƯƠNG PHÁP HIỆN CÓ

2.1. Phương pháp tiếp cận bài toán

Trích xuất thông tin từ danh thiếp là một bài toán đa mô hình, kết hợp giữa xử lý ảnh, nhận dạng văn bản và hiểu ngữ nghĩa tự nhiên của ngôn ngữ. Mặc dù có vẻ đơn giản về mặt hình thức vì chỉ là việc đọc thông tin từ một tấm danh thiếp nhưng trên thực tế, đây là một bài toán mang nhiều thách thức. Bố cục đa dạng, văn bản song ngữ, phông chữ lạ, chất lượng ảnh không ổn định, và đặc biệt là tính ngữ cảnh phức tạp. Từ những yêu cầu này, báo cáo được xác định hướng tiếp cận bài toán theo mô hình pipeline gồm ba giai đoạn: nhận diện và trích xuất văn bản, gắn nhãn thông tin và tinh chỉnh kết quả bằng mô hình ngữ nghĩa. Mỗi giai đoạn đều ứng dụng các lý thuyết và kỹ thuật hiện đại trong trí tuệ nhân tạo, nhằm đạt được sự chính xác, linh hoạt và khả năng mở rộng.



Hình 2.1: Luồng hoạt động của phương pháp

Quy trình trích xuất thông tin danh thiếp gồm 4 giai đoạn chính:

- Trích xuất văn bản từ ảnh bằng OCR
- Chuẩn hóa văn bản và hiệu chỉnh lỗi bằng mô hình ngôn ngữ lớn (LLM)
- Nhận diện và gán nhãn thực thể bằng mô hình NER

Trích xuất văn bản từ ảnh bằng OCR

Trong bước đầu tiên, hệ thống nhận cần phát hiện cần chuyển đổi dữ liệu từ dạng hình ảnh sang dạng văn bản. Đây là bước tiền xử lý thiết yếu nhằm tạo ra đầu vào cho các bước phân tích ngôn ngữ tiếp theo.

- Đầu vào: Hình ảnh danh thiếp ở định dạng chuẩn (JPG, PNG, TIFF, v.v.).
- Phương pháp: Sử dụng các thuật toán nhận dạng ký tự quang học (OCR) như PaddleOCR hoặc các mô hình tiên tiến hơn dựa trên Transformer để trích xuất văn bản từ hình ảnh.
- Yêu cầu:
    - Nhận diện chính xác văn bản trong điều kiện chụp ảnh thực tế (góc nghiêng, ánh sáng yếu).
    - Giữ nguyên vị trí và thứ tự dòng văn bản để hỗ trợ cho các bước xử lý ngôn ngữ sau.
- Đầu ra: Văn bản thô, có thể chứa lỗi do OCR gây ra, chưa được chuẩn hóa.

Ví dụ đầu ra:

Nguyen Van A

CEO - ABC Corp

123 Nguyen Trai, HN

0909 123 456 - a.nguyen@abc.com

Hiệu chỉnh lỗi bằng mô hình ngôn ngữ lớn (LLM)

Sau khi thu thập văn bản thô, bước tiếp theo là cải thiện chất lượng văn bản bằng cách hiệu chỉnh lỗi chính tả và chuẩn hóa cấu trúc câu nhằm đảm bảo tính chính xác và dễ xử lý hơn cho các mô hình nhận diện thực thể.

- Đầu vào: Văn bản thô từ kết quả OCR.
- Phương pháp: Áp dụng mô hình ngôn ngữ lớn (Large Language Model - LLM) để sửa lỗi chính tả, phục hồi dấu tiếng Việt nếu bị thiếu, và chuẩn hóa các định dạng đặc thù như số điện thoại, email.
- Yêu cầu:
  - Giữ nguyên ngữ cảnh và ý nghĩa ban đầu của văn bản.
  - Tăng cường độ chính xác và sự nhất quán cho dữ liệu văn bản đầu ra.
- Đầu ra: Văn bản đã được chỉnh sửa và chuẩn hóa, sẵn sàng cho quá trình nhận diện thực thể.

Ví dụ đầu ra:

Nguyễn Văn A

CEO - ABC Corp

123 Nguyễn Trãi, Hà Nội

0909 123 456 - a.nguyen@abc.com

Nhận diện và gán nhãn thực thể bằng mô hình NER

Sau khi có chuỗi văn bản đã được chính xác hóa, bước tiếp theo là xác định từng phần thông tin có vai trò gì. Đây chính là nhiệm vụ của mô hình gán nhãn thực thể có tên (Named Entity Recognition – NER). Trong ngữ cảnh danh thiếp, các thực thể cần được gán nhãn bao gồm: tên người, chức danh, tên công ty, email, số điện thoại, địa chỉ, v.v. Tuy nhiên, khác với văn bản thông thường, văn bản trong danh thiếp ngắn gọn, thiếu cấu trúc ngữ pháp rõ ràng, và thường không tuân theo ngữ cảnh đầy đủ, gây khó khăn cho các mô hình NER cổ điển.

- Đầu vào: Văn bản đã được chỉnh sửa từ bước trước.

Phương pháp: Sử dụng mô hình nhận diện thực thể có tên (Named Entity Recognition - NER) để phát hiện và gán nhãn các thực thể như tên người, chức vụ, công ty, số điện thoại, email, địa chỉ,... Để giải quyết vấn đề này, sử dụng các mô hình NER dựa trên kiến trúc Transformer, tiêu biểu là BERT và các biến thể như:

  - Bert-base-multilingual-cased: hỗ trợ đa ngôn ngữ, phù hợp với dữ liệu danh thiếp tiếng Việt – Anh.
  - BiLSTM-CRF: chuyên dụng cho các bài toán NER, với khả năng hiểu ngữ cảnh và ít tốn bộ nhớ.
  - XLM-RoBERTa: mô hình mạnh cho bài toán NER đa ngữ, với khả năng học biểu diễn ngữ nghĩa sâu.

- Yêu cầu:
  - Nhận diện chính xác các thực thể dù văn bản không theo một định dạng cố định.
  - Tối ưu cho các trường thông tin phổ biến trên danh thiếp.
- Đầu ra: Dữ liệu các thực thể đã được gán nhãn được viết dưới dạng có cấu trúc như json.

Ví dụ đầu ra:

```
{
      “Name”: “Nguyễn Văn A”
      “Position”: “CEO”
      “Company”: “AB Crop”
      “Address”: “123 Nguyễn Trãi, Hà Nội”
      “Phone”: “0909 123 456”
      “Email”: "a.nguyen@abc.com"
}
```

2.2. Nhận diện ký tự quang học

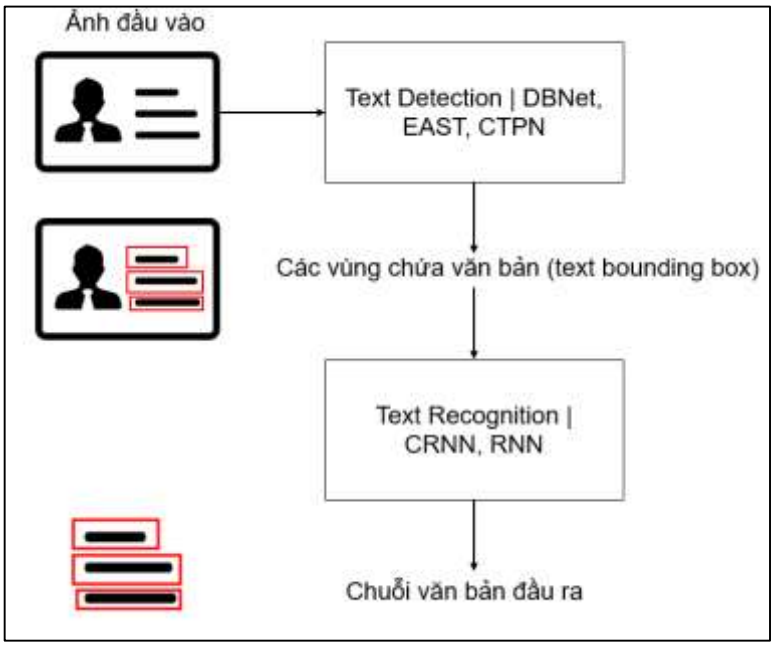
Nhận diện ký tự quang học (Optical Character Recognition – OCR) là kỹ thuật cho phép máy tính chuyển đổi hình ảnh chứa văn bản (ảnh chụp, scan tài liệu, biển hiệu, v.v.) thành các ký tự văn bản có thể chỉnh sửa và tìm kiếm được. Đây là một trong những kỹ thuật nền tảng trong lĩnh vực thị giác máy tính (Computer Vision) và xử lý ngôn ngữ tự nhiên (NLP), với nhiều ứng dụng thực tế như số hóa tài liệu, xử lý hóa đơn, trích xuất thông tin từ CMND, bằng lái xe, biển số xe và đặc biệt là danh thiếp – đối tượng nghiên cứu trong đề tài này.

Quá trình OCR bao gồm ba giai đoạn chính:

- Tiền xử lý ảnh (Pre-processing): Làm sạch ảnh, tăng cường độ tương phản, chuyển đổi ảnh màu sang ảnh nhị phân hoặc ảnh xám, loại bỏ nhiễu, hiệu chỉnh nghiêng,...
- Phát hiện vùng văn bản (Text Detection): Xác định các khối văn bản trong ảnh (bounding boxes).
- Nhận dạng ký tự (Text Recognition): Chuyển các vùng ảnh chứa văn bản thành chuỗi ký tự.

Hiện nay, các phương pháp OCR hiện đại thường sử dụng mạng nơ-ron sâu (deep neural networks), đặc biệt là sự kết hợp giữa CNN, RNN và các kiến trúc Transformer để xử lý hình ảnh và nhận dạng văn bản chính xác hơn nhiều so với các phương pháp truyền thống (dựa trên template, matching hoặc handcrafted features).

Một hệ thống OCR hiện đại thường được thiết kế theo kiến trúc pipeline gồm 2 tầng xử lý chính, như Hình 2.2 mô tả:



Hình 2.2: Luồng xử lý của các mô hình OCR

- Text Detection**:** Dùng các mạng CNN để phát hiện vùng văn bản. Ví dụ nổi bật gồm DBNet (Differentiable Binarization Network), EAST (Efficient and Accurate Scene Text detector), CTPN (Connectionist Text Proposal Network).
- Text Recognition**:** Sau khi cắt các vùng văn bản, hệ thống dùng RNN hoặc Transformer để đọc ký tự tuần tự từ ảnh vùng chữ. Các mô hình nổi bật gồm CRNN (Convolutional Recurrent Neural Network), STAR-Net, và Transformer-based recognizers.

Việc tách riêng hai giai đoạn detection và recognition giúp tăng độ chính xác và dễ tối ưu từng phần.

2.2.1. Một số công nghệ OCR phổ biến hiện nay

Dưới đây là một số framework OCR phổ biến, được đánh giá cao trong cả cộng đồng học thuật lẫn ứng dụng công nghiệp:

Tesseract OCR

Phát triển bởi HP, nay do Google duy trì.

Dựa trên mô hình LSTM để nhận dạng văn bản.

Hỗ trợ nhiều ngôn ngữ, có thể huấn luyện thêm.

- Ưu điểm: Nhẹ, dễ tích hợp.
- Nhược điểm: Độ chính xác không cao trong trường hợp ảnh phức tạp hoặc layout tự do như danh thiếp.

EasyOCR

Dựa trên PyTorch, hỗ trợ trên 80 ngôn ngữ.

Phát hiện văn bản bằng CRAFT, nhận dạng bằng CRNN.

- Ưu điểm: Cài đặt đơn giản, hỗ trợ tiếng Việt.
- Nhược điểm**:** Dễ sai trong layout phức tạp, ít tinh chỉnh theo domain cụ thể.

PaddleOCR

Phát triển bởi Baidu, mã nguồn mở, hiệu năng cao.

Kiến trúc: DBNet + CRNN (hoặc Transformer).

Có các bản pretrained cho tiếng Việt, tiếng Anh, Trung,...

Hỗ trợ phân tích bố cục (layout analysis), nhận dạng bảng (table recognition), đa ngôn ngữ (multilingual), mô-đun phát hiện và nhận dạng (detection/recognition) riêng biệt.

- Ưu điểm**:** Mạnh mẽ, chính xác, mở rộng tốt.
- Nhược điểm**:** Cần cấu hình kỹ khi tích hợp vào hệ thống riêng.

TrOCR (Transformer-based OCR – Microsoft)

Kiến trúc encoder-decoder giống như BART hoặc T5.

Sử dụng Vision Transformer (ViT) cho ảnh và Transformer Decoder cho text.

- Ưu điểm**:** Tối ưu cho tài liệu phức tạp, có thể fine-tune tốt.
- Nhược điểm**:** Cần GPU mạnh, thời gian inference lâu.

Bảng 2.1: So sánh các công nghệ OCR phổ biến

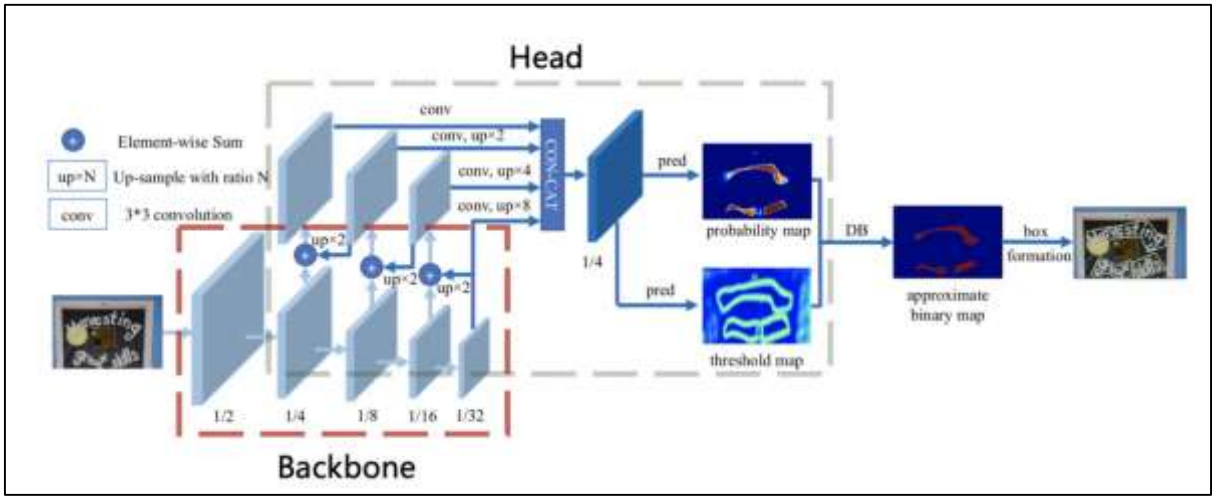
Công nghệ	**Độ chính xác**	**Đa ngôn ngữ**	**Tốc độ**
Tesseract	Trung bình	Có	Nhanh
EasyOCR	Tốt	Có	Khá
PaddleOCR	Rất tốt	Có	Tốt
TrOCR	Rất tốt	Có	Chậm

Trong phạm vi đồ án này, em quyết định chọn PaddleOCR vì các lý do chính sau:

- Có mô hình pretrained chất lượng cao cho tiếng Việt.
- Hỗ trợ detection và recognition riêng biệt, dễ kiểm soát.
- Hoạt động tốt trong môi trường danh thiếp với layout phức tạp.
- Có thể tùy chỉnh mô hình recognition riêng theo dữ liệu thực tế.

2.2.2. PaddleOCR

PaddleOCR là thư viện mã nguồn mở được phát triển bởi Baidu PaddlePaddle [6] Kiến trúc mô hình được nhóm tác giả đã đề xuất một hệ thống nhận dạng ký tự quang học (OCR) siêu nhẹ có tính thực tiễn cao mang tên PP-OCR. Mô hình được thiết kế với mục tiêu giảm thiểu kích thước và thời gian suy luận mà vẫn đảm bảo độ chính xác nhận dạng.



Hình 2.3: Kiến trúc mô hình PP-OCR của PaddleOCR [6]

Hình trên mô tả kiến trúc của mô hình PP-OCR, gồm các phần sau:

- Text Detector**:** DBNet, SAST, hoặc EAST.
- Text Recognizer**:** CRNN hoặc Transformer.
- Text Direction Classifier**:** giúp nhận dạng text bị xoay/ngược.
- Post-Processing**:** sử dụng CTC hoặc Attention-based decoding để chuyển logits thành chuỗi ký tự.

PaddleOCR còn hỗ trợ Layout Analysis giúp xác định khu vực logic như tiêu đề, bảng, đoạn – điều này đặc biệt hữu ích trong danh thiếp có nhiều vùng thông tin chồng chéo.

Nghiên cứu đã thành công trong việc phát triển một hệ thống OCR siêu nhẹ, hiệu quả và có tính thực tiễn cao, đồng thời công bố các mô hình và tập dữ liệu quy mô lớn để phục vụ cộng đồng nghiên cứu và phát triển ứng dụng. Cụ thể, kích thước toàn bộ mô hình chỉ 3.5MB khi nhận dạng 6622 ký tự tiếng Trung và 2.8MB khi nhận dạng 63 ký tự chữ số và tiếng Anh. Nhóm tác giả đã áp dụng một tập hợp các chiến lược để tối ưu hiệu quả mô hình cũng như làm nhẹ mô hình, đồng thời thực hiện các thí nghiệm ablation để đánh giá tác động của từng chiến lược.

2.3. Sửa chính tả từ OCR

Sau khi các thực thể có tên được nhận diện và phân loại từ văn bản danh thiếp bằng mô hình NER, vẫn có khả năng tồn tại các lỗi hoặc sự không nhất quán trong kết quả. Điều này có thể xuất phát từ những hạn chế của mô hình OCR trong việc nhận diện chính xác văn bản, hoặc từ sự mơ hồ và đa dạng trong cách thông tin được trình bày trên danh thiếp. Mô hình ngôn ngữ lớn (Large Language Models - LLM), với khả năng hiểu và tạo ra ngôn ngữ tự nhiên ở mức độ cao, có tiềm năng to lớn trong việc tinh chỉnh và chuẩn hóa các kết quả trích xuất này, đảm bảo tính chính xác và nhất quán của thông tin cuối cùng. Chương này sẽ giới thiệu về LLM, kiến trúc cơ bản của chúng, và cách chúng có thể được sử dụng để cải thiện chất lượng thông tin trích xuất từ danh thiếp.

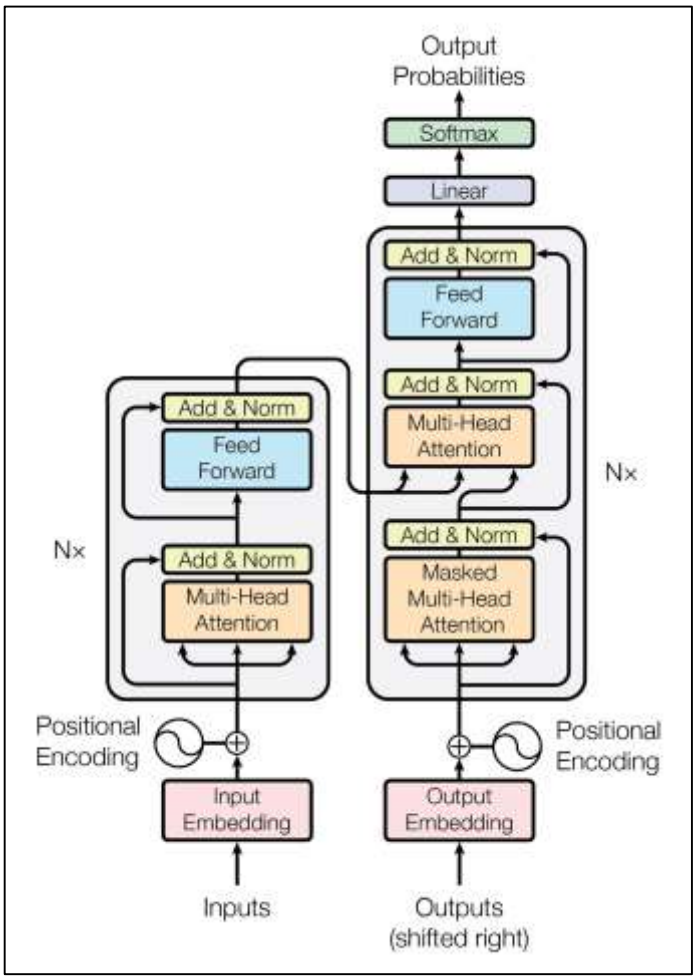
2.3.1. Tổng quan về mô hình ngôn ngữ lớn

Mô hình ngôn ngữ lớn (LLM) là một loại mô hình học sâu được huấn luyện trên một lượng khổng lồ dữ liệu văn bản và mã nguồn. Nhờ quy mô lớn về dữ liệu và số lượng tham số (thường lên đến hàng tỷ hoặc thậm chí hàng nghìn tỷ), LLM có khả năng hiểu, tóm tắt, dịch, và tạo ra văn bản giống như con người một cách đáng kinh ngạc. Các đặc điểm nổi bật của LLM bao gồm:

- Khả năng học ngữ cảnh (Contextual Understanding): LLM có thể hiểu ý nghĩa của một từ hoặc cụm từ dựa trên ngữ cảnh xung quanh nó trong một đoạn văn bản dài.
- Khả năng tạo sinh văn bản (Text Generation): LLM có thể tạo ra các đoạn văn bản mạch lạc, phù hợp với một chủ đề hoặc một phong cách nhất định.
- Khả năng thực hiện đa dạng các tác vụ NLP (Few-shot/Zero-shot Learning): Một số LLM tiên tiến có thể thực hiện các tác vụ NLP mới mà không cần hoặc chỉ cần một vài ví dụ (few-shot learning) hoặc thậm chí không cần bất kỳ ví dụ nào (zero-shot learning) thông qua việc đưa ra các prompt (mồi) phù hợp.
- Khả năng suy luận (Reasoning): Một số LLM thể hiện khả năng suy luận logic và trả lời các câu hỏi phức tạp.

2.3.2. Kiến trúc Transformer

Hầu hết các LLM hiện đại đều dựa trên kiến trúc **Transformer**, được giới thiệu lần đầu trong bài báo "Attention is All You Need". Kiến trúc này đã thay thế các mô hình dựa trên RNN trong nhiều tác vụ NLP nhờ khả năng xử lý song song và nắm bắt các phụ thuộc dài hạn hiệu quả hơn. Một kiến trúc Transformer cơ bản bao gồm:



Hình 2.4: Kiến trúc mô hình Transformer

Cơ chế Self-Attention (Self-Attention Mechanism): Đây là thành phần cốt lõi của Transformer, cho phép mô hình tính toán mức độ liên quan giữa các từ (tokens) khác nhau trong chuỗi đầu vào. Thay vì xử lý tuần tự như RNN, self-attention cho phép mỗi từ "tương tác" trực tiếp với tất cả các từ khác trong câu để xác định tầm quan trọng của chúng trong ngữ cảnh hiện tại. Cơ chế này bao gồm việc tính toán ba ma trận: Query (Q), Key (K), và Value (V) cho mỗi token, và sau đó sử dụng chúng để tính toán trọng số attention.

Multi-Head Attention: Để tăng cường khả năng biểu diễn, Transformer sử dụng nhiều "đầu" attention độc lập (multi-head attention), cho phép mô hình học được nhiều loại mối quan hệ khác nhau giữa các token.

Feed-Forward Networks: Sau mỗi lớp multi-head attention, một mạng nơ-ron feed-forward được áp dụng một cách độc lập cho mỗi vị trí token.

Add & Norm: Các kết nối residual (add) và layer normalization (norm) được sử dụng để ổn định quá trình huấn luyện và giúp mô hình học sâu hơn.

Encoder và Decoder: Kiến trúc Transformer ban đầu bao gồm cả một bộ mã hóa (encoder) và một bộ giải mã (decoder). Bộ mã hóa xử lý chuỗi đầu vào và tạo ra các biểu diễn ngữ cảnh. Bộ giải mã sử dụng các biểu diễn này để tạo ra chuỗi đầu ra. Tuy nhiên, nhiều LLM hiện đại chỉ sử dụng kiến trúc decoder-only (ví dụ: GPT) hoặc encoder-only (ví dụ: BERT cho các tác vụ hiểu ngôn ngữ).

2.3.3. Các mô hình LLM phổ biến và phân tích kiến trúc

Dưới đây là một số kiến trúc LLM phổ biến và phân tích ngắn gọn về chúng:

Mô hình GPT

Các mô hình GPT (Generative Pre-trained Transformer) sử dụng kiến trúc decoder-only của Transformer. Chúng được huấn luyện theo mục tiêu tự sinh (auto-regressive), tức là dự đoán token tiếp theo trong một chuỗi văn bản dựa trên các token đã xuất hiện trước đó. Các lớp Transformer decoder trong GPT bao gồm masked multi-head self-attention (masking đảm bảo rằng khi dự đoán một token, mô hình chỉ có thể nhìn vào các token ở vị trí trước đó) và các lớp feed-forward. Các phiên bản GPT khác nhau chủ yếu khác nhau về quy mô (số lượng lớp và tham số) và dữ liệu huấn luyện.

Kiến trúc decoder-only, tiêu biểu là các mô hình GPT, đặc biệt phù hợp cho các tác vụ sinh văn bản nhờ được thiết kế theo hướng tự sinh (auto-regressive). Khi được huấn luyện ở quy mô lớn, các mô hình này thể hiện năng lực few-shot và zero-shot learning rất ấn tượng, tức là có thể thực hiện nhiều tác vụ khác nhau chỉ với một số ít ví dụ hoặc thậm chí không cần ví dụ nào, chỉ

thông qua việc cung cấp prompt bằng ngôn ngữ tự nhiên. Điều này làm cho các mô hình GPT trở nên cực kỳ linh hoạt và mạnh mẽ trong các ứng dụng xử lý ngôn ngữ tự nhiên như tổng hợp văn bản, trả lời câu hỏi, và hội thoại tự động.

Tuy nhiên, do được huấn luyện theo cơ chế sinh tuần tự từ trái sang phải, GPT không khai thác trực tiếp ngữ cảnh hai chiều như các mô hình encoder-based như BERT. Điều này khiến GPT có thể kém hiệu quả hơn trong các tác vụ thiên về hiểu ngôn ngữ, nơi việc xem xét đầy đủ ngữ cảnh trước và sau của từ là rất quan trọng, chẳng hạn như phân loại văn bản hoặc trích xuất thực thể. Vì vậy, mặc dù nổi bật ở khả năng sinh ngôn ngữ tự nhiên, GPT không phải là lựa chọn tối ưu cho tất cả các loại tác vụ NLP.

Mô hình BERT

Các mô hình BERT sử dụng kiến trúc encoder-only của Transformer. Chúng được huấn luyện bằng hai mục tiêu chính: Masked Language Modeling (MLM) - dự đoán các token bị che giấu ngẫu nhiên trong câu - và Next Sentence Prediction (NSP) - dự đoán liệu một câu có phải là câu tiếp theo của câu trước đó trong văn bản gốc hay không (NSP đã được chứng minh là không quá hiệu quả và đã bị loại bỏ trong các phiên bản sau này như RoBERTa).

Kiến trúc encoder-only của BERT, cùng với phương pháp huấn luyện Masked Language Modeling (MLM), cho phép mô hình học được các biểu diễn ngữ cảnh hai chiều một cách sâu sắc. Nhờ khả năng đồng thời xem xét cả phần trước và sau của từ trong câu, BERT đặc biệt hiệu quả trong các tác vụ yêu cầu hiểu ngôn ngữ, chẳng hạn như phân loại văn bản, trích xuất thông tin, hay nhận dạng thực thể có tên (NER). Việc mô hình hóa ngữ cảnh toàn diện giúp BERT nắm bắt được ý nghĩa chính xác hơn của từ trong từng ngữ cảnh cụ thể, từ đó cải thiện đáng kể độ chính xác của các hệ thống xử lý ngôn ngữ tự nhiên dựa trên mô hình này.

Tuy nhiên, do được thiết kế và huấn luyện với mục tiêu chính là hiểu văn bản, BERT không được tối ưu hóa cho các tác vụ sinh văn bản tự do như GPT. Việc thiếu cơ chế sinh tuần tự và định hướng sinh nội dung khiến BERT gặp nhiều hạn chế trong các ứng dụng như viết tiếp văn bản, hội thoại tự động hoặc tổng hợp nội dung sáng tạo. Vì vậy, trong khi BERT vượt trội ở các tác vụ hiểu ngôn ngữ, nó không phải là lựa chọn hàng đầu cho những ứng dụng thiên về tạo sinh nội dung.

2.3.4. Sử dụng LLM để tinh chỉnh chính tả của OCR

Kết quả OCR thường chứa các lỗi chính tả do chất lượng hình ảnh thấp, font chữ không chuẩn, hoặc nhiễu nền. Việc sử dụng mô hình ngôn ngữ lớn (LLM) đã mở ra một hướng tiếp cận mới để tinh chỉnh và cải thiện độ chính xác của văn bản OCR. Phần này trình bày quy trình ứng dụng LLM trong việc sửa lỗi chính tả, phân tích các lỗi phổ biến của OCR, và thảo luận các phương pháp triển khai hiệu quả.

Kết quả OCR thường gặp phải các lỗi sau:

- Lỗi thay thế ký tự: Các ký tự tương tự về hình dạng bị nhận diện sai, ví dụ, "l" bị nhầm thành "1" hoặc "O" bị nhầm thành "0".
- Lỗi thiếu hoặc dư ký tự: OCR có thể bỏ sót hoặc thêm ký tự, đặc biệt trong các văn bản có độ phân giải thấp hoặc nhiễu nền, ví dụ, "học" bị nhận thành "hợc" hoặc "học sinh" thành "họcsinh".
- Lỗi ngữ cảnh: OCR không hiểu ngữ nghĩa, dẫn đến các từ không hợp lý trong văn bản, ví dụ, "máy tính" bị nhận thành "máy tình".
- Lỗi dấu câu và định dạng: Dấu câu bị bỏ qua hoặc nhận diện sai, ví dụ, dấu phẩy (,) bị nhầm thành dấu chấm (.) hoặc dấu cách bị bỏ sót.

- Lỗi do font chữ hoặc ngôn ngữ: Font chữ không chuẩn hoặc ngôn ngữ đặc thù (như tiếng Việt với dấu thanh) gây khó khăn trong việc nhận diện chính xác.

Việc sử dụng LLM trong xử lý hậu kỳ văn bản nhận dạng cần có kỹ thuật tạo câu prompt (prompt engineering). Prompt Engineering là phương pháp phổ biến nhất để tận dụng khả năng của các LLM lớn mà không cần huấn luyện lại toàn bộ mô hình. Chúng ta có thể thiết kế các prompt một cách cẩn thận, cung cấp cho LLM văn bản đã được trích xuất bởi OCR và NER cùng với các hướng dẫn cụ thể về cách chuẩn hóa, sửa lỗi hoặc cấu trúc lại thông tin.

Cấu trúc của câu prompt:

- Vai trò của LLM: Được định nghĩa là một trợ lý hữu ích, chuyên sửa lỗi và tinh chỉnh văn bản OCR từ danh thiếp.
- Dữ liệu đầu vào: Văn bản OCR được cung cấp dưới dạng chuỗi.
- Yêu cầu nhiệm vụ: Sửa các lỗi OCR phổ biến (như lỗi chính tả, thay thế ký tự) dựa trên đặc điểm ngôn ngữ được chỉ định (ví dụ, tiếng Việt).
- Định dạng đầu ra: Yêu cầu giữ nguyên định dạng gốc và chỉ trả về kết quả, không kèm bình luận.
- Lịch sử hội thoại (chat_history): Cung cấp các ví dụ cặp OCR và văn bản đã sửa để hướng dẫn LLM học ngữ cảnh và cách sửa lỗi.

Lựa chọn các tham số phù hợp:

- Temperature: Điều chỉnh mức độ ngẫu nhiên trong câu trả lời. Giá trị thấp (gần 0): kết quả chắc chắn, ít sáng tạo. Giá trị cao (gần 1 hoặc >1): kết quả đa dạng, sáng tạo hơn.

- Top_k: Chỉ chọn từ trong k từ có xác suất cao nhất. Ví dụ: top_k = 50 chỉ xét 50 từ đứng đầu. Giá trị thấp: an toàn, xác định; cao: đa dạng hơn
- Top_p (nucleus sampling): Giới hạn lựa chọn trong nhóm các từ có xác suất cộng dồn ≥ p. Ví dụ: top_p = 0.9 chỉ chọn từ trong nhóm chiếm 90% xác suất. Linh hoạt hơn top_k vì số lượng từ chọn là động.

Kỹ thuật prompt sử dụng LLM để tinh chỉnh chính tả OCR từ danh thiếp là một phương pháp mạnh mẽ, đặc biệt khi được thiết kế với lịch sử hội thoại và tham số mô hình phù hợp. Prompt được cung cấp tận dụng in-context learning để sửa các lỗi phổ biến như thay thế ký tự, lỗi dấu thanh, và định dạng sai, đồng thời đảm bảo giữ nguyên cấu trúc gốc của văn bản. Việc tối ưu hóa prompt bằng cách bổ sung ví dụ, cụ thể hóa yêu cầu ngôn ngữ, và điều chỉnh tham số mô hình có thể nâng cao đáng kể độ chính xác, đặc biệt với ngôn ngữ phức tạp như tiếng Việt. Trong tương lai, việc tích hợp thêm từ điển hoặc mô hình kiểm tra lỗi chuyên biệt sẽ giúp cải thiện hiệu suất và mở rộng khả năng ứng dụng của kỹ thuật này.

2.4. Nhận dạng thực thể có tên

Sau khi chuyển đổi hình ảnh danh thiếp thành văn bản thông qua OCR, bước tiếp theo quan trọng là xác định và phân loại các thông tin có ý nghĩa chứa trong đó, chẳng hạn như tên người, số điện thoại, địa chỉ email, tổ chức, địa chỉ, v.v. Đây chính là nhiệm vụ của Trích xuất thực thể có tên (Named Entity Recognition - NER). NER là một lĩnh vực con của xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) nhằm mục đích định vị và phân loại các thực thể có tên trong văn bản thành các nhóm được định nghĩa trước. Trong đồ án này, NER đóng vai trò then chốt trong việc cấu trúc hóa dữ liệu văn bản thu được từ OCR, biến nó thành thông tin có thể sử dụng được. Phần này sẽ trình bày lý thuyết cơ bản về NER, giới thiệu một số kiến trúc mô hình học sâu tiên

tiến thường được sử dụng cho NER, và thảo luận về quy trình tinh chỉnh các mô hình này để đạt được hiệu suất tốt nhất trên dữ liệu danh thiếp.

2.4.1. Tổng quan về trích xuất thực thể có tên

Trích xuất thực thể có tên (NER) là một nhiệm vụ trong NLP, tập trung vào việc xác định và phân loại các *thực thể có tên* (named entities) trong một đoạn văn bản. Các thực thể có tên là các đối tượng thực tế có một tên riêng, chẳng hạn như:

Ví dụ, từ văn bản:

- "Nguyễn Văn An – Trưởng phòng Kỹ thuật – Công ty ABC – 0977123456"

Kết quả của hệ thống NER có thể là:

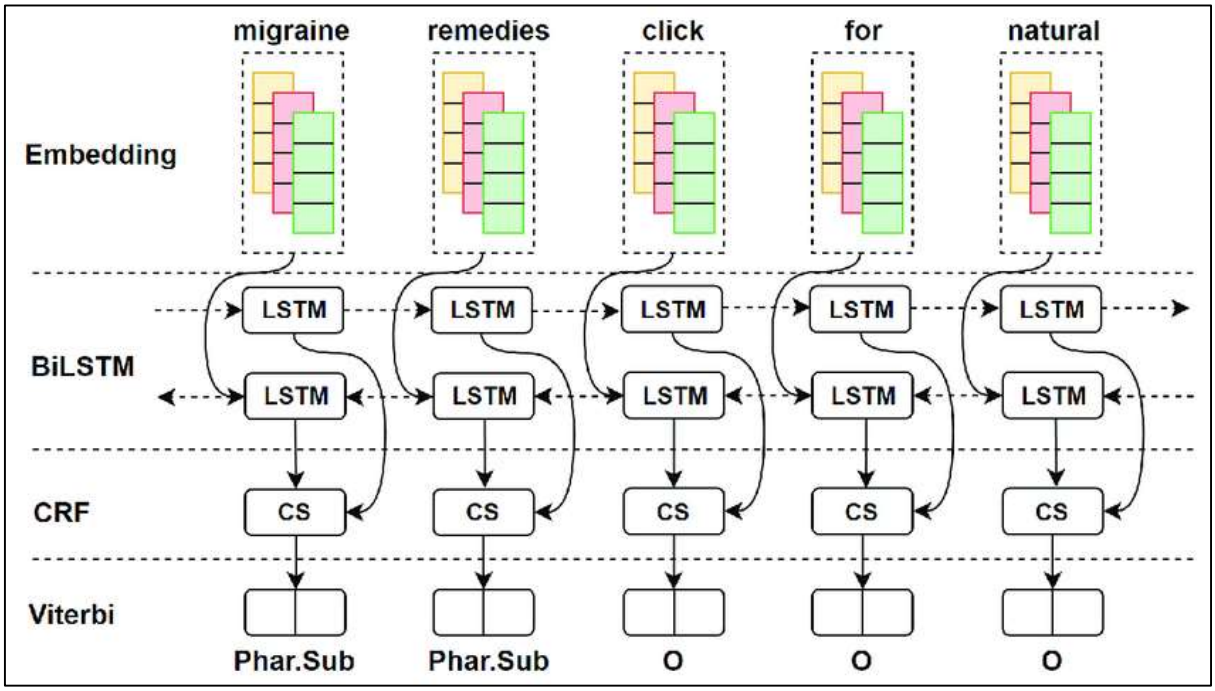
- Nguyễn Văn An → NAME
- Trưởng phòng Kỹ thuật → POSITION
- Công ty ABC → COMPANY
- 0977123456 → PHONE

NER thường là bước tiếp theo trong pipeline xử lý văn bản sau OCR, đặc biệt quan trọng trong các hệ thống trích xuất thông tin như hóa đơn, giấy tờ tùy thân, và đặc biệt là danh thiếp – nơi chứa các thông tin có cấu trúc bán tự do và ngữ cảnh ngắn gọn, khó đoán.

Với khả năng mô hình hóa các mối quan hệ dài hạn và nắm bắt ngữ cảnh toàn cục hiệu quả, các mô hình dựa trên Transformer như BERT, RoBERTa, và các biến thể của chúng đã trở thành những kiến trúc state-of-the-art cho nhiều tác vụ NLP, bao gồm cả NER. Chúng ta sẽ tập trung vào hai kiến trúc mô hình học sâu phổ biến và hiệu quả cho NER, đặc biệt là trong bối cảnh tinh chỉnh (finetuning): BiLSTM-CRF và Transformer-based Model (BERT).

2.4.2. Mô hình BiLSTM-CRF

Kiến trúc: Mô hình BiLSTM-CRF [7] kết hợp sức mạnh của mạng nơ-ron hồi quy hai chiều (BiLSTM) trong việc nắm bắt ngữ cảnh chuỗi và lớp Conditional Random Field (CRF) trong việc mô hình hóa các phụ thuộc giữa các nhãn NER. Kiến trúc này thường bao gồm các lớp sau:



Hình 2.5: Kiến trúc mô hình BiLSTM-CRF [7]

Lớp nhúng từ (Embedding Layer): Mỗi từ trong câu đầu vào được ánh xạ thành một vectơ biểu diễn mật độ thấp chiều cố định (word embedding). Các embedding này có thể được học từ đầu trong quá trình huấn luyện hoặc sử dụng các embedding được huấn luyện trước (pre-trained word embeddings) như Word2Vec, GloVe, FastText.

Lớp BiLSTM: Chuỗi các word embeddings được đưa vào một mạng BiLSTM. BiLSTM bao gồm hai LSTM: một LSTM xử lý chuỗi từ trái sang phải và một LSTM xử lý chuỗi từ phải sang trái. Bằng cách kết hợp đầu ra của cả hai LSTM tại mỗi vị trí thời gian (mỗi từ), mô hình có thể nắm bắt được thông tin ngữ cảnh từ cả phía trước và phía sau của từ đó. Mỗi từ trong câu

được ánh xạ thành một vector nhúng từ (word embedding). Sau đó, chuỗi các vector này được đưa vào một lớp BiLSTM để trích xuất các đặc trưng ngữ cảnh hai chiều (trước và sau):

$$\overrightarrow{h_t} = \text{LSTM}_{\text{forward}}\left(\text{x}_t, \overrightarrow{h_{t-1}}\right)$$

$$\overrightarrow{h_t} = \text{LSTM}_{\text{backward}}\left(\text{x}_t, \overrightarrow{h_{t+1}}\right)$$

Vector biểu diễn tại mỗi thời điểm t là phép nối (concatenation) của hai hướng:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$$

Trong đó:

- x_t là vector đầu vào tại thời điểm t.
- $\overrightarrow{h_t}$ và $\overleftarrow{h_t}$ lần lượt là hidden state theo chiều xuôi và chiều ngược.

Sau khi có được đặc trưng h_t tại mỗi vị trí, mô hình sử dụng một lớp tuyến tính để tính điểm số dự đoán (emission score) cho mỗi nhãn y tại vị trí t:

$$s_t = Wh_t + b$$

Trong đó:

- W là ma trận trọng số.
- b là vector bias.
- s_t là vector điểm số cho tất cả các nhãn tại vị trí t.

Lớp tuyến tính (Linear Layer): Đầu ra của lớp BiLSTM tại mỗi vị trí thời gian được chuyển qua một lớp tuyến tính để tạo ra một vectơ điểm số cho mỗi nhãn NER có thể có. Nếu có k nhãn NER, thì vectơ đầu ra sẽ có kích thước k.

Lớp CRF: Thay vì đưa ra quyết định nhãn độc lập cho mỗi từ dựa trên điểm số từ lớp tuyến tính, lớp CRF xem xét toàn bộ chuỗi nhãn và mô hình hóa

các phụ thuộc giữa các nhãn liền kề. Ví dụ, nhãn "B-PER" (bắt đầu của một người) thường đi trước nhãn "I-PER" (tiếp tục của một người) chứ không phải "I-ORG" (tiếp tục của một tổ chức). CRF học các ma trận chuyển đổi nhãn (transition probabilities) để đảm bảo tính nhất quán của chuỗi nhãn đầu ra. Lớp CRF sử dụng các điểm số s_t và học thêm các trọng số chuyển tiếp (transition scores) giữa các nhãn. Điểm số tổng cho toàn bộ chuỗi nhãn $y = (y_1, \dots, y_n)$ được định nghĩa là:

$$score(X, y) = t = \sum_{t=0}^{n} (T_{y_t, y_t} + s_{t+1, y_{t+1}})$$

Trong đó:

- $T_{i,j}$ là trọng số chuyển tiếp từ nhãn i sang nhãn j.
- s_{t, y_t} là điểm số phát xạ của nhãn y_t tại vị trí t.
- y_0 và y_{n+1} là nhãn đặc biệt (bắt đầu và kết thúc).

CRF sẽ tối đa hóa xác suất có điều kiện của chuỗi nhãn đúng:

$$P(y|X) = \frac{e^{score(X,y)}}{\sum_{\tilde{y} \in y(X)} e^{score(X,\tilde{y})}}$$

Trong đó: $P(X)$ là tập hợp tất cả các chuỗi nhãn hợp lệ cho chuỗi đầu vào X.

Khi đó, hàm mất mát (negative log-likelihood) cần tối ưu hóa là:

$$\mathcal{L} = -\log \mathrm{P}(y|X)$$

Trong quá trình suy luận, ta cần tìm chuỗi nhãn y^* có điểm số cao nhất:

$$y^* = \arg \max_{y \in y(X)} score(X, y)$$

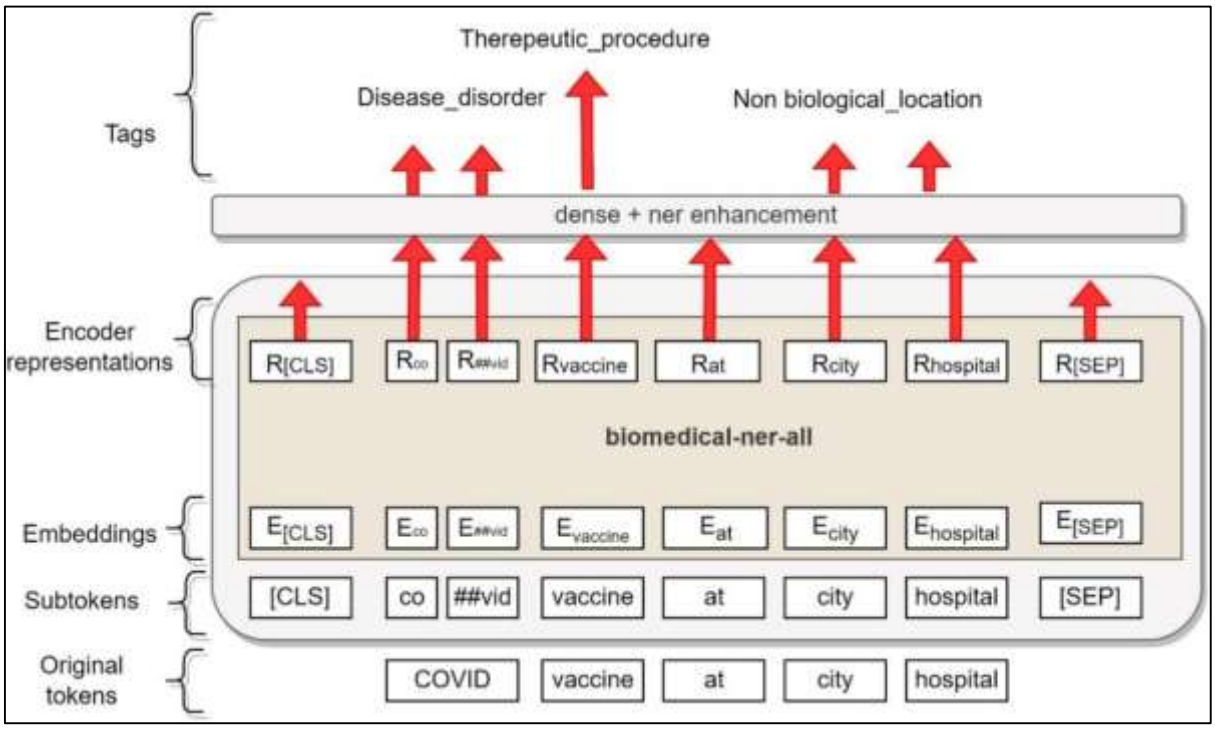
Việc này được thực hiện hiệu quả bằng thuật toán Viterbi.

Một trong những ưu điểm nổi bật của kiến trúc BiLSTM kết hợp với CRF là khả năng nắm bắt ngữ cảnh một cách hiệu quả. BiLSTM cho phép mô hình tiếp nhận thông tin từ cả hai chiều của chuỗi dữ liệu, giúp hiểu rõ hơn về ngữ nghĩa tổng thể của từng từ trong ngữ cảnh cụ thể. Khi kết hợp với CRF, hệ thống có thể cải thiện độ chính xác dự đoán bằng cách đảm bảo tính hợp lệ của chuỗi nhãn đầu ra, tránh được các tổ hợp nhãn không hợp lý. Kiến trúc này đã được chứng minh là hiệu quả trong nhiều tác vụ trích xuất thực thể có tên (NER) và được sử dụng rộng rãi trong các hệ thống xử lý ngôn ngữ tự nhiên.

Tuy nhiên, mô hình dựa trên RNN như BiLSTM vẫn tồn tại một số hạn chế. Điển hình là việc gặp khó khăn khi xử lý các chuỗi văn bản rất dài, do khả năng ghi nhớ các phụ thuộc dài hạn bị suy giảm. So với các kiến trúc hiện đại hơn như Transformer, vốn có khả năng mô hình hóa các mối quan hệ từ xa tốt hơn nhờ cơ chế attention, BiLSTM + CRF có thể kém hiệu quả hơn trong các ngữ cảnh yêu cầu xử lý sâu và phức tạp. Do đó, mặc dù vẫn mang lại kết quả tốt trong nhiều ứng dụng, kiến trúc này dần được thay thế trong các hệ thống NLP hiện đại.

2.4.3. Mô hình BERT-base

Các mô hình dựa trên Transformer, tiêu biểu là BERT (Bidirectional Encoder Representations from Transformers), đã đạt được những kết quả vượt trội trong nhiều tác vụ NLP nhờ kiến trúc Transformer mạnh mẽ và phương pháp pre-training hiệu quả trên lượng lớn dữ liệu văn bản [9]. Kiến trúc BERT bao gồm nhiều lớp Transformer encoder xếp chồng lên nhau. Mỗi encoder chứa các cơ chế self-attention và feed-forward.



Hình 2.6: Kiến trúc mô hình BERT [9]

Lớp nhúng (Embedding Layer): BERT sử dụng ba loại nhúng đầu vào:

- Token Embeddings**:** Nhúng vectơ cho mỗi token (từ hoặc subword) trong chuỗi đầu vào. BERT sử dụng WordPiece tokenization.
- Segment Embeddings**:** Nhúng để phân biệt giữa các câu khác nhau trong một cặp câu đầu vào (không quan trọng trong tác vụ NER trên một câu đơn).
- Position Embeddings**:** Nhúng để biểu diễn vị trí của mỗi token trong chuỗi, vì Transformer không có tính chất tuần tự vốn có như RNN.

Các lớp Transformer Encoder: Chuỗi các embeddings đầu vào được truyền qua nhiều lớp Transformer encoder. Mỗi encoder bao gồm một cơ chế multi-head self-attention, cho phép mô hình học các mối quan hệ giữa các token khác nhau trong chuỗi, và một mạng feed-forward hai lớp. Các kết nối residual và normalization được sử dụng để cải thiện quá trình huấn luyện.

Lớp đầu ra cho NER: Để thực hiện NER, đầu ra của lớp Transformer encoder tại vị trí của mỗi token thường được đưa qua một lớp tuyến tính để dự đoán nhãn NER cho token đó. Tương tự như BiLSTM, một lớp CRF cũng có thể được thêm vào sau lớp tuyến tính để mô hình hóa các phụ thuộc giữa các nhãn.

Ở mỗi lớp của encoder, BERT sử dụng cơ chế self-attention để tính toán sự liên hệ giữa các từ trong câu. Đầu tiên, mỗi từ x_i được ánh xạ thành 3 vector: Query (Q_i), Key (K_i) và Value (V_i) thông qua các phép biến đổi tuyến tính: $Q_i = W^Q x_i, K_i = W^K x_i, V_i = W^V x_i$.

Điểm attention giữa từ i và từ j được tính bằng:

$$Attention(i,j) = \frac{Q_i . K_j^T}{\sqrt{d_k}}$$

Trong đó, d_k là kích thước chiều của vector key, giúp việc tính attention ổn định hơn về mặt số học.

Sau đó, các attention scores được chuẩn hóa bằng hàm softmax:

$$\alpha_{ij} = softmax_j\left(\frac{Q_i . K_j^T}{\sqrt{d_k}}\right)$$

Và đầu ra tại vị trí i là tổng trọng số các vector value:

$$output_i = \sum_j \alpha_{ij} V_j$$

Self-attention cho phép mô hình nắm bắt quan hệ giữa bất kỳ cặp từ nào trong câu, bất kể khoảng cách, từ đó học được các phụ thuộc dài hạn một cách hiệu quả — điều mà các mô hình RNN truyền thống gặp khó khăn.

Trong quá trình tiền huấn luyện, BERT sử dụng nhiệm vụ Masked Language Modeling (MLM). Một số từ trong câu được thay thế bằng token đặc

biệt [MASK], và mô hình được yêu cầu dự đoán các từ bị che đó. Mục tiêu MLM là tối đa hóa xác suất các từ đúng tại các vị trí bị che:

$$\mathcal{L}_{MLM} = -\sum_{i \epsilon \mathcal{M}} \log P(x_i | X_{\mathcal{M}})$$

Trong đó:

- $\mathcal{M}$ là tập hợp các vị trí bị che.
- $X_{\mathcal{M}}$ là chuỗi đầu vào sau khi che.

Nhiệm vụ MLM giúp BERT học được biểu diễn ngữ nghĩa sâu sắc và đầy đủ hơn so với huấn luyện dự đoán từ tiếp theo đơn thuần, vì mô hình phải sử dụng thông tin từ cả hai hướng của chuỗi.

BERT còn sử dụng thêm nhiệm vụ Next Sentence Prediction (NSP) để học về mối quan hệ giữa các câu. Cho hai câu A và B, mô hình dự đoán liệu B có thực sự theo sau A trong tài liệu gốc hay không.

$$\mathcal{L}_{NSP} = -\log P(IsNext | A, B)$$

Nhiệm vụ NSP đặc biệt hữu ích cho các tác vụ như trả lời câu hỏi và suy luận văn bản, nơi cần hiểu quan hệ giữa các đoạn văn bản.

Sau khi pre-training, BERT có thể được tinh chỉnh (fine-tune) cho các tác vụ cụ thể bằng cách thêm một hoặc nhiều lớp đầu ra phù hợp với yêu cầu bài toán. Toàn bộ mô hình, bao gồm cả các trọng số pre-trained, được tối ưu hóa lại trong quá trình fine-tuning. Với tác vụ phân loại, xác suất nhãn y được tính bằng:

$$P(y|X) = softmax(W_o h_{[CLS]} + W_o)$$

Trong đó:

- $h_{[CLS]}$ là vector biểu diễn tại token [CLS].
- W_o, b_o là các tham số của lớp phân loại.

Quá trình fine-tuning cho phép BERT nhanh chóng thích nghi với nhiều loại tác vụ khác nhau chỉ với một lượng nhỏ dữ liệu huấn luyện, nhờ vào khả năng học biểu diễn ngôn ngữ mạnh mẽ trong quá trình pre-training.

Transformer là một kiến trúc mạnh mẽ với khả năng nắm bắt ngữ cảnh toàn cục rất tốt nhờ vào cơ chế self-attention. Cơ chế này cho phép mô hình quan sát toàn bộ chuỗi đầu vào cùng lúc và xác định được các mối liên hệ quan trọng giữa các từ, bất kể khoảng cách của chúng trong văn bản. Đặc biệt, các mô hình Transformer được huấn luyện sẵn (pre-trained) như BERT đã học được các biểu diễn ngôn ngữ phong phú từ lượng dữ liệu khổng lồ, giúp cải thiện đáng kể hiệu suất khi được tinh chỉnh cho các tác vụ cụ thể như nhận dạng thực thể có tên (NER). Nhờ vào khả năng học sâu về ngữ nghĩa và cấu trúc ngôn ngữ, các mô hình này vẫn có thể đạt kết quả tốt ngay cả khi dữ liệu huấn luyện cho tác vụ mục tiêu bị hạn chế.

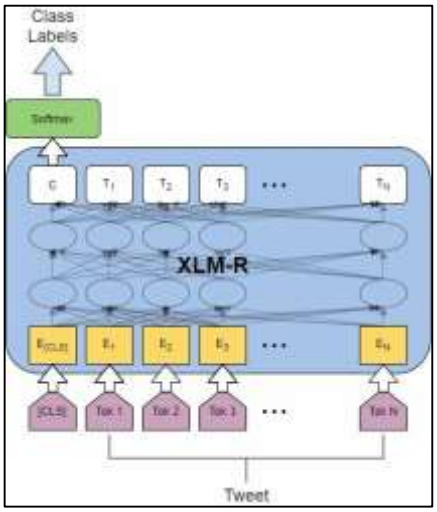
Tuy nhiên, điểm hạn chế của các mô hình Transformer là có số lượng tham số rất lớn, dẫn đến yêu cầu cao về tài nguyên tính toán trong cả quá trình huấn luyện và suy luận. Việc triển khai các mô hình này trên các thiết bị có cấu hình thấp hoặc trong các ứng dụng thời gian thực có thể gặp nhiều khó khăn. Do đó, trong khi Transformer mang lại hiệu quả vượt trội về mặt chất lượng, chi phí tính toán vẫn là một yếu tố cần cân nhắc trong thực tiễn triển khai.

2.4.4. Mô hình XLM-RoBERTa

XLM-RoBERTa (XLM-R) là một mô hình ngôn ngữ đa ngữ mạnh mẽ, được phát triển bởi Facebook AI vào năm 2019 [8], dựa trên kiến trúc Transformer và mô hình RoBERTa. Đây là một mô hình ngôn ngữ đa ngôn ngữ lớn, được đào tạo trên 2,5TB văn bản từ 100 ngôn ngữ khác nhau, sử dụng tokenizer kiểu SentencePiece với từ vựng chung gồm 250.000 token. Điều này giúp mô hình học được biểu diễn ngữ nghĩa có thể chia sẻ giữa nhiều ngôn ngữ, bao gồm cả những ngôn ngữ có dữ liệu ít. XLM-RoBERTa được huấn luyện

trên một tập dữ liệu đa ngữ lớn hơn nhiều so với các mô hình tiền nhiệm như mBERT, và không sử dụng thông tin gắn nhãn ngôn ngữ (language ID) hay phân đoạn câu, điều này giúp mô hình học được các biểu diễn ngôn ngữ ngữ cảnh sâu hơn và thống nhất hơn giữa các ngôn ngữ.

XLM-RoBERTa kế thừa kiến trúc từ BERT và RoBERTa, tức là kiến trúc Transformer hai chiều (bidirectional Transformer), bao gồm nhiều lớp encoder chồng lên nhau. Mỗi lớp encoder bao gồm hai thành phần chính: cơ chế attention đa đầu (multi-head self-attention) và mạng feed-forward hoàn toàn kết nối (position-wise feed-forward network).



Hình 2.7: Kiến trúc mô hình XLM-RoBERTa [8]

Cụ thể, với một chuỗi đầu vào được mã hóa thành embedding vector $X \in R^{n \times d}$, mô hình áp dụng cơ chế attention như sau:

$$\text{Attention}(\text{Q}, \text{K}, \text{V}) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Trong đó:

- $Q = XW^Q$, $K = XW^K$, $V = XW^V$ là các ma trận truy vấn, khóa và giá trị.

- d_k là kích thước của vector khóa.
- W^Q, W^K, W^V là các ma trận trọng số học được.

Cơ chế multi-head attention được định nghĩa như sau:

$$MultiHead(X) = Concat(head_1, \dots, head_h)W^O$$

Trong đó, $head_i = Attention(\mathrm{X}W_i^Q, \mathrm{X}W_i^K, \mathrm{X}W_i^V)$

Tiếp theo attention là mạng feed-forward áp dụng độc lập cho từng vị trí:

$$FFN(x) = ReLU(xW_1 + b_1)W_2 + b_2$$

XLM-RoBERTa sử dụng tổng cộng 24 lớp encoder (với phiên bản lớn – XLM-R Large), 16 đầu attention, và kích thước ẩn 1024 chiều.

Khác với BERT sử dụng mục tiêu Masked Language Modeling (MLM) và Next Sentence Prediction (NSP), RoBERTa và XLM-RoBERTa chỉ giữ lại MLM, nhưng tăng hiệu quả bằng cách mở rộng tập dữ liệu và cải thiện kỹ thuật huấn luyện. Cụ thể, mục tiêu của MLM là dự đoán từ bị che (masked) dựa trên ngữ cảnh xung quanh:

$$\mathcal{L}_{MLM} = -\sum_{i \epsilon \mathcal{M}} \log P(w_i | X_{\mathcal{M}})$$

Trong đó:

- $\mathcal{M}$ là tập các vị trí từ bị che.
- w_i là từ gốc ở vị trí i.
- $X_{\mathcal{M}}$ là chuỗi đầu vào sau khi che các vị trí trong $\mathcal{M}$.

XLM-R là mô hình đa ngữ mạnh mẽ vượt trội hơn mBERT nhờ được huấn luyện trên lượng dữ liệu lớn mà không phụ thuộc vào nhãn ngôn ngữ, giúp cải thiện khả năng học chuyển tiếp liên ngôn ngữ và thể hiện tốt trong các tác vụ zero-shot. Tuy nhiên, mô hình đòi hỏi chi phí tính toán cao, thiếu kiến thức

về cấu trúc ngôn ngữ cụ thể và vẫn gặp vấn đề về định kiến, đặc biệt với các ngôn ngữ ít tài nguyên.

2.4.5. Tinh chỉnh mô hình NER trên dữ liệu danh thiếp

Quá trình tinh chỉnh một mô hình pre-trained (chẳng hạn như BERT) hoặc một mô hình được huấn luyện trên một tập dữ liệu NER tổng quát (chẳng hạn như BiLSTM-CRF) cho tác vụ NER trên dữ liệu danh thiếp bao gồm các bước chính sau:

Chuẩn bị dữ liệu: Thu thập và gán nhãn cho dữ liệu danh thiếp. Mỗi từ (hoặc token) trong văn bản OCR được gán một nhãn NER tương ứng với loại thông tin mà nó biểu thị (ví dụ: "B-NAME", "I-NAME", "B-PHONE", "O" cho các từ không phải là thực thể). Định dạng dữ liệu thường là một danh sách các câu, trong đó mỗi câu là một danh sách các cặp (từ, nhãn).

Điều chỉnh kiến trúc mô hình (nếu cần): Đối với các mô hình pre-trained như BERT, lớp đầu ra được thay thế bằng một lớp phân loại có số lượng đầu ra bằng với số lượng nhãn NER (bao gồm cả nhãn "O"). Đối với các mô hình như BiLSTM-CRF, kiến trúc cơ bản thường được giữ nguyên.

Huấn luyện (Tinh chỉnh): Sử dụng dữ liệu danh thiếp đã được gán nhãn để huấn luyện mô hình. Quá trình này bao gồm việc điều chỉnh các trọng số của mô hình để cực đại hóa khả năng dự đoán đúng các nhãn NER cho các từ trong dữ liệu huấn luyện. Các kỹ thuật như learning rate scheduling, weight decay, và dropout thường được sử dụng để tối ưu hóa quá trình huấn luyện và tránh overfitting.

Đánh giá: Sau khi huấn luyện, mô hình được đánh giá trên một tập dữ liệu kiểm tra độc lập để đo lường hiệu suất của nó. Các chỉ số đánh giá phổ biến cho NER bao gồm Precision, Recall, F1-score (tính cho từng loại thực thể và tổng thể).

Với BiLSTM

Chuỗi đầu vào $X = (x_1, x_2, \dots, x_n)$ được ánh xạ thành chuỗi ẩn $H = (h_1, h_2, \dots, h_n)$ thông qua mô hình BiLSTM.

$$h_i = BiLSTM(x_i)$$

BiLSTM kết hợp thông tin ngữ cảnh từ cả quá khứ và tương lai cho từng vị trí token.

Với BERT và XLM-RoBERTa

Chuỗi đầu vào được mã hóa bởi Transformer encoder thành các biểu diễn ẩn:

$$h_i = model(x_i)$$

Nhờ cơ chế self-attention, mỗi vector h_i đã tích hợp ngữ cảnh hai chiều toàn cục

Ở giai đoạn này, dù có khác kiến trúc, cả BiLSMT và BERT đều tạo ra một chuỗi biểu diễn ẩn h_i cho từng token, chứa thông tin ngữ nghĩa phong phú

Cho mỗi biểu diễn h_i ta tính toán xác suất y_i bằng softmax:

$$P(y_i|h_i) = \text{softmax}\,(\mathrm{W}h_i + b)$$

Trong đó:

- W là ma trận trọng số,
- b là vector bias

Nếu chỉ dùng softmax cho từng token độc lập, hàm mất mát (loss) trên toàn bộ chuỗi là tổng cross-entropy:

$$\mathcal{L}_{token} = \sum_{i=1}^{n} \log P\,(y_i^{true}|h_i)$$

Cách dự đoán độc lập này nhanh và đơn giản nhưng có thể không đảm bảo sự hợp lệ của chuỗi nhãn (ví dụ nhãn "I-Company" không nên xuất hiện ngay sau "O").

Để cải thiện tính hợp lý chuỗi nhãn, một lớp CRF (Conditional Random Field) được thêm sau BiLSTM hoặc BERT.

Xác suất cho toàn bộ chuỗi nhãn $y = (y_1, y_2, \dots, y_n)$ được tính:

$$P(y|H) = \frac{\exp(Score(H, y))}{\sum_{y'} \exp(Score\ (H, y'))}$$

Trong đó Score được định nghĩa:

$$\text{Score}(\text{H}, \text{y}) = \sum_{i=1}^{n} \left(T_{y_i-1, y_i} + E_{i, y_i}\right)$$

Với:

- T là ma trận điểm chuyển tiếp giữa các nhãn.
- E_{i,y_i} là điểm emission từ vector ẩn h_i cho nhãn y_i.
- Loss CRF cần tối đa hóa xác suất chuỗi nhãn đúng:

$$\mathcal{L}_{CRF} = -\log P(y^{true}|H)$$

CRF bổ sung thêm sự ràng buộc giữa các nhãn liền kề, ví dụ như đảm bảo rằng sau "B-Company" thì chỉ có thể là "I-Company" hoặc "O", chứ không thể "B-Name", giúp mô hình NER có độ chính xác cao hơn và nhãn chuỗi hợp lý hơn.

Trong bối cảnh đồ án này, việc sử dụng một mô hình pre-trained dựa trên Transformer như BERT (hoặc một biến thể đa ngôn ngữ như mBERT) có nhiều lợi thế. Các mô hình này đã được huấn luyện trên lượng lớn dữ liệu đa dạng, giúp chúng nắm bắt được các đặc trưng ngôn ngữ chung. Việc tinh chỉnh BERT

trên một tập dữ liệu danh thiếp đã được gán nhãn có thể mang lại hiệu suất cao ngay cả khi kích thước của tập dữ liệu này không quá lớn.

Ngoài ra, một mô hình BiLSTM-CRF được huấn luyện từ đầu hoặc sử dụng các word embeddings pre-trained cho tiếng Việt cũng là một lựa chọn khả thi, đặc biệt nếu tài nguyên tính toán hạn chế hơn.

Việc lựa chọn cuối cùng sẽ phụ thuộc vào kích thước và chất lượng của dữz liệu danh thiếp được thu thập và gán nhãn, cũng như các ràng buộc về tài nguyên và thời gian. Cả hai kiến trúc BERT và BiLSTM-CRF đều là những lựa chọn mạnh mẽ cho tác vụ NER và có thể được tinh chỉnh hiệu quả cho bài toán trích xuất thông tin từ danh thiếp.

CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ

3.1. Dữ liệu thực nghiệm

Dữ liệu được sử dụng trong báo cáo này là dữ liệu tổng hợp (synthetic data), được tạo ra nhân tạo thay vì thu thập từ các sự kiện thực tế. Việc sử dụng dữ liệu tổng hợp mang lại nhiều lợi ích như tăng cường bảo mật, mở rộng khối lượng dữ liệu và cải thiện độ đa dạng của các mẫu huấn luyện, đồng thời giảm sự phụ thuộc vào dữ liệu thực có thể chứa thông tin nhạy cảm.

Trong nghiên cứu này, chúng tôi đã sử dụng các mô hình ngôn ngữ lớn (Large Language Models - LLMs) như ChatGPT, Gemini và Grok để sinh dữ liệu thực tế giả dựa trên các mẫu danh thiếp thật. Quá trình này tạo ra hơn 6000 cặp mẫu huấn luyện với trên 100 ngữ cảnh khác nhau, bao gồm nhiều ngôn ngữ như tiếng Việt, tiếng Anh, tiếng Hàn và tiếng Nhật, giúp mô hình học được sự đa dạng ngữ nghĩa và cấu trúc của các danh thiếp từ nhiều quốc gia.

Data cụ thể được công khai ở google driver.

Với các dạng json được format theo chuẩn IOB (Inside-Outside-Beginning) - một định dạng phổ biến cho bài toán NER như sau:

Tập dữ liệu sử dụng bộ nhãn được chuẩn hóa, bao gồm 15 loại để nhận diện 7 thực thể trong danh thiếp được chia thành các nhãn đơn và nhãn chuỗi, cụ thể như sau:

- O: Từ không thuộc bất kỳ thực thể nào.
- B-Name, I-Name: Tên người, bắt đầu (B-) và tiếp tục (I-) của thực thể "Name".
- B-Position, I-Position: Chức vụ, bắt đầu (B-) và tiếp tục (I-) của thực thể "Position".

- B-Company, I-Company: Tên công ty, bắt đầu (B-) và tiếp tục (I-) của thực thể "Company".
- B-Address, I-Address: Địa chỉ, bắt đầu (B-) và tiếp tục (I-) của thực thể "Address".
- B-Phone, I-Phone: Số điện thoại, bắt đầu (B-) và tiếp tục (I-) của thực thể "Phone".
- B-Email, I-Email: Địa chỉ email, bắt đầu (B-) và tiếp tục (I-) của thực thể "Email".
- B-Department, I-Department: Phòng ban, bắt đầu (B-) và tiếp tục (I-) của thực thể "Department".

Dữ liệu được lưu trữ dưới dạng JSON, với mỗi mục trong tệp dữ liệu bao gồm:

- tokens: Danh sách các từ (tokens) đã được tách từ văn bản OCR.
- labels: Danh sách các nhãn tương ứng với từng từ trong tokens, tuân theo định dạng B- và I- cho các thực thể nhiều từ và O cho từ không thuộc thực thể.

Ví dụ về cấu trúc của một mục dữ liệu như sau:

```
{
    "tokens": ["Nguyễn", "Văn", "An", "Manager", "ABC", "Company", "123", "Cầu", "Diễn", "an.nguyen@abc.com", "0977123456"],
    "labels": ["B-Name", "I-Name", "I-Name", "B-Position", "B-Company", "I-Company", "B-Address", "I-Address", "I-Address", "B-Email", "B-Phone"]
}
```

Trong mẫu ví dụ này:

- “Nguyễn Văn An” được nhận diện là một thực thể "Name", với "Nguyễn" là từ đầu tiên (B-Name) và "Văn An" là phần tiếp theo (I-Name).
- “Manager” là chức vụ (B-Position).
- “ABC Company” là tên công ty, với "ABC" là từ đầu tiên (B-Company) và "Company" là phần tiếp theo (I-Company).
- “123 Cầu Diễn” là địa chỉ, với "123" là từ đầu (B-Address) và "Cầu Diễn" là phần tiếp theo (I-Address).
- “an.nguyen@abc.com” là địa chỉ email (B-Email).
- “0977123456” là số điện thoại (B-Phone).

Ví dụ này cho thấy khả năng của định dạng IOB trong việc nhận diện các thực thể nhiều từ và phân tách rõ ràng giữa các loại thông tin khác nhau trên danh thiếp. Việc sử dụng dữ liệu tổng hợp với định dạng IOB cho phép mô hình học sâu nhận diện được cấu trúc phức tạp của thông tin trên danh thiếp, từ đó cải thiện độ chính xác và khả năng khái quát hóa khi triển khai thực tế.

3.2. Quy trình thực nghiệm

3.2.1. Tiền xử lý dữ liệu

Tiền xử lý dữ liệu là bước quan trọng trong pipeline huấn luyện mô hình học máy, đặc biệt đối với các bài toán Named Entity Recognition (NER). Mục tiêu của giai đoạn này là chuyển đổi dữ liệu thô thành dạng có thể sử dụng trực tiếp bởi mô hình, đồng thời tối ưu hóa độ chính xác và hiệu quả huấn luyện.

Trong trường hợp bài toán NER trên danh thiếp từ văn bản OCR, dữ liệu cần được chuẩn bị cẩn thận để duy trì mối quan hệ giữa các tokens và nhãn (labels) của chúng, giúp mô hình nhận diện chính xác các thực thể như tên, chức vụ, công ty, địa chỉ, số điện thoại, email và phòng ban.

Tokenization và gán nhãn

Tokenization là quá trình chuyển đổi văn bản thành các đơn vị nhỏ hơn, được gọi là tokens, để mô hình có thể xử lý. Đối với các mô hình ngôn ngữ, tokenization đóng vai trò quan trọng, bởi nó không chỉ cắt từ mà còn chuẩn hóa các ký tự đặc biệt và xử lý các từ phức hợp theo ngữ cảnh đa ngôn ngữ.

Mục tiêu chính của bước này là:

- Chia văn bản thành các tokens có độ dài cố định.
- Đảm bảo mỗi token nhận được nhãn (label) tương ứng từ dữ liệu đầu vào.
- Xử lý các token đặc biệt (như [CLS], [SEP], [PAD], <s>, </s>) mà mô hình sử dụng để đánh dấu ranh giới câu và điền vào các vị trí trống.

Phương pháp căn chỉnh nhãn

Việc căn chỉnh nhãn với các token là một thách thức đặc biệt, do quá trình tokenization có thể phá vỡ cấu trúc từ gốc, tạo ra nhiều token từ một từ ban đầu. Để giải quyết vấn đề này, phương pháp căn chỉnh nhãn (label alignment) được sử dụng với các nguyên tắc:

- Gán nhãn cho token đầu tiên của mỗi từ theo nhãn gốc.
- Gán nhãn cho các token tiếp theo bằng nhãn kiểu I- nếu từ đó ban đầu thuộc một thực thể có nhiều từ.
- Bỏ qua các token đặc biệt bằng cách gán giá trị -100, giúp mô hình không tính chúng vào quá trình tính loss khi huấn luyện.

Ví dụ, nếu từ "Nguyễn Văn An" (có nhãn "B-Name, I-Name, I-Name") được token hóa thành ["Nguyễn", "Văn", "An"], thì các token này sẽ được gán lần lượt là B-Name, I-Name, I-Name. Tuy nhiên, nếu từ "Nguyễn" bị phân chia

thành ["Ng", "uy", "ễn"], nhãn "B-Name" chỉ áp dụng cho token đầu tiên, còn các token sau sẽ nhận nhãn "I-Name".

```python
def tokenize_and_align_labels(examples):
    # print(examples)
    tokenized_inputs = tokenizer(
        examples["tokens"],
        truncation=True,
        padding="max_length",
        max_length=128,
        is_split_into_words=True
    )

    labels = []
    word_ids = tokenized_inputs.word_ids()
    previous_word_idx = None

    for word_idx in word_ids:
        if word_idx is None:
            labels.append(-100)  # Token đặc biệt (CLS, SEP, PAD)
        elif word_idx < len(examples["ner_tags"]):  # Đảm bảo không vượt quá độ dài danh sách nhãn
            if word_idx != previous_word_idx:
                labels.append(label2id.get(examples["ner_tags"][word_idx], 0))
            else:
                prev_label = examples["ner_tags"][previous_word_idx]
                if prev_label.startswith("B-"):
                    labels.append(label2id.get("I-" + prev_label[2:], 0))
                else:
                    labels.append(label2id.get(prev_label, 0))
            previous_word_idx = word_idx
        else:
            labels.append(-100)  # Nếu word_idx vượt quá danh sách nhãn, bỏ qua token này

    tokenized_inputs["labels"] = labels
    return tokenized_inputs
```

Hình 3.1: Code tiền xử lý dữ liệu

Bước tiền xử lý dữ liệu không chỉ đảm bảo tính nhất quán và chính xác khi ánh xạ nhãn với tokens, mà còn giúp mô hình học được cấu trúc ngữ nghĩa phức tạp của danh thiếp. Điều này đóng vai trò quan trọng trong việc cải thiện hiệu suất của mô hình NER và đảm bảo mô hình có thể xử lý tốt các ngữ cảnh đa ngôn ngữ trong thực tế.

3.2.2. Tạo tập train test

Việc tạo tập huấn luyện (train set) và tập kiểm tra (test set) là bước quan trọng trong quy trình huấn luyện mô hình học máy, đặc biệt đối với các bài toán Named Entity Recognition (NER), nơi tính chính xác và độ tổng quát của mô

hình phụ thuộc rất lớn vào cách chia dữ liệu. Để đảm bảo mô hình không bị lệch hoặc quá khớp (overfitting) do sự khác biệt về phân phối dữ liệu, phương pháp k-fold cross-validation được áp dụng, một kỹ thuật phổ biến giúp đánh giá mô hình một cách khách quan hơn.

K-Fold Cross-Validation là một chiến lược đánh giá mô hình mạnh mẽ, trong đó tập dữ liệu được chia thành k phần (folds) bằng nhau. Quá trình này bao gồm các bước:

- Chia tập dữ liệu thành k phần bằng nhau.
- Ở mỗi lần lặp (fold), sử dụng một phần để kiểm tra (test set), và k-1 phần còn lại để huấn luyện (train set).
- Lặp lại quá trình trên k lần để đảm bảo tất cả các mẫu dữ liệu đều được sử dụng cả cho huấn luyện và kiểm tra.
- Trung bình các kết quả đánh giá từ k lần lặp để có được hiệu suất tổng thể của mô hình.

```
data = load_data(json_path)
dataset = Dataset.from_list([tokenize_and_align_labels(sample) for sample in data])

kf = KFold(n_splits=k, shuffle=True, random_state=42)
```

Hình 3.2: Code chia tập dữ liệu

Với k = 10 được sử dụng trong báo cáo này, dữ liệu sẽ được chia thành 10 phần bằng nhau, với mỗi phần lần lượt đóng vai trò là test set, giúp giảm thiểu nguy cơ overfitting và đảm bảo đánh giá mô hình một cách khách quan.

K-Fold Cross-Validation mang lại nhiều lợi ích như tối ưu hóa việc sử dụng dữ liệu khi mỗi mẫu đều tham gia cả huấn luyện và kiểm tra, giảm bias nhờ đánh giá không phụ thuộc vào một tập kiểm tra cố định, giảm variance bằng cách lấy trung bình trên nhiều lần lặp để mô hình ổn định hơn, và giúp phát hiện sớm tình trạng overfitting khi mô hình quá khớp với tập huấn luyện.

Để đảm bảo mỗi fold là đại diện tốt cho toàn bộ tập dữ liệu, các mẫu được shuffle (xáo trộn) trước khi phân chia. Điều này giúp loại bỏ sự lệch do sắp xếp ban đầu của dữ liệu, chẳng hạn như các mẫu từ cùng một ngữ cảnh hoặc ngôn ngữ nằm gần nhau. Ngoài ra, việc đặt random seed (ở đây là 42) giúp đảm bảo quá trình chia dữ liệu có thể tái lập (reproducibility), một yếu tố quan trọng trong nghiên cứu khoa học và kỹ thuật.

Sau khi áp dụng kỹ thuật k-fold cross-validation với k = 10, ta thu được 10 tập huấn luyện và kiểm tra khác nhau, sẵn sàng cho bước huấn luyện và đánh giá mô hình. Điều này không chỉ cải thiện độ chính xác của mô hình mà còn giúp đánh giá tính tổng quát hóa của mô hình một cách khách quan hơn.

3.2.3. Huấn luyện mô hình

Quá trình huấn luyện mô hình Named Entity Recognition (NER) là bước trung tâm của pipeline học máy, trong đó mô hình học cách phân loại từng token thành các thực thể được xác định trước như tên, chức vụ, công ty, địa chỉ, số điện thoại, email và phòng ban từ văn bản OCR của danh thiếp. Mô hình được huấn luyện trên nền tảng Google Colab với GPU T4, cho phép tận dụng khả năng tính toán song song và tăng tốc quá trình xử lý với phần cứng mạnh mẽ.

Trong đồ án này, ba mô hình tiền huấn luyện (pretrained models) đã được sử dụng, bao gồm:

- PassbyGrocer/bert_bilstm_crf-ner-weibo: Mô hình BERT tích hợp BiLSTM và CRF, chuyên cho các bài toán NER tiếng Trung.
- dslim/bert-base-NER: Mô hình BERT cơ bản, được huấn luyện trên tập dữ liệu NER chuẩn, phù hợp cho các ngôn ngữ như tiếng Anh.
- Davlan/xlm-roberta-base-ner-hrl: Mô hình XLM-RoBERTa đa ngôn ngữ, tối ưu cho các bài toán NER với dữ liệu từ nhiều ngôn ngữ như tiếng Việt, tiếng Nhật, tiếng Hàn và tiếng Anh.

Các mô hình này được lựa chọn dựa trên khả năng xử lý đa ngôn ngữ và hiệu quả trong các bài toán NER đã được chứng minh trên nhiều bộ dữ liệu khác nhau.

```
training_args = TrainingArguments(
    output_dir=f"./ner_fold{fold_id}",
    gradient_checkpointing=True,
    gradient_checkpointing_kwargs={'use_reentrant':False}, # suppress warnings
    save_strategy="epoch",
    eval_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=100,
    weight_decay=0.01,
    logging_dir=f"./logs_fold{fold_id}",
    logging_steps=10,
    logging_strategy="epoch",
    report_to="none",
    fp16=True,                                  # Use 16-bit floating point precision to reduce memory usage and speed up training.
    optim="paged_adamw_8bit",                   # Use an 8-bit AdamW optimizer for memory efficiency and faster computation.
    load_best_model_at_end=True,                # <-- Chỉ lưu mô hình tốt nhất
    metric_for_best_model="accuracy",                 # <-- Dựa theo F1
    greater_is_better=True,                     # <-- F1 càng lớn càng tốt
    save_total_limit=1                          # <-- Chỉ giữ 1 checkpoint tốt nhất
)
csv_logger = CSVLoggerCallback(log_path=f"./ner_fold{fold_id}/metrics.csv")

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=val_dataset,
    tokenizer=tokenizer,  # sửa tên đúng là "tokenizer", không phải "processing_class"
    data_collator=data_collator,
    compute_metrics=compute_metrics,
    callbacks=[csv_logger]
)
```

Hình 3.3: Code huấn luyện mô hình

Trong quá trình huấn luyện, các tham số quan trọng được thiết lập để đảm bảo sự tối ưu hóa mô hình, gồm có:

- Số epoch: 100 (giúp mô hình học đầy đủ từ tập dữ liệu)
- Batch size: 8 (phù hợp với khả năng xử lý của GPU T4)
- Learning rate: 2e-5 (tốc độ học nhỏ để tránh mất ổn định trong quá trình tối ưu)
- Weight decay: 0.01 (giúp giảm overfitting)
- Chiến lược lưu mô hình: Lưu checkpoint sau mỗi epoch và chỉ giữ lại mô hình tốt nhất, dựa trên tiêu chí độ chính xác (accuracy)

Tối ưu hóa và Kỹ thuật tiết kiệm bộ nhớ

- FP16 Precision: Sử dụng độ chính xác 16-bit (FP16) giúp tiết kiệm bộ nhớ và tăng tốc huấn luyện.
- AdamW Optimizer: Tối ưu hóa bằng paged_adamw_8bit, một phiên bản nhẹ và hiệu quả hơn, phù hợp với các mô hình có dung lượng lớn.
- Gradient Checkpointing: Kỹ thuật này giúp giảm bộ nhớ bằng cách lưu trữ ít hơn trong quá trình lan truyền ngược (backpropagation), đổi lại là tăng thời gian tính toán.

Do mô hình sử dụng chiến lược 10-fold cross-validation (đã đề cập ở mục trước), việc huấn luyện được thực hiện với mỗi fold để đảm bảo không có dữ liệu trùng lặp giữa tập huấn luyện và tập kiểm tra.

Việc sử dụng nền tảng Google Colab với GPU T4 cùng các mô hình pretrained mạnh mẽ đã giúp quá trình huấn luyện diễn ra nhanh chóng, hiệu quả và đảm bảo độ chính xác cao. Việc fine-tuning trên mô hình XLM-RoBERTa, BERT-base và BiLSTM-CRF cho phép mô hình học được các ngữ cảnh đa dạng, giúp cải thiện khả năng nhận diện các thực thể từ danh thiếp đa ngôn ngữ.

Trong các lần huấn luyện, mô hình đã hội tụ tốt và giữ được tính ổn định qua nhiều epoch nhờ sử dụng chiến lược gradient checkpointing và tối ưu hóa paged_adamw_8bit. Điều này khẳng định sự phù hợp của chiến lược huấn luyện đã lựa chọn đối với bài toán NER từ văn bản OCR danh thiếp.

3.2.4. Đánh giá

Đánh giá mô hình là bước quan trọng trong quy trình phát triển mô hình học máy, giúp xác định độ chính xác và khả năng tổng quát hóa của mô hình trên dữ liệu chưa thấy trước. Đối với bài toán Named Entity Recognition (NER) từ văn bản OCR danh thiếp, việc đánh giá cần đặc biệt chú ý đến độ chính xác trong việc nhận diện các thực thể như tên, chức vụ, công ty, địa chỉ, số điện

thoại, email và phòng ban, bởi các lỗi nhỏ có thể gây ảnh hưởng lớn đến hiệu suất tổng thể.

Quá trình đánh giá mô hình trong đồ án này sử dụng các chỉ số phổ biến sau:

- Validation Loss: Độ mất mát (loss) trên tập kiểm tra, đo lường mức độ khác biệt giữa dự đoán của mô hình và nhãn thực tế. Mức loss càng thấp, mô hình càng tốt.
- F1-score: Trung bình điều hòa giữa precision và recall, thể hiện độ cân bằng giữa tỷ lệ phát hiện đúng và tỷ lệ bỏ sót thực thể. Đây là chỉ số quan trọng nhất cho bài toán NER, đặc biệt khi dữ liệu không cân bằng.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- Precision (Độ chính xác): Tỷ lệ các token được dự đoán là thực thể và thực sự là thực thể. Được tính theo công thức:

$$Recall = \frac{TP}{TP + FN}$$

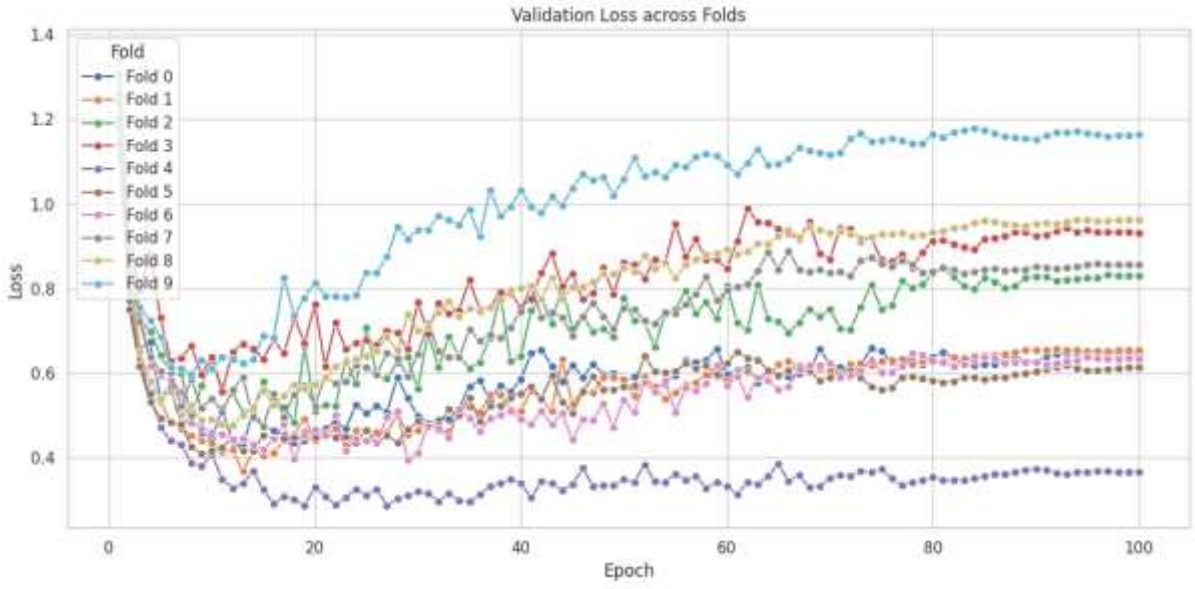
Trong đó:

- TP (True Positive): Số token dự đoán đúng là thực thể.
- FN (False Negative): Số token là thực thể nhưng mô hình không phát hiện được.
- Accuracy (Độ chính xác tổng thể): Tỷ lệ các token được mô hình dự đoán đúng (bao gồm cả thực thể và không phải thực thể). Đây là một chỉ số bao quát nhưng có thể không phản ánh chính xác hiệu suất của mô hình khi dữ liệu không cân bằng.

Một mô hình tốt sẽ có Validation Loss thấp và F1-score cao, cho thấy mô hình không chỉ dự đoán chính xác mà còn không bỏ sót các thực thể quan trọng. Tuy nhiên, mô hình cũng cần duy trì Precision và Recall cân bằng để tránh tình trạng phát hiện quá nhiều thực thể giả hoặc bỏ sót thực thể.

3.3. Kết quả đạt được

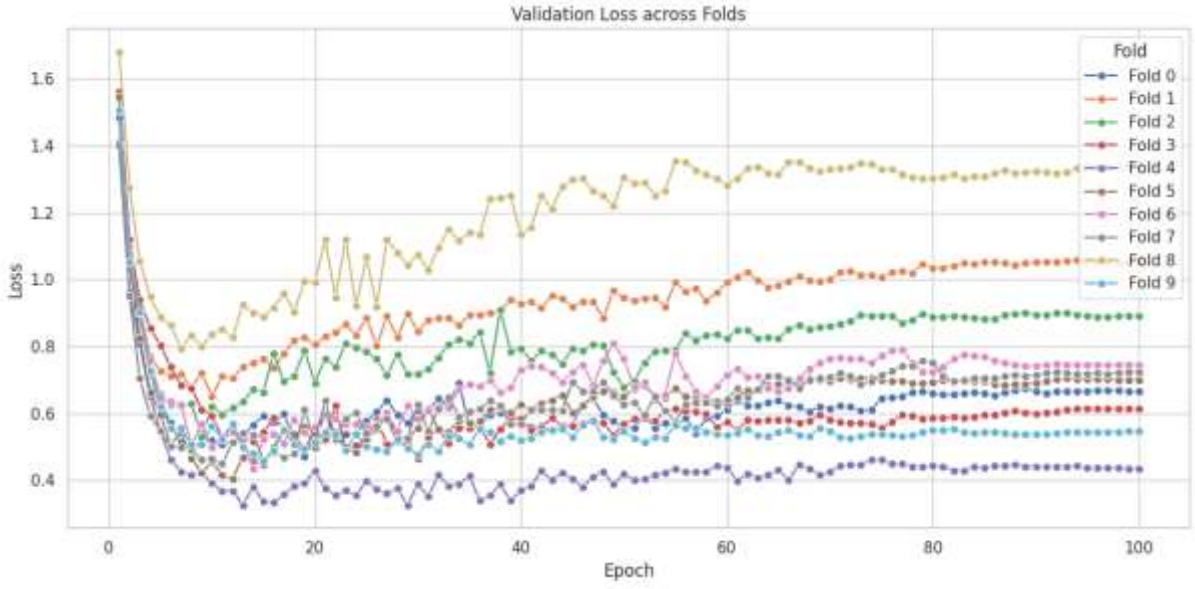
Sau khi hoàn thành quá trình huấn luyện, các mô hình được đánh giá dựa trên tập kiểm tra từ từng fold trong chiến lược k-fold cross-validation. Việc này giúp đảm bảo mô hình không chỉ hoạt động tốt trên tập huấn luyện mà còn có khả năng tổng quát hóa tốt trên dữ liệu thực tế, giảm thiểu rủi ro overfitting. Kết quả trung bình từ k = 10 fold của ba mô hình BiLSTM-CRF, BERT và XLM-RoBERTa được tổng hợp như sau:



Hình 3.4: Biểu đồ Validation Loss trên từng fold của mô hình BiLSTM-CRF

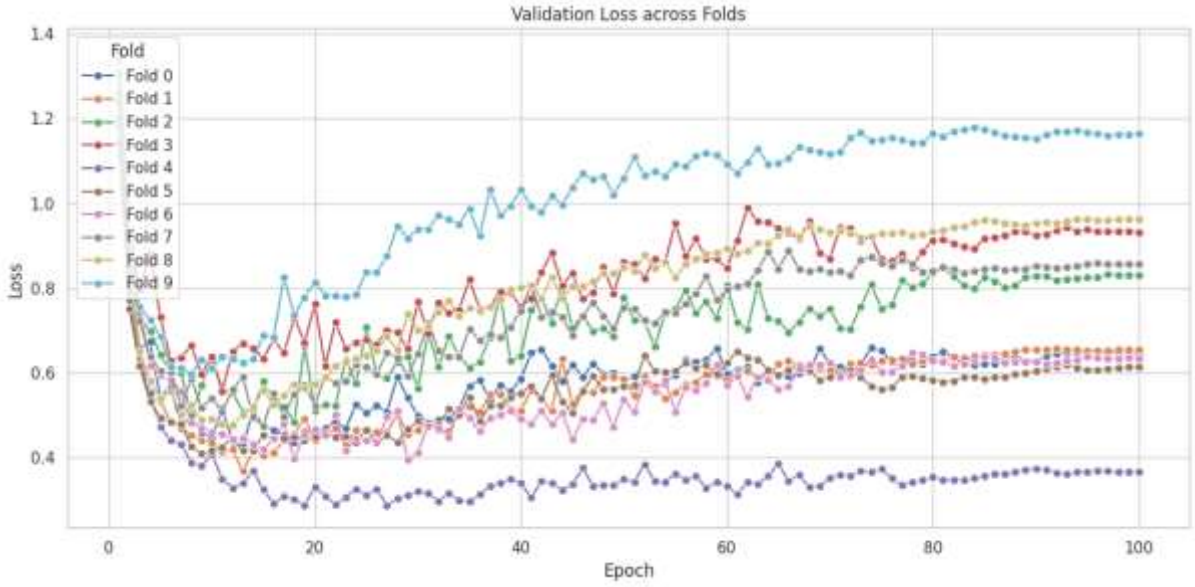
Biểu đồ Validation Loss của mô hình BiLSTM-CRF cho thấy quá trình hội tụ không ổn định. Mô hình này có xu hướng dao động mạnh sau khoảng 20-30 epoch, với một số fold thậm chí có Validation Loss tăng dần, thay vì giảm như mong đợi. Điều này cho thấy BiLSTM-CRF dễ bị overfitting khi số lượng epoch tăng, làm giảm khả năng tổng quát hóa trên dữ liệu chưa thấy trước. Đây

có thể là do mô hình này phụ thuộc nhiều vào cấu trúc tuần tự của BiLSTM và cơ chế CRF, khiến nó khó xử lý các mẫu dữ liệu đa ngôn ngữ phức tạp từ danh thiếp.



Hình 3.5: Biểu đồ Validation Loss trên từng fold của mô hình BERT

Biểu đồ của BERT cho thấy mô hình này hội tụ rõ ràng hơn so với BiLSTM-CRF, với Validation Loss giảm nhanh trong 10-20 epoch đầu. Tuy nhiên, sau đó, loss bắt đầu dao động và thậm chí có xu hướng tăng nhẹ ở một số fold. Điều này cho thấy mặc dù BERT có khả năng học biểu diễn từ ngữ tốt hơn, nhưng vẫn gặp khó khăn trong việc ổn định khi huấn luyện trên dữ liệu đa ngôn ngữ phức tạp. Điều này có thể do kiến trúc Transformer của BERT không được tối ưu cho các ngôn ngữ ngoài tiếng Anh.



Hình 3.6: Biểu đồ Validation Loss trên từng fold của mô hình XLM-RoBERTa

Ngược lại, biểu đồ của XLM-RoBERTa cho thấy mô hình này hội tụ nhanh và ổn định nhất trong ba mô hình, với Validation Loss liên tục giảm và duy trì ở mức thấp sau 20-30 epoch. Mặc dù vẫn có một số biến động nhỏ, nhưng mô hình này duy trì được mức loss thấp hơn đáng kể so với hai mô hình trước. Điều này phản ánh khả năng học tốt hơn từ dữ liệu đa ngôn ngữ, nhờ vào kiến trúc Transformer được thiết kế đặc biệt để xử lý nhiều ngôn ngữ khác nhau.

Bảng 3.1: Bảng so sánh kết quả trung bình từ k = 10 fold của ba mô hình BiLSTM-CRF, BERT và XLM-RoBERTa

Mô hình	**Validation Loss**	**F1-scroce**	**Precision**	**Recall**	**Accuracy**
BiLSTM-CRF	0.767s	0.593	0.553	0.553	0.872
BERT	0.771	0.612	0.557	0.683	0.869
XLM-RoBERTa	0.555	0.844	0.821	0.867	0.949

Kết quả cho thấy XLM-RoBERTa có Validation Loss thấp nhất (0.555), chứng tỏ mô hình này học được tốt nhất và ít bị overfitting so với BiLSTM-CRF (0.767) và BERT (0.771). Điều này phản ánh khả năng học biểu diễn ngôn ngữ mạnh mẽ của XLM-RoBERTa khi xử lý dữ liệu đa ngôn ngữ phức tạp từ danh thiếp.

XLM-RoBERTa cũng đạt F1-score cao nhất (0.844), vượt trội so với BiLSTM-CRF (0.593) và BERT (0.612). Điều này cho thấy XLM-RoBERTa không chỉ dự đoán chính xác mà còn có khả năng nhận diện đầy đủ các thực thể từ văn bản OCR, nhờ vào khả năng mô hình hóa ngữ nghĩa tốt hơn.

XLM-RoBERTa cũng đạt F1-score cao nhất (0.844), vượt trội so với BiLSTM-CRF (0.593) và BERT (0.612). Điều này cho thấy XLM-RoBERTa không chỉ dự đoán chính xác mà còn có khả năng nhận diện đầy đủ các thực thể từ văn bản OCR, nhờ vào khả năng mô hình hóa ngữ nghĩa tốt hơn.

Về Accuracy, XLM-RoBERTa đạt 0.949, gần đạt mức hoàn hảo, cho thấy mô hình này không chỉ nhận diện tốt các thực thể mà còn phân loại chính xác các từ không thuộc thực thể (O). Ngược lại, BiLSTM-CRF (0.872) và BERT (0.869) có Accuracy thấp hơn, thể hiện sự hạn chế khi xử lý các mẫu phức tạp hơn trong tập dữ liệu đa ngôn ngữ.

Từ các kết quả trên, có thể kết luận rằng XLM-RoBERTa là mô hình tốt nhất cho bài toán NER từ văn bản OCR danh thiếp, nhờ vào:

- Khả năng học biểu diễn từ nhiều ngôn ngữ.
- Độ chính xác và khả năng tổng quát hóa vượt trội.
- Hiệu quả trong việc cân bằng giữa precision và recall, giúp mô hình nhận diện đầy đủ và chính xác hơn các thực thể quan trọng.

Trong khi đó, các mô hình như BiLSTM-CRF và BERT có thể phù hợp với các bài toán NER đơn ngữ hoặc có cấu trúc đơn giản hơn, nhưng không đạt hiệu suất cao khi xử lý dữ liệu đa ngôn ngữ phức tạp như trong đồ án này.

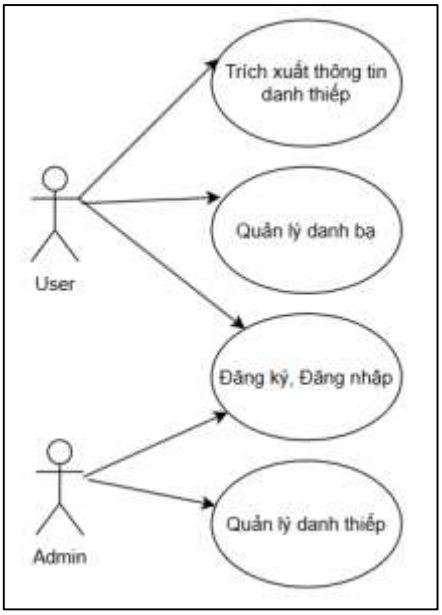
Dựa trên các kết quả đã phân tích, có thể kết luận rằng XLM-RoBERTa là mô hình tốt nhất cho bài toán NER từ văn bản OCR danh thiếp, nhờ khả năng hội tụ nhanh, loss thấp và độ ổn định cao. Trong khi đó, BiLSTM-CRF và BERT mặc dù vẫn đạt được kết quả chấp nhận được, nhưng cần được tinh chỉnh thêm để giảm thiểu overfitting và cải thiện khả năng tổng quát hóa, đặc biệt khi làm việc với dữ liệu đa ngôn ngữ phức tạp.

CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG TRÍCH XUẤT THÔNG TIN DANH THIẾP

4.1. Phân tích thiết kế ứng dụng demo

4.1.1. Các use case

Ứng dụng demo trong báo cáo chỉ mang một số chức năng cơ bản để demo cách người dùng cuối có thể sử dụng hệ thống AI trích xuất thông tin danh thiếp như thế nào. Một số use case cho các chức năng chính sẽ được trình bày dưới đây với 2 tác nhân là người dùng (user), người quản trị (admin).



Hình 4.1: Sơ đồ tổng quan use case

Chi tiết từng use case được mô tả chi tiết dưới đây:

Use case Đăng ký, đăng nhập

1. Tên use case: Đăng ký, đăng nhập

2. Mô tả vắn tắt: Use case này cho phép người dùng tạo tài khoản và đăng nhập trên hệ thống.

3. Luồng sự kiện

3.1. Luồng cơ bản:

Đăng ký

1) Use case này bắt đầu khi người dùng truy cập trang đăng ký. Hệ thống hiển thị biểu mẫu đăng ký gồm tên, email, mật khẩu.
2) Người dùng nhập thông tin cần thiết và nhấn nút "Đăng ký". Hệ thống kiểm tra hợp lệ thông tin đầu vào. Nếu thông tin hợp lệ, hệ thống tạo tài khoản và thông báo đăng ký thành công. Use case kết thúc.

Đăng nhập

1) Use case bắt đầu khi người dùng truy cập trang đăng nhập. Hệ thống hiển thị biểu mẫu đăng nhập gồm email và mật khẩu.
2) Người dùng nhập thông tin và nhấn nút "Đăng nhập". Hệ thống xác thực thông tin. Nếu đúng, hệ thống chuyển hướng người dùng đến trang chính. Use case kết thúc.

3.2. Luồng rẽ nhánh:

Đăng ký

1) Ở bước 2, nếu email đã tồn tại thì hệ thống hiển thị thông báo lỗi "Email đã được đăng ký". Thực hiện lại bước 2.

Đăng nhập

1) Ở bước 2, nếu thông tin không chính xác thì hệ thống hiển thị lỗi "Email hoặc mật khẩu không đúng".

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Không có

6. Hậu điều kiện: Không có

7. Điểm mở rộng: Không có

Use case Trích xuất thông tin danh thiếp

1. Tên use case: Trích xuất thông tin danh thiếp

2. Mô tả vắn tắt: Use case này cho phép người dùng tải ảnh danh thiếp, nhập ngôn ngữ, và trích xuất thông tin tự động.

3. Luồng sự kiện

3.1. Luồng cơ bản:

1) Use case bắt đầu khi người dùng đăng nhập và truy cập trang trích xuất. Hệ thống hiển thị biểu mẫu gồm: nút tải ảnh danh thiếp, lựa chọn ngôn ngữ, nút "Trích xuất".

2) Người dùng chọn ảnh danh thiếp, chọn ngôn ngữ, và nhấn nút "Trích xuất". Hệ thống gửi ảnh và ngôn ngữ lên server, xử lý bằng OCR. Trong lúc chờ xử lý sẽ hiển thị hiệu ứng loading.

1) Khi nhận được kết quả, hệ thống hiển thị thông tin trích xuất gồm: họ tên, email, số điện thoại, công ty, chức vụ, thông tin khác lên màn hình và cho phép người dùng chỉnh sửa kết quả.

2) Người dùng nhấn nút "Lưu" thông tin về danh thiếp sẽ được lưu trong hệ thống. Use case kết thúc.

3.2. Luồng rẽ nhánh:

1) Ở bước 3, nếu không trích xuất thông tin gì được từ ảnh, hệ thống sẽ hiển thị các mục dữ liệu trống.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Đã đăng nhập vào tài khoản người dùng

6. Hậu điều kiện: Không có

7. Điểm mở rộng: Không có

Use case Quản lý danh bạ

1. Tên use case: Quản lý danh bạ

2. Mô tả vắn tắt: Use case này cho phép người dùng xem các danh thiếp đã lưu.

3. Luồng sự kiện

3.1. Luồng cơ bản:

1) Use case bắt đầu khi người dùng ở trang chủ của mình. Hệ thống hiển thị danh sách tất cả danh thiếp.

2) Người dùng có thể xem chi tiết danh thiếp và chỉnh sửa chúng. Hệ thống cập nhật tương ứng theo hành động. Use case kết thúc.

3.2. Luồng rẽ nhánh:

1) Ở bước 1, nếu chưa có danh thiếp nào được lưu thì hiển thị khoảng trống. Use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Đã đăng nhập vào tài khoản người dùng

6. Hậu điều kiện: Không có

7. Điểm mở rộng: Không có

Use case Quản lý danh thiếp (Admin)

1. Tên use case: Quản lý danh thiếp

2. Mô tả vắn tắt: Use case này cho phép admin xem, xuất danh thiếp của tất cả người dùng.

3. Luồng sự kiện

3.1. Luồng cơ bản:

1) Use case bắt đầu khi người dùng ở trang chủ admin. Hệ thống hiển thị danh sách tất cả danh thiếp của tất cả người dùng.

2) Admin có thể xem chi tiết danh thiếp và xuất chúng. Hệ thống cập nhật tương ứng theo hành động. Use case kết thúc.

3.2. Luồng rẽ nhánh:

2) Ở bước 1, nếu chưa có danh thiếp nào được lưu thì hiển thị khoảng trống. Use case kết thúc.

4. Các yêu cầu đặc biệt: Không có

5. Tiền điều kiện: Đã đăng nhập vào tài khoản admin

6. Hậu điều kiện: Không có 7. Điểm mở rộng: Không có

4.1.2. Thiết kế cơ sở dữ liệu

Hệ thống sử dụng cơ sở dữ liệu quan hệ để lưu trữ thông tin người dùng và danh thiếp. Việc thiết kế cơ sở dữ liệu nhằm đảm bảo tính toàn vẹn dữ liệu, khả năng mở rộng, và hỗ trợ truy xuất hiệu quả cho các chức năng chính như quản lý người dùng, tải lên ảnh danh thiếp, xử lý và chỉnh sửa thông tin trích xuất.

Cơ sở dữ liệu gồm hai bảng chính: User và BusinessCard.

Bảng User lưu trữ thông tin người dùng của hệ thống, bao gồm cả người dùng thông thường và quản trị viên.

Bảng 4.1: Bảng User

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Integer	Primary Key	Mã định danh duy nhất
username	String(100)	Unique, Not null	Tên người dùng
email	String(100)	Unique, Not null	Địa chỉ email
password	String(100)	Not null	Mật khẩu đã mã hóa
is_admin	Boolean	Mặc định (false)	Đánh dấu quyền quản trị
created_at	DateTime	Mặc định là thời gian hiện tại	Thời điểm tạo tài khoản

Bảng BusinessCard lưu trữ thông tin liên quan đến từng danh thiếp được người dùng tải lên, bao gồm trạng thái xử lý và dữ liệu trích xuất.

Bảng 4.2: Bảng BusinessCard

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Integer	Primary Key	Mã định danh duy nhất danh thiếp
user_id	Integer	Foreign Key (User.Id), Not Null	Tên người dùng
filename	String(200)	Not null	Tên file hệ thống lưu trữ
original_filename	String(200)	Not null	Tên file gốc khi người dùng tải lên
status	String(20)	Mặc định là 'pending'	Trạng thái xử lý: pending, processed, failed
uploaded_at	DateTime	Mặc định là thời gian hiện tại	Thời điểm tải ảnh lên
processed_at	DateTime		Thời điểm xử lý hoàn tất
extracted_data	Text		Dữ liệu trích xuất (dạng JSON)
edited_data	Text		Dữ liệu sau khi người dùng chỉnh sửa (JSON)
logs	Text		Nhật ký quá trình xử lý (log hệ thống)

Quan hệ giữa các bảng:

- Một User có thể có nhiều nhiều BusinessCard (1-N)

4.2. Công nghệ sử dụng

Để phát triển ứng dụng demo, em đã lựa chọn các công nghệ phù hợp nhằm đảm bảo tính hiệu quả, khả năng mở rộng và trải nghiệm người dùng tốt. Các công nghệ chính được sử dụng bao gồm Flask, HTML, CSS và Jinja, được tích hợp một cách hợp lý để xây dựng một hệ thống web hoàn chỉnh. Dưới đây là mô tả chi tiết về từng công nghệ và vai trò của chúng trong ứng dụng.

4.2.3. Flask

Flask là một framework phát triển web nhẹ (micro-framework) được viết bằng ngôn ngữ lập trình Python. Framework này được chọn vì tính đơn giản, linh hoạt và khả năng tùy chỉnh cao, phù hợp với các ứng dụng web quy mô nhỏ đến trung bình. Trong ứng dụng demo, Flask đóng vai trò là nền tảng chính để xử lý các yêu cầu HTTP, định tuyến URL, và quản lý logic phía server. Các tính năng chính của Flask được khai thác bao gồm:

- Quản lý định tuyến: Flask cho phép định nghĩa các tuyến đường (routes) để xử lý các yêu cầu từ client, giúp tổ chức mã nguồn một cách rõ ràng và dễ bảo trì.
- Xử lý yêu cầu và phản hồi: Flask hỗ trợ xử lý các phương thức HTTP (GET, POST) và trả về phản hồi dưới dạng dữ liệu JSON hoặc giao diện HTML được render.
- Tích hợp với Jinja: Flask tích hợp sẵn Jinja, cho phép tạo giao diện động một cách hiệu quả.

Jinja là một template engine được tích hợp trong Flask, cho phép tạo các giao diện web động bằng cách kết hợp HTML với dữ liệu từ server. Trong ứng dụng demo, Jinja đóng vai trò quan trọng với các tính năng sau:

- Hiển thị dữ liệu động: Jinja cho phép chèn dữ liệu từ server (ví dụ: danh sách sản phẩm, thông tin người dùng) vào các tệp HTML, tạo ra giao diện linh hoạt và cá nhân hóa.
- Kế thừa template: Jinja hỗ trợ cơ chế kế thừa template, giúp tái sử dụng các thành phần giao diện chung, giảm thiểu lặp lại mã nguồn và tăng tính bảo trì.
- Xử lý logic trong template: Jinja cung cấp các câu lệnh điều kiện và vòng lặp, giúp hiển thị nội dung theo các kịch bản cụ thể.

Việc sử dụng Jinja giúp tối ưu hóa quá trình phát triển giao diện, đảm bảo tính linh hoạt và khả năng mở rộng của ứng dụng.

Việc sử dụng Flask giúp giảm độ phức tạp trong quá trình phát triển, đồng thời cung cấp khả năng mở rộng khi cần tích hợp thêm các tính năng mới.

4.2.4. HTML, CSS

HTML (HyperText Markup Language) và CSS (Cascading Style Sheets) là hai công nghệ cốt lõi để xây dựng và định dạng giao diện người dùng của ứng dụng web. Trong ứng dụng demo, HTML và CSS được sử dụng đồng bộ để tạo ra giao diện trực quan, thân thiện và đáp ứng, với các vai trò chính như sau:

- Cấu trúc giao diện (HTML): HTML được sử dụng để định nghĩa cấu trúc của các trang web, bao gồm các thành phần như biểu mẫu, bảng, và các yếu tố tương tác. HTML đảm bảo nội dung được tổ chức logic, dễ tiếp cận trên nhiều trình duyệt khác nhau.
- Thiết kế và định dạng (CSS): CSS chịu trách nhiệm định dạng giao diện, bao gồm màu sắc, bố cục, phông chữ và các hiệu ứng trực quan. CSS cũng hỗ trợ thiết kế đáp ứng (responsive design) thông qua

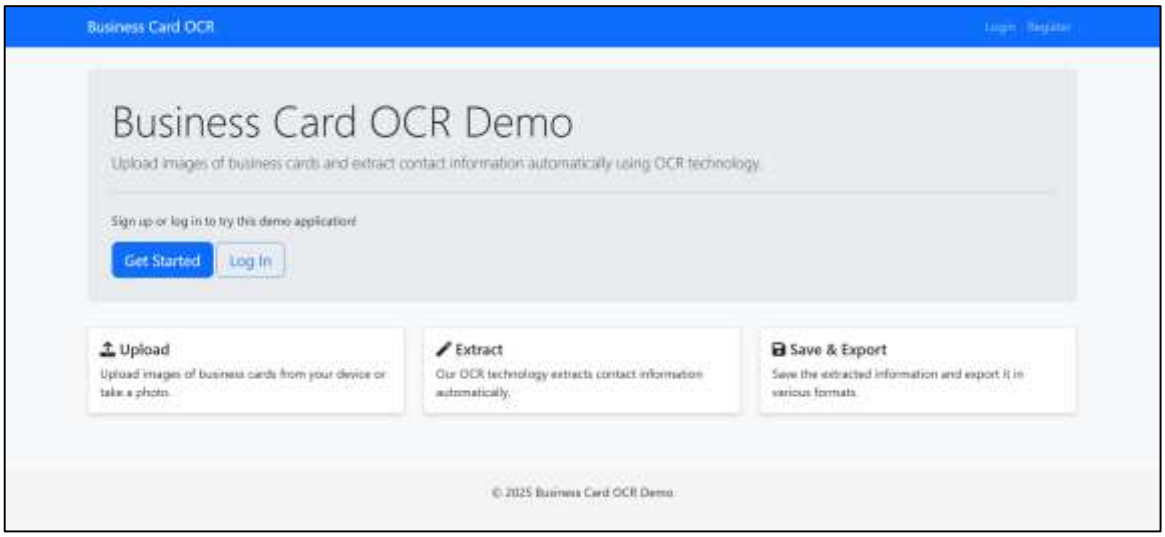
media queries, giúp giao diện hiển thị tốt trên nhiều thiết bị như máy tính, điện thoại và máy tính bảng.

- Tích hợp và tái sử dụng: HTML kết hợp với Jinja để hiển thị dữ liệu động, trong khi CSS sử dụng các tệp riêng biệt để quản lý kiểu dáng, giúp tái sử dụng và bảo trì dễ dàng.

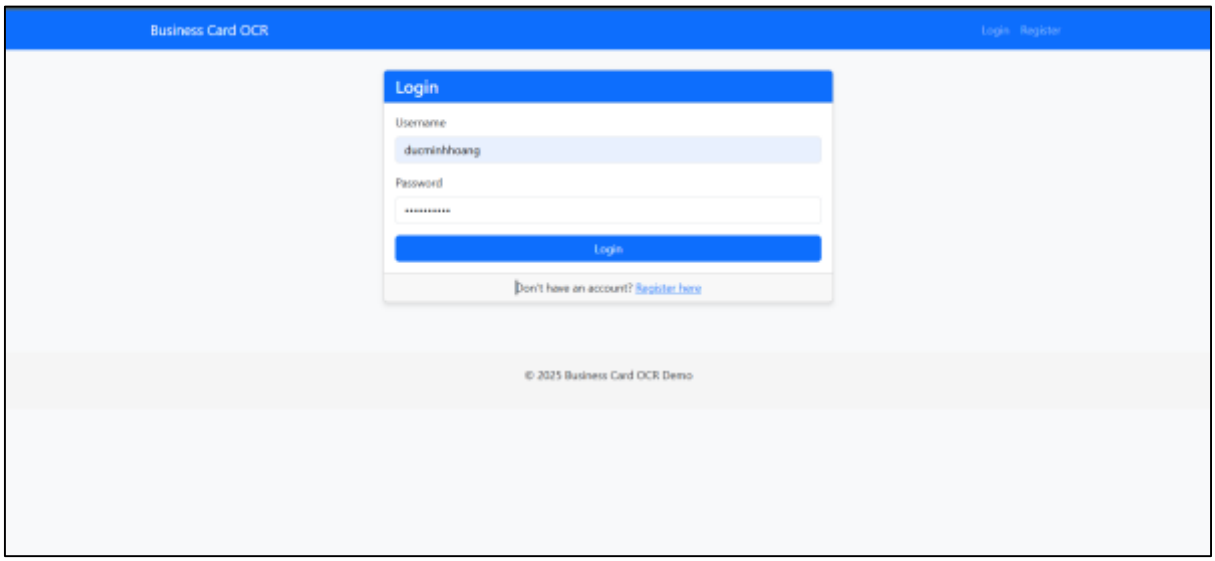
Sự kết hợp giữa HTML và CSS tạo nên một giao diện người dùng thẩm mỹ, dễ sử dụng và tương thích đa nền tảng, đồng thời giảm thời gian phát triển nhờ tách biệt giữa cấu trúc và thiết kế.

4.3. Giao diện chương trình

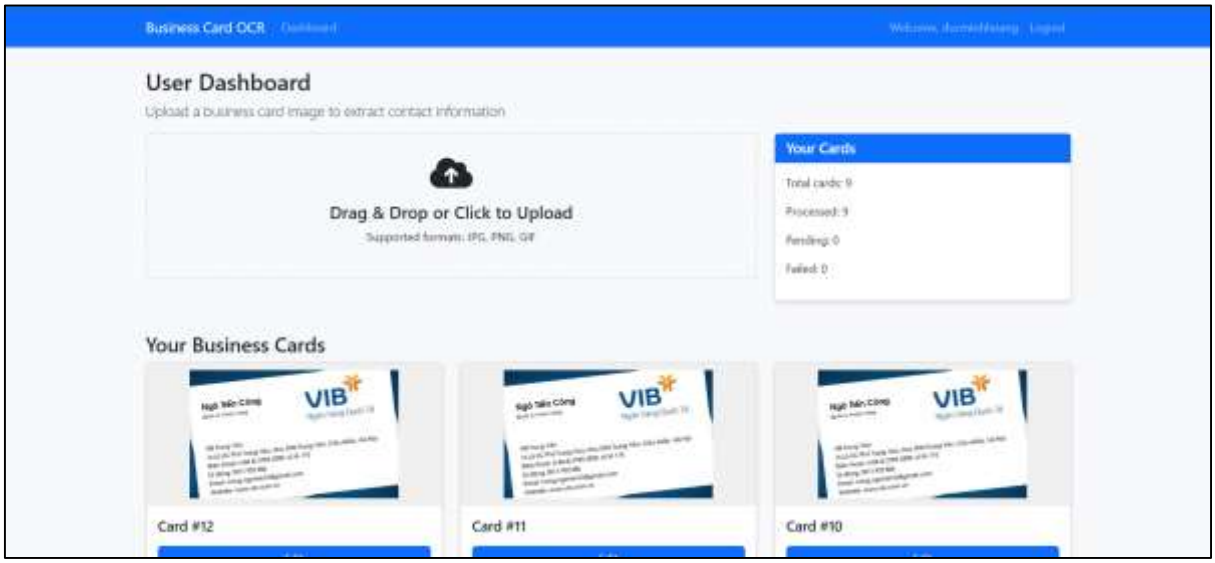
Sau quá trình thiết kế và triển khai, ứng dụng trích xuất thông tin danh thiếp đã được xây dựng hoàn chỉnh với đầy đủ các chức năng cơ bản như: đăng ký, đăng nhập, tải lên ảnh danh thiếp, trích xuất thông tin tự động, chỉnh sửa thông tin sau xử lý, xuất dữ liệu và quản lý danh thiếp. Để minh họa cụ thể cách thức hoạt động của hệ thống, phần này sẽ trình bày các màn hình giao diện tiêu biểu được chụp từ phiên bản chạy thực tế của ứng dụng. Các ảnh chụp này phản ánh trực quan luồng thao tác của người dùng cũng như kết quả hoạt động của các chức năng chính.



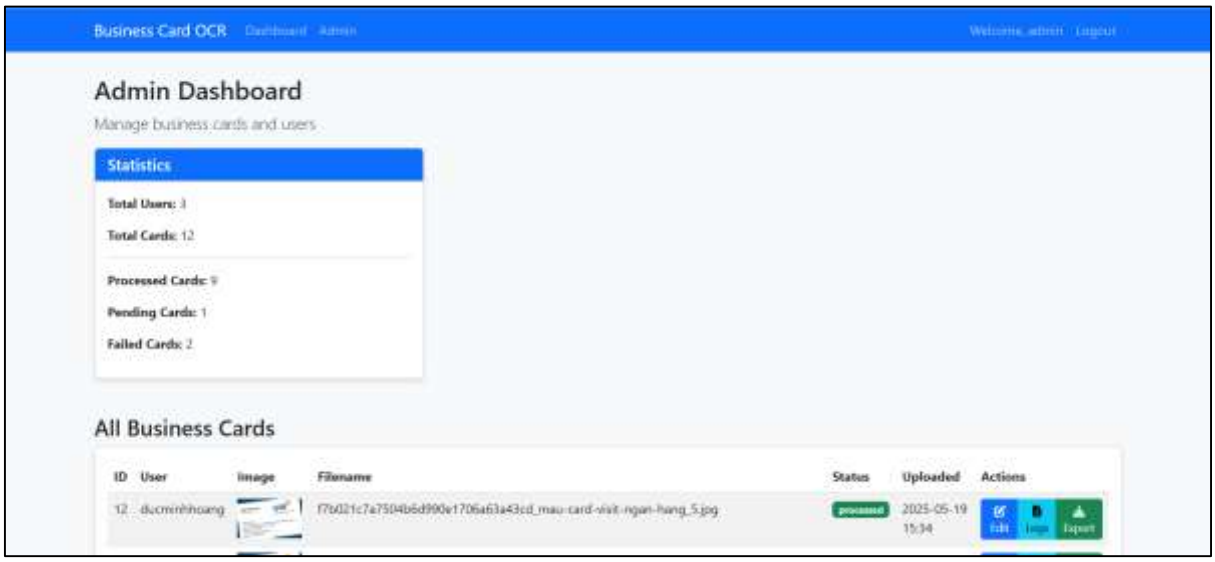
Hình 4.2: Màn hình trang chủ chưa đăng nhập



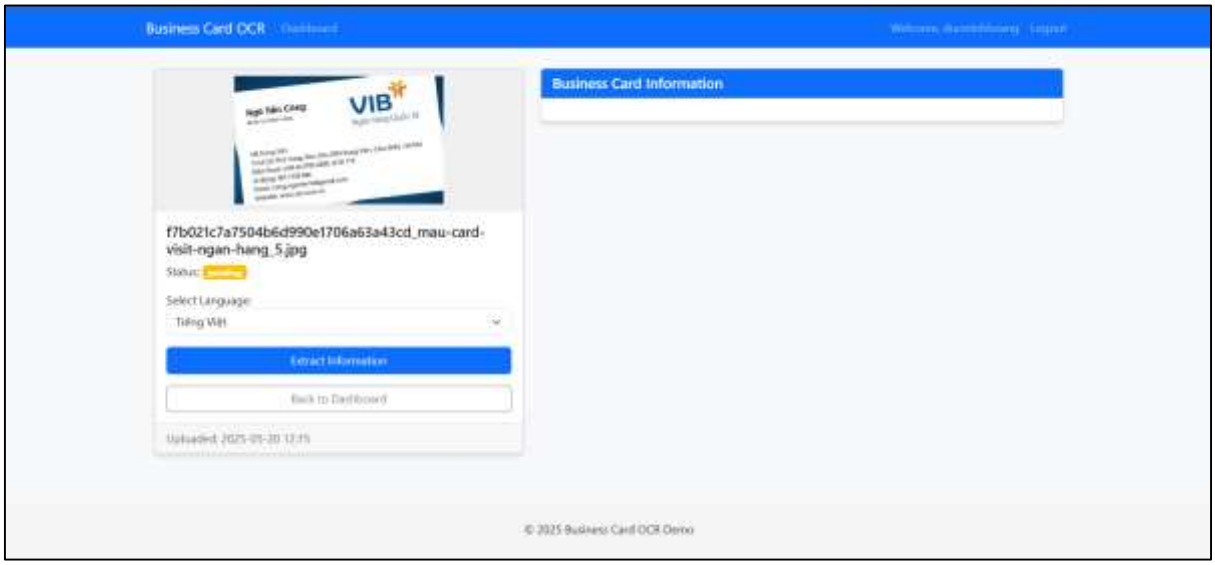
Hình 4.3: Màn hình đăng nhập



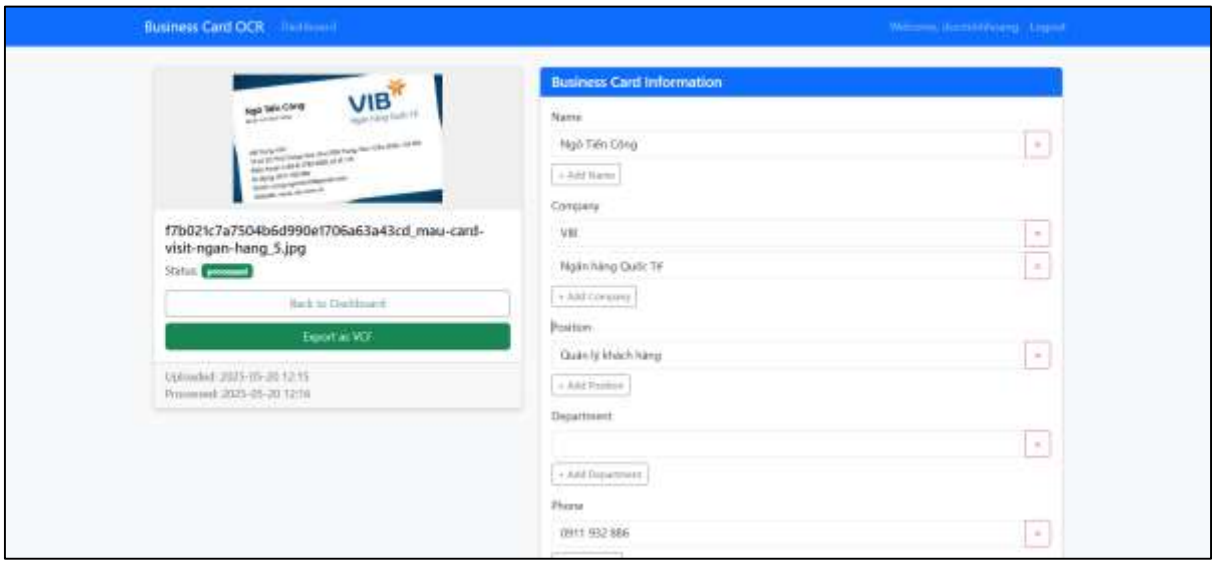
Hình 4.4: Màn hình trang chủ người dùng



Hình 4.5: Màn hình trang chủ admin



Hình 4.6: Màn hình thực hiện trích xuất thông tin danh thiếp



Hình 4.7: Màn hình hiển thị chi tiết kết quả

Qua các ảnh chụp màn hình giao diện, có thể thấy rằng ứng dụng đã đáp ứng được yêu cầu ban đầu đề ra, đảm bảo tính trực quan, thân thiện với người dùng và hỗ trợ đầy đủ quy trình từ tải ảnh danh thiếp đến trích xuất, chỉnh sửa và lưu trữ thông tin. Các chức năng hoạt động ổn định, giao diện đơn giản nhưng hiệu quả, phù hợp với người dùng không chuyên kỹ thuật. Kết quả này cho thấy ứng dụng hoàn toàn khả thi để triển khai trong thực tế hoặc phát triển thêm các tính năng nâng cao như nhận diện đa ngôn ngữ, tìm kiếm danh thiếp theo thông tin trích xuất hoặc tích hợp với hệ thống CRM.

KẾT LUẬN

Trong khuôn khổ đồ án, em đã xây dựng thành công một hệ thống trích xuất thông tin từ danh thiếp, kết hợp giữa các mô hình học sâu và ứng dụng thực tế. Hệ thống sử dụng PaddleOCR để trích xuất văn bản từ ảnh, Gemma-3-12b-it để sửa lỗi chính tả, sau đó áp dụng XLM-RoBERT được finetuning lại để gán nhãn và trích xuất đa thông tin như họ tên, email, số điện thoại, chức vụ, địa chỉ,... Kết quả thực nghiệm cho thấy XLM-RoBERTa vượt trội hơn BiLSTM-CRF và BERT nhờ khả năng xử lý tốt văn bản đa ngôn ngữ và hội tụ ổn định, phù hợp với đặc thù dữ liệu nhiễu từ ảnh danh thiếp.

Ứng dụng có những chức năng cho phép người dùng đăng ký, đăng nhập, tải ảnh danh thiếp, chỉnh sửa và xuất thông tin, đồng thời cung cấp chức năng quản lý danh thiếp cho admin. Giao diện đơn giản, dễ sử dụng, hỗ trợ đầy đủ quy trình từ tải ảnh đến lưu thông tin đã xử lý.

Tuy nhiên, hệ thống còn hạn chế ở việc chưa học được từ phản hồi người dùng, chưa xử lý tốt các ảnh chất lượng kém và chưa tích hợp các chức năng nâng cao như tìm kiếm, phân loại hay kết nối với hệ thống CRM, cũng như chưa tối ưu hóa việc sửa chính tả tốc độ chạy tốt hơn. Trong tương lai, nhóm đề xuất mở rộng dữ liệu huấn luyện, bổ sung khả năng học liên tục, cải tiến tính năng, tăng cường bảo mật và nâng cao trải nghiệm người dùng. Nhìn chung, đồ án đã chứng minh tính khả thi của bài toán cả về mặt kỹ thuật và ứng dụng thực tiễn.

TÀI LIỆU THAM KHẢO

[1]. Russell, S., & Norvig, (2020), *P. Artificial Intelligence: A Modern Approach (4th ed.)*. Pearson.

[2]. Dr. Wongthawat Liawrungrueang, (2024), *How to start understanding Machine Learning in spine research,* AO spine.

[3]. ReportLinker, (2023), *Optical Character Recognition (OCR) Systems Global Market Report 2023*, Global New Swire.

[4]. Sausalito, Calif., (2023), *2023 Cybersecurity Almanac: 100 Facts, Figures, Predictions, And Statistics, Cybersecurity Predictions,* Cybercrime Managize.

[5]. Bộ Nội vụ, (2023), *Kết quả Chuyển đổi số quốc gia năm 2023*, Cải cách hành chính.

[6]. Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, Haoshuang Wang, (2020), "PP-OCR: A Practical Ultra Lightweight OCR System", Baidu Inc, 2020.

[7]. CONNEAU, Alexis, et al, (2019), "Unsupervised cross-lingual representation learning at scale". arXiv preprint arXiv:1911.02116.

[8]. HUANG, Zhiheng; XU, Wei; YU, Kai, (2015), "Bidirectional LSTM-CRF models for sequence tagging", arXiv preprint arXiv:1508.01991.

[9]. DEVLIN, Jacob, et al. (2019), "Bert: Pre-training of deep bidirectional transformers for language understanding", In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers). p. 4171-4186.