RESEARCH ARTICLE

NHẬN DẠNG VÀ SỬA LỖI KÝ TỰ QUANG HỌC

Lê Thị Bảo Trân

Khoa Công nghệ thông tin, Trường Đại học Ngoại ngữ -Tin học TP.HCM

tranltb@huflit.edu.vn

TÓM TẮT— Hệ thống nhận dạng ký tự quang học (OCR) giúp nhận diện các ký tự nằm trong tài liệu hình ảnh. Tuy nhiên, hình ảnh kém chất lượng và những hạn chế của kỹ thuật phát hiện và làm sạch lỗi văn bản dẫn đến các kết quả văn bản không chính xác. Để nâng cao chất lượng đầu ra của văn bản, nghiên cứu đề xuất một cách tiếp cận mới về phát hiện và sửa lỗi OCR bằng cách sử dụng mô hình học máy kết hợp BERT pretrained bởi google AI và mô hình seq2seq, sau đó sử dụng thuật toán tính khoảng cách để giải quyết các vấn đề tối ưu hóa. Thông qua việc cài đặt hiệu quả các tham số thuật toán, mô hình này có thể được thực hiện với việc tạo ứng viên chất lượng cao và sửa lỗi. Và mô hình được huấn luyện trên tập dữ liệu với 1000 hình ảnh thu thập từ google, sau đó cũng xây dựng một web để thử nghiệm. Kết quả thử nghiệm cho thấy phương pháp được đề xuất vượt trội hơn các phương pháp truyền thống.

Từ khóa— Tesseract OCR, BERT, seq2seq, phát hiện, sửa lỗi.

I. GIỚI THIỆU

Với thời đại bùng nổ công nghệ 4.0 như hiện nay thì nhu cầu về chuyển đổi các tài liệu trên giấy truyền thống sang các tài liệu số trở nên vô cùng lớn. Và con người đã và đang nỗ lực để chuyển các tài liệu trên giấy thành văn bản điện tử để phục vụ cho việc xử lý văn bản bằng máy tính (ví dụ: tìm kiếm hoặc tóm tắt), lưu trữ, truy cập dễ dàng hơn. Quá trình số hóa tài liệu liên quan đến việc quét hoặc chụp ảnh tài liệu từng trang một và chuyển đổi mỗi trang thành văn bản có thể cho máy tính đọc được. Phương pháp chuyển đổi có thể dựa vào nhiều yếu tố như phương tiện, văn bản in hoặc viết tay, ngôn ngữ và nhiều yếu tố khác. Hiện nay con người có thể số hóa tài liệu giấy thông qua các phương thức như nhập văn bản thủ công, sử dụng phần mềm nhận dạng ký tự quang học (OCR) và phương pháp bán tự động.

Việc nhập văn bản thủ công là một giải pháp dễ dàng nhưng lại tốn khá nhiều chi phí và có thể gây ra các vấn đề về bảo mật do lộ tài liệu với bên thứ 3. Hơn nữa giá chung của thị trường việc thuê đánh máy một trang tài liệu có giá khoảng từ 10.000 ~ 20.000 VNĐ.

Phần mềm nhận diện ký tự quang học (OCR) là một giải pháp tất yếu giúp cá nhân/ tổ chức có thể tiết kiệm chi phí trong quá trình chuyển đổi tài liệu giấy sang tài liệu số cũng như bảo đảm về vấn đề bảo mật thông tin. Phương pháp này cung cấp tỷ lệ nhận dạng tốt và đã trở thành một trong những cách phổ biến nhất để chuyển đổi văn bản số.

Mặc dù các công cụ OCR được cải thiện liên tục về kết quả nhận diện văn bản và hoạt động tốt trên văn bản hiện đại, nhưng vẫn thiếu dữ liệu huấn luyện, chất lượng (vật lý) của tài liệu không đáp ứng được dẫn đến kết quả OCR có thể xuất hiện lỗi sai và ảnh hưởng đến kết quả đầu ra. Theo nhiều nghiên cứu về việc truy xuất thông tin và xử lý ngôn ngữ tự nhiên đã cho thấy ảnh hưởng tiêu cực khi tài liệu được số hóa sau bước OCR bỏ lỡ một số thông tin quan trọng có thể trả về những thông tin không liên quan khi người dùng truy vấn hoặc tìm kiếm. Tuy nhiên, hiện nay chưa có nhiều nghiên cứu về sửa lỗi sai của quá trình OCR.

Chính từ những lý do trên, việc xây dựng mô hình có thể phát hiện và sửa lỗi OCR là vô cùng cần thiết.

II. CÁC CÔNG TRÌNH LIÊN QUAN

A. KHÁI NIỆM NHẬN DIỆN KÍ TỰ QUANG HỌC (OCR)ERROR! REFERENCE SOURCE NOT FOUND.

OCR được viết tắt bởi cụm từ Optical Character Recognition (dịch là nhận dạng ký tự quang học). OCR được biết đến là một công cụ scan kỹ thuật số chuyên nhận dạng các ký tự, chữ viết tay, hay chữ đánh máy. Hay nói cách khác đây là ứng dụng công nghệ sử dụng trí tuệ nhân tạo chuyên dùng để đọc text ở file ảnh. Và công nghệ này chuyên dùng để truyền tải, nhập liệu dữ liệu.

Theo AWS, nhận dạng ký tự quang học (OCR) là quá trình chuyển đổi một hình ảnh văn bản thành định dạng văn bản mà máy có thể đọc được. Nói tóm lại nhận dạng kí tự quang học là một công nghệ có thể phát hiện và nhận dạng các kí tự trong các bức ảnh chuyển thành văn bản mà máy tính có thể xử lý được như file.txt , .docx, v.v... Ví dụ như khi chúng ta quét một biểu mẫu hoặc biên lai, máy tính sẽ lưu bản quét đó dưới dạng tệp hình ảnh. Chúng ta không thể sử dụng trình soạn thảo văn bản để chỉnh sửa, tìm kiếm hoặc đếm số từ trong tệp hình ảnh. Tuy nhiên, ta có thể sử dụng OCR để chuyển đổi hình ảnh thành tài liệu văn bản, trong đó phần nội dung sẽ được lưu trữ dưới dạng dữ liệu văn bản. Từ đó chúng ta có thể dễ dàng truy xuất thông tin, tìm kiếm, sửa chữa những thông tin đã lưu. Ngoài ra trong lĩnh vực trinh sát mạng BTL 86 thường xuyên phải thu thập dữ liệu trên không

gian mạng gồm nhiều dạng dữ liệu khác nhau bao gồm ảnh, công cụ OCR có thể hỗ trợ rất nhiều trong việc chuyển nội dung trong ảnh thành văn bản để tiện tóm tắt và tổng hợp.

B. CÁCH HOẠT ĐỘNG CỦA PHẦN MỀM NHẬN DIỆN KÝ TỰ QUANG HỌC OCR *ERROR! REFERENCE SOURCE NOT FOUND.*

a) Thu nhận hình ảnh

Một máy quét sẽ đọc tài liệu và chuyển đổi chúng thành dữ liệu nhị phân. Phần mềm OCR phân tích hình ảnh đã quét và phân loại vùng sáng làm nền và vùng tối làm văn bản.

b) Tiền xử lý

Trước tiên, phần mềm OCR sẽ làm sạch hình ảnh và loại bỏ các lỗi để chuẩn bị cho bước đọc. Sau đây là một số kỹ thuật làm sạch của phần mềm OCR:

- Chỉnh thẳng hoặc nghiêng nhẹ tài liệu đã quét để khắc phục lỗi về căn chỉnh trong quá trình quét.
- Khử nhiễu đốm hoặc loại bỏ mọi đốm ảnh kỹ thuật số hay làm mịn các viền của hình ảnh văn bản.
- Làm sạch đường viền khung và đường thẳng trong hình ảnh.
- Nhận dạng chữ viết cho công nghệ OCR đa ngôn ngữ

c) Nhận dạng văn bản

Hai loại thuật toán OCR hoặc quy trình phần mềm chính mà phần mềm OCR sử dụng để nhận dạng văn bản được gọi là so khớp mẫu và trích xuất đặc điểm.

d) So khớp mẫu

Cách thức hoạt động của so khớp mẫu là tách biệt một hình ảnh ký tự, được gọi là hình dạng chữ và so sánh với một hình dạng chữ tương tự được lưu trữ. Tính năng nhận dạng mẫu chỉ hoạt động hiệu quả khi hình dạng chữ được lưu trữ có phông chữ và tỷ lệ tương tự với hình dạng chữ đầu vào. Phương thức này hoạt động tốt đối với hình ảnh quét từ tài liệu được đánh máy bằng phông chữ đã biết.

e) Trích xuất đặc điểm

Trích xuất đặc điểm sẽ chia nhỏ hoặc phân tách hình dạng chữ thành các đặc điểm như nét thẳng, nét vòng khép kín, hướng nét và giao điểm nét. Sau đó, hệ thống sử dụng các đặc điểm này để tìm kết quả phù hợp nhất hoặc kết quả gần đúng nhất trong số các hình dạng chữ khác nhau được lưu trữ.

f) Hậu xử lý

Sau khi phân tích, hệ thống sẽ chuyển đổi dữ liệu văn bản được trích xuất thành tệp trên máy tính. Một số hệ thống OCR có thể tạo các tệp PDF có chú thích bao gồm cả phiên bản trước và sau của tài liệu được quét.

C. TESSERACT OCR

Tesseract là một OCR engine hàng đầu hiện nay. Công cụ này được phân phối với bản quyền mã nguồn mở Apache 2.0. Nó hỗ trợ nhận diện ký tự trên các tập tin hình ảnh và xuất ra dưới dạng kí tự thuần, html, pdf, tsv, invisible-text-only pdf. Người dùng có thể sử dụng trực tiếp hoặc lập trình viên có thể sử dụng các chức năng thông qua API.

Tesseract giả định rằng đầu vào của nó là một hình ảnh nhị phân với các vùng văn bản đa giác tùy chọn được xác định.

Bước đầu tiên trong quy trình là phân tích các thành phần kết nối trong hình ảnh và các đường viền của các thành phần này sẽ được lưu trữ. Có thể nói rằng Tesseract có lẽ là bộ máy OCR đầu tiên có khả năng xử lý các văn bản trắng trên nền đen một cách dễ dàng như vậy. Ở giai đoạn này, các đường viền được tập hợp lại thông qua việc lồng vào nhau để tạo thành các đối tượng gọi là "Blobs".

Các phần của hình ảnh (gọi là Blobs) chứa thông tin về các ký tự được phân tích thành các đường viền và được tổ chức lại thành các dòng văn bản. Các dòng văn bản và các vùng văn bản khác nhau sau đó được xem xét để xác định liệu chúng có viết với khoảng cách giữa các ký tự cố định hay tỷ lệ tỉ lệ. Sau đó, các dòng văn bản được chia thành từng từ theo cách khác nhau, tùy thuộc vào cách mà các ký tự được cách nhau. Nếu văn bản được viết với khoảng cách giữa các ký tự cố định, chúng sẽ được cắt ngay lập tức thành các ô ký tự riêng lẻ. Đối với văn bản có khoảng cách tỷ lệ, các từ sẽ được tách ra bằng cách sử dụng các khoảng trống xác định và khoảng trống không xác định.

Tiến trình nhận dạng văn bản được thực hiện dưới dạng một quá trình hai bước. Trong bước đầu tiên, Tesseract cố gắng nhận dạng từng từ một cách tuần tự. Mỗi từ mà nhận dạng tốt đủ được sử dụng để đào tạo một bộ phân loại thích nghi. Bộ phân loại thích nghi sau đó được cung cấp cơ hội để nhận dạng văn bản một cách chính xác hơn ở phần dưới của trang. Ở bước thứ hai do bộ phân loại thích nghi có thể học được thông tin hữu ích một

cách quá muộn để ảnh hưởng đến vùng đầu của trang, quá trình nhận dạng được thực hiện lần thứ hai trên toàn trang. Ở lần này, các từ mà không nhận dạng đủ tốt trong lần đầu tiên sẽ được nhận dạng lại. Cuối cùng, Tesseract giải quyết các khoảng trống không xác định (những khoảng trống không rõ ràng) và kiểm tra các giả thuyết khác nhau về chiều cao của ký tự. Điều này giúp xác định văn bản chữ nhỏ một cách chính xác.

D. LỖI OCR

Lỗi OCR là toàn bộ những từ, ký tự mà hệ thống nhận diện kí tự quang học OCR nhận diện sai so với ảnh thực tế. Theo kết quả ở bảng 1.1, Tesseract đạt được tỷ lệ lỗi ký tự dao động trong khoảng 1.31% - 2.48% trên các bộ dữ liệu khác nhau. Đối với lỗi từ, tỷ lệ sai sót có phần cao hơn, trong khoảng 3.06% - 5.13%. Nhìn chung, so với nhiều hệ thống OCR khác cùng thời điểm, kết quả này là khá ấn tượng.

Bên cạnh đó, so với phiên bản cũ năm 1995, phiên bản Tesseract mới đã có cải thiện đáng kể về độ chính xác, với tỷ lệ lỗi giảm tổng cộng 7.31% cho lỗi ký tự và 5.39% cho lỗi từ. Điều này cho thấy Tesseract vẫn đang được cải tiến và nâng cấp liên tục.

III. PHƯƠNG PHÁP ĐỀ XUẤT

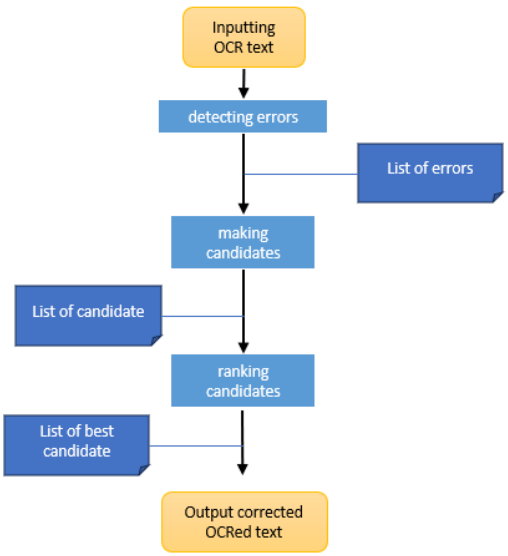
A. MỘT SỐ PHƯƠNG PHÁP SỬA LỖI KÝ TỰ QUANG HỌC

1. QUY TRÌNH XỬ LÝ KÝ TỰ QUANG HỌC

Bài toán nhận dạng và sửa lỗi ký tự quang học dựa trên ý tưởng bài toán sửa lỗi chính tả có đầu vào là từ sai và đầu ra là từ đúng thay thế cho từ sai. Tuy nhiên, bài toán sửa lỗi ký tự quang học có thể nhận diện các ký tự nằm trong các hình ảnh kém chất lượng, ký tự bị biến dạng, hoặc một số ký tự bị giống nhau do kiểu chữ. Trong khi đó, bài toán sửa lỗi chính tả thường phát hiện lỗi gõ văn bản sai do đánh nhầm vào ký tự liền kề.

Quy trình xử lý ký tự quang học gồm 3 giai đoạn:

- Preprocessing (tiền xử lý): Tiền xử lý dữ liệu hình ảnh.
- Processing (xử lý): Định vị vùng văn bản và nhận dạng văn bản để cho ra văn bản ký tự quang học.
- Postprocessing (hậu xử lý) như **Error! Reference source not found.**: Nhận diện lỗi sai ký tự quang học sau đó cho ra danh sách đánh dấu để xử lý lại nếu cần. Sửa lỗi sai dựa trên danh sách các ứng cử viên thay thế cho lỗi đó, trong đó ứng cử viên có xếp hạng cao nhất sẽ được thay thế cho ký tự lỗi **Error! Reference source not found.**.



Hình 1. *Quy trình hậu xử lý kết quả OCR*

2. MỘT SỐ CÁCH SỬA LỖI KÝ TỰ QUANG HỌC

Hiện nay, có rất nhiều phương pháp sửa lỗi chính tả hay lỗi chính tả từ các kết quả nhận dạng ký tự quang học. Tuy nhiên, có thể phân thành ba dạng chính như sau: thủ công, bán tự động và tự động hoàn toàn. Trong bài báo này, phương pháp đề xuất là phát hiện lỗi theo phương pháp bán tự động.

a) Tiếp cận sửa lỗi từng từ: nhận diện và sửa lỗi cho từng từ một cho đến hết toàn văn bản

- Sử dụng và kết hợp nhiều kết quả từ các hệ thống nhận diện ký tự quang học để tăng độ chính xác.
- Tham khảo thông tin từ vựng và ngữ pháp để cải thiện nhận diện ký tự quang học.
- Tham khảo mô hình lỗi để nhận dạng và sửa các lỗi sai thường gặp trong nhận diện ký tự quang học.

- Cải thiện độ chính xác của nhận diện ký tự quang học bằng cách sử dụng mô hình ngôn ngữ theo chủ đề.

b) Tiếp cận dựa vào ngữ cảnh Error! Reference source not found.Error! Reference source not found.
Mô hình ngôn ngữ: là một mô hình để cải thiện độ chính xác của văn bản bằng cách đánh giá lỗi sai của từ hoặc chuỗi từ, sau đó thay thế các lỗi này bằng các từ hoặc chuỗi từ được đề xuất dựa trên do lường xác suất phù hợp nhất với ngữ cảnh ngôn ngữ tự nhiên.

- Mô hình thống kê: là mô hình tính toán xác suất xuất hiện của các từ dựa trên các thống kê trên cơ sở dữ liệu lớn về ngôn ngữ.
- Mô hình dựa trên trí tuệ nhân tạo: đây là mô hình sử dụng trí tuệ nhân tạo để học cấu trúc và quy luật của ngôn ngữ tự nhiên
- Mô hình học máy: sử dụng các đặc trưng rút trích từ thuộc tính văn bản để đưa ra các dự đoán và cải thiện kết quả các ký tự quang học

Mô hình seq2seq: mô hình học máy bằng cách sử dụng mạng nơ ron để tạo ra một chuỗi đầu ra các ký tự quang học tốt hơn chuỗi đầu vào.

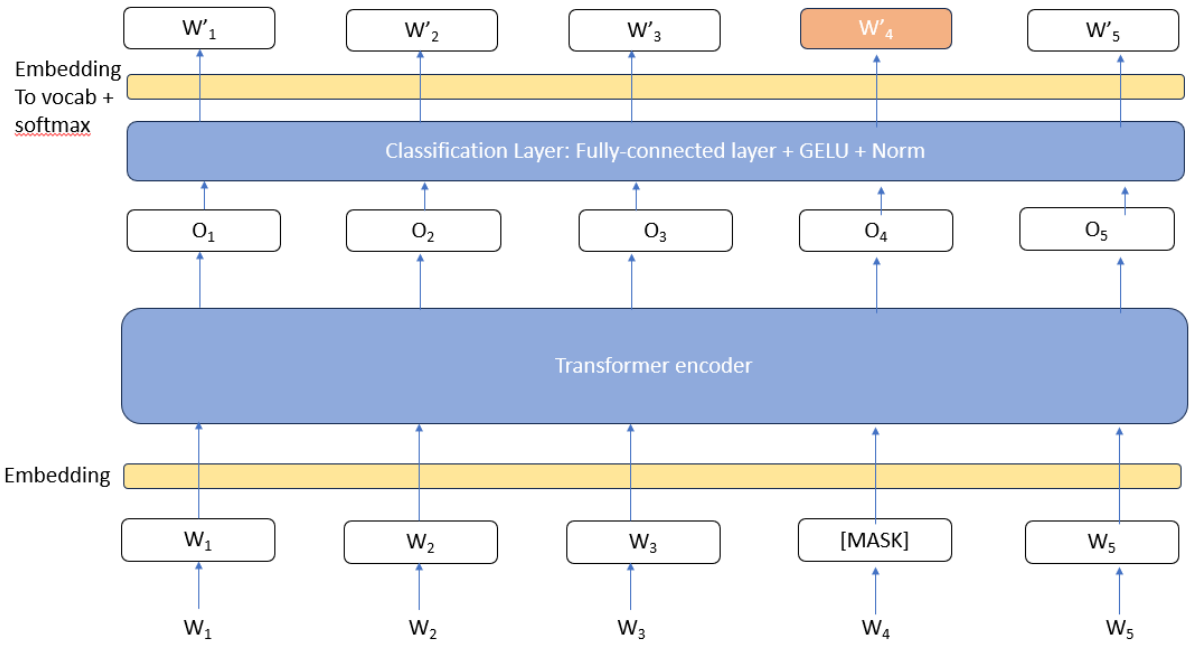
Trong bài nghiên cứu này, sẽ kết hợp mô hình seq2seq và BERT, trong đó BERT để tạo ra danh sách các kết quả có thể thay thế từ được đánh dấu lỗi, seq2seq sẽ chịu trách nhiệm tổng hợp và tìm kết quả được đánh giá tốt nhất để thay thế cho từ bị lỗi.

B. PHƯƠNG PHÁP ĐỀ XUẤT

Mô hình BERT sẽ phát hiện một từ sai trong chuỗi đầu ra của quá trình nhận dạng ký tự quang học, sau đó sẽ gán nhãn thông báo trước từ đó và ngay sau từ đó, sau đó đưa cả câu vào mô hình và lấy ra được một danh sách các kết quả có khả năng thay thế từ được đánh dấu. Mô hình seq2seq sẽ tổng hợp các kết quả và tìm cái tốt nhất để thay thế cho từ sai.

1. MÔ HÌNH BERT

BERT là viết tắt của cụm từ Bidirectional Encoder Representation from Transformer có nghĩa là mô hình biểu diễn từ theo 2 chiều ứng dụng kỹ thuật Transformer. BERT được thiết kế để huấn luyện trước các biểu diễn từ (pre-train word embedding). Điểm đặc biệt ở BERT đó là nó có thể điều hòa cân bằng bối cảnh theo cả 2 chiều trái và phải.
Cơ chế attention của Transformer sẽ truyền toàn bộ các từ trong câu văn đồng thời vào mô hình một lúc mà không cần quan tâm đến chiều của câu. Do đó Transformer được xem như là huấn luyện hai chiều (bidirectional) mặc dù trên thực tế chính xác hơn chúng ta có thể nói rằng đó là huấn luyện không chiều (non-directional). Đặc điểm này cho phép mô hình học được bối cảnh của từ dựa trên toàn bộ các từ xung quanh nó bao gồm cả từ bên trái và từ bên phải.



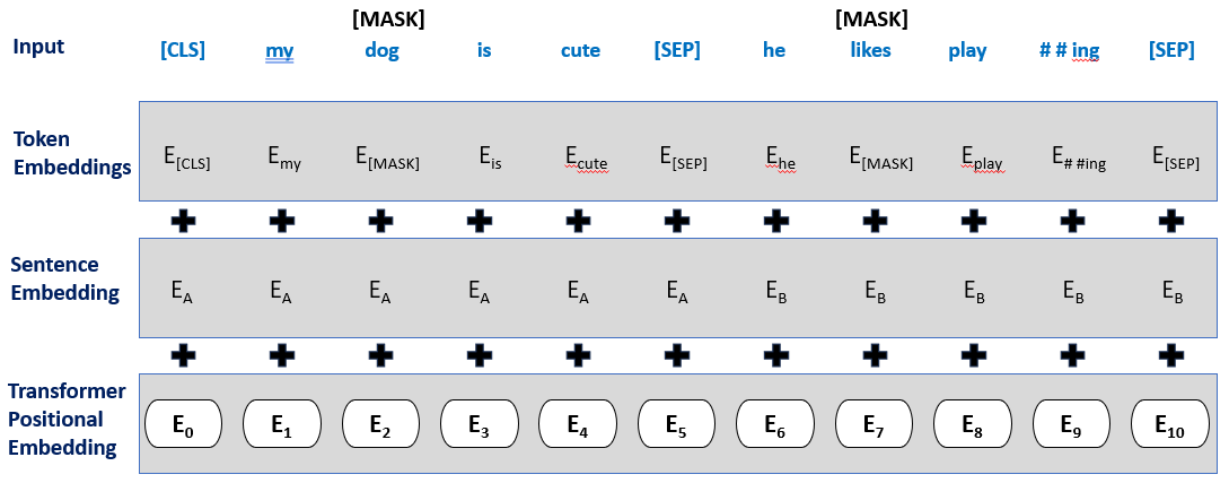
Hình 2. *Kiến trúc BERT*

BERT được coi là bước đột phá lớn trong lĩnh vực xử lý ngôn ngữ tự nhiên bởi khả năng ứng dụng của nó vào nhiều bài toán NLP khác nhau: Trả lời câu hỏi, suy luận ngôn ngữ tự nhiên,... với kết quả rất tốt.
Một trong những công thức lớn nhất của NLP là dữ liệu có vấn đề. Trên internet có hàng tá dữ liệu, nhưng những dữ liệu đó không đồng nhất; mỗi phần của nó chỉ được sử dụng cho một mục tiêu riêng biệt, khi giải quyết một bài toán cụ thể, ta cần trích ra một bộ dữ liệu thích hợp cho bài toán của mình và kết quả là ta chỉ có một lượng rất ít dữ liệu. Nhưng có một điều nghịch lý là các mô hình Deep Learning cần lượng dữ liệu rất lớn - lên hàng triệu - để có thể đạt được kết quả tốt. Có một vấn đề cần được đặt ra: làm cách nào để tận dụng nguồn dữ liệu vô cùng sẵn có để giải quyết bài toán của mình. Đó là tiền đề cho một kỹ thuật mới ra đời: Transfer Learning . Với Transfer Learning các mô hình (mô hình) "chung" tốt nhất với dữ liệu trên internet (pre-trained) được xây dựng và có thể fine-tunning cho các bài toán khác nhau. Giúp có kỹ thuật này kết quả cho các bài toán được cải thiện tốt hơn, không chỉ trong NLP mà còn trong các lĩnh vực khác như Thị giác máy tính,... và BERT là một trong những đại diện ưu tú nhất trong Transfer Learning cho NLP.
BERT sử dụng Transformer là một mô hình (attention) học mối quan hệ giữa các từ (hoặc một phần của từ) trong một văn bản nói cách khác mô hình có thêm vector đại diện cho ngữ cảnh (contex) của dữ liệu đầu vào. Transformer bao gồm 2 phần chính: Encoder và Decoder, Encoder thực hiện đọc dữ liệu đầu vào và đầu ra dự kiến của bộ giải mã. Ở đây, BERT chỉ sử dụng Encoder.
Khác với các mô hình định hướng (các mô hình chỉ đọc dữ liệu theo 1 chiều duy nhất - trái→phải, phải→trái) đọc dữ liệu theo dạng tuần tự, bộ mã hóa đọc toàn bộ dữ liệu trong một lần, công việc này thực hiện BERT có khả năng huấn luyện dữ liệu theo cả hai chiều, nhờ đó hình ảnh có thể học được ngữ cảnh (contex) của từ tốt hơn bằng cách sử dụng các từ xung quanh nó (phải & trái).

Theo hoạt động của kiến trúc BERT, đầu vào là chuỗi các w1, w2,... được biểu diễn dưới dạng vector trước khi đưa vào mạng. Đầu ra của mô hình là các vector có kích thước bằng đầu vào. Để học được ngữ cảnh của từ, BERT có 2 chiến lược đào tạo:

Gán mặt nạ: 15% token của input được thay thế bởi [MASK] như **Error! Reference source not found.** trước khi truyền vào model. Mô hình sẽ dựa trên các từ không được che và context để dự đoán giá trị của từ gốc bị che. Như vậy, BERT vẫn gồm 2 nhánh encoder giúp embedding các từ input và decoder tìm ra phân phối xác suất của các từ output. Sau khi thực hiện self- attention và feed forward ta sẽ thu được đầu ra là các vector o1, o2, ... Để tính phân phối xác suất cho output, mô hình có thêm Fully connect layer sau Transformer Encoder sử dụng softmax để tính toán phân phối xác suất. Cuối cùng ta có được vector của mỗi từ tại vị trí [MASK] là vector giảm chiều của oi sau khi qua fully connected

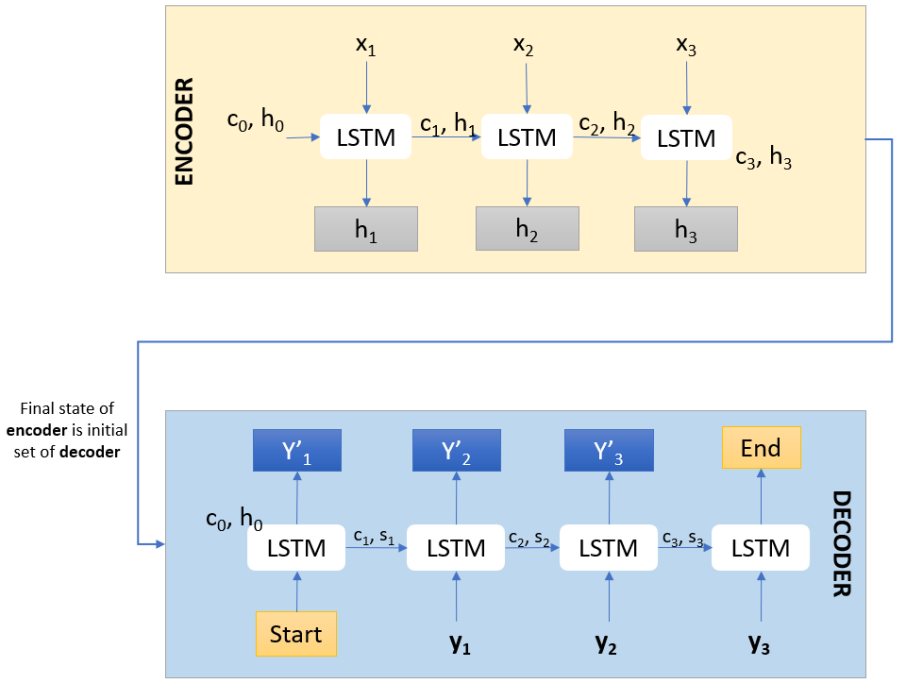
Dự đoán câu từ tiếp theo: mô hình sẽ sử dụng một cặp câu làm đầu vào dữ liệu, và dự đoán câu thứ 2 có khả năng là câu tiếp theo của câu thứ 1 hay không. Trong quá trình huấn luyện, 50% lượng dữ liệu đầu vào là cặp câu trong đó câu thứ 2 thực sự là câu tiếp theo của câu thứ 1, còn lại 50% thì câu thứ 2 được chọn ngẫu nhiên từ dữ liệu. Cụ thể, dữ liệu sẽ được xử lý như sau: khi một từ được phát hiện là sai trong output, sẽ chèn mã [CLS] vào trước từ đó và [SEP] ngay sau từ đó, mã thông báo trong từng câu được đánh dấu là A hoặc B, chèn thêm vector biểu tượng vị trí của mã thông báo trong câu. Sau đó, đưa cả câu vào mô hình và được output là một danh sách kết quả có thể thay thế được các từ được đánh dấu. Tuy nhiên, để tổng hợp và tìm kết quả được đánh giá tốt nhất có thể thay thế cho từ bị sai, mô hình seq2seq sẽ chịu trách nhiệm công việc này.



Hình 3. *Đánh dấu [MASK] cho từ*

2. MÔ HÌNH SEQ2SEQ

Mô hình seq2seq đã có kết quả rất tốt đối với bài toán sửa lỗi chính tả. Chính vì vậy, nghiên cứu này đề xuất sử dụng mô hình này với mục đích sửa lỗi sai từ kết quả ocr trainning trên bộ dữ liệu xây dựng riêng cho bài toán sửa lỗi sai OCR nhằm đạt được kết quả là với đầu vào của mô hình là một từ trong câu bị sai đưa vào mô hình sao cho kết quả đầu ra là một từ đúng để thay thế từ bị sai đã được đưa vào mô hình nhằm tạo ra chuỗi đúng và có ý nghĩa.



Hình 4. *Mô hình seq2seq*

Error! Reference source not found. mô tả kiến trúc của seq2seq gồm 2 nhánh Encoder và Decoder. Phần Encoder sẽ nhận vào một chuỗi vector, mỗi vector biểu diễn một ký tự. Các lớp LSTM trích xuất thông tin từ chuỗi đầu vào và truyền nó tới lớp tiếp theo. Trạng thái cuối cùng của Encoder sẽ được truyền cho Decoder và đóng vai trò là giá trị khởi tạo của Decoder. Các lớp LSTM ở phần Decoder cũng sẽ xử lý chuỗi và truyền từ trái qua phải. Ở mỗi bước thời gian trong Decoder, một lớp Softmax thường được áp dụng để dự đoán xác suất xuất hiện của từ tiếp theo trong chuỗi đích.

Quá trình huấn luyện mô hình sẽ được thực hiện như sau:

Encoder:

Encoder xử lý chuỗi đầu vào theo từng bước thời gian (time step t) tại các LSTM Cell. Mỗi t tương ứng với một phần tử trong chuỗi đầu vào và được thực hiện từ trái sang phải chuỗi. Tại mỗi t, LSTM Cell trong Encoder nhận đầu vào của time step t và trạng thái ẩn hiddenstate h_t từ time step t-1. LSTM Cell sau đó tính toán hiddenstate (h_t) mới và cellstate $(C-t)$ mới cho time step t. Hiddenstate (h_t) và cellstate $(C-t)$ tại time step t này được truyền tới time step t+1 để được sử dụng cho việc tính toán ở time step tiếp theo.

Internal state của LSTM Cell được khởi tạo ngẫu nhiên. Internal state của LSTM Cell tại time step t-1 được sử dụng cùng với đầu vào (input) của time step t để tính toán hidden state và cell state tại time step t, vì thế mà mô hình có thể giữ lại thông tin quan trọng. Internal state của LSTM Cell cuối cùng của Encoder sẽ chứa thông tin tổng hợp của toàn bộ chuỗi sẽ làm đầu vào cho Decoder để bắt đầu quá trình sinh ra chuỗi đầu ra.

Decoder:

Tương tự như Encoder, Decoder nhận toàn bộ chuỗi đích mà bạn muốn tạo ra. Chuỗi đích gắn tiền tố "START_" và hậu tố "_END" để chỉ ra điểm bắt đầu và kết thúc.

Internal state khởi tạo của Decoder bằng với internal state của LSTM Cell cuối cùng của Encoder. Decoder cũng xử lý chuỗi đích từng bước thời gian tại các LSTM Cell như Encoder. Từ dự đoán (predicted word) tại time step t được tạo ra từ LSTM Cell tại time step t dựa vào từ đầu vào (input) tại time step t và thông tin từ time step (t-1) như: hidden state và cell state, từ dự đoán ở time step (t-1).

Tại mỗi time step t, Decoder sẽ dự đoán một từ (hoặc ký tự) tại chuỗi đích và so sánh nó với từ trong chuỗi đích đúng. Độ sai khác giữa dự đoán và đúng là lỗi (loss). Lỗi này sau đó được sử dụng trong quá trình backpropagation để cập nhật lại các trọng số (weight) của mô hình. Mục tiêu là điều chỉnh mô hình sao cho dự đoán của nó gần giống với chuỗi đích đúng.

Predict:

Trong quá trình dự đoán thì Encoder vẫn giống như quá trình huấn luyện còn đối với Decoder quá trình diễn ra như sau: Internal State của Decoder được khởi tạo bằng Internal State cuối cùng trong Encoder. Input của Decoder luôn bắt đầu bằng "Start_". Ở mỗi Time Step, LSTM Cell dự đoán một từ, từ này và internal state được sử dụng cho Time Step tiếp theo. Khi Decoder dự đoán ra là _END, quá trình sẽ kết thúc.

3. TIỀN XỬ LÝ DỮ LIỆU

Thu thập 1000 hình ảnh chứa các văn bản tiếng Anh từ google với các từ khóa " OCR", "scanned document", "quote" như **Error! Reference source not found.**, rồi tiến hành đưa từng ảnh qua tesseract ocr để thu được các đoạn text trong ảnh và lọc ra được 50000 lỗi sai.

Hình 5. *Hình ảnh trong bộ dữ liệu*

4. GÁN NHÃN

Tiến hành gán nhãn đúng cho các từ sai như sau:

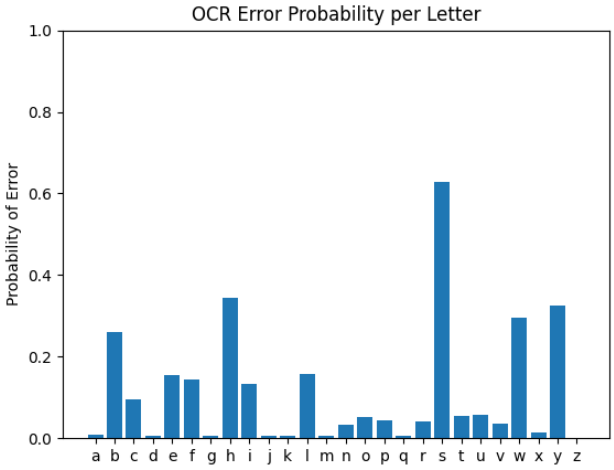
Bảng 1. *Bảng gán nhãn đúng cho từ sai*

Từ bị sai	Nhãn đúng
yung	young
dcar	dear
tlat	that
wie	wife
....	.....

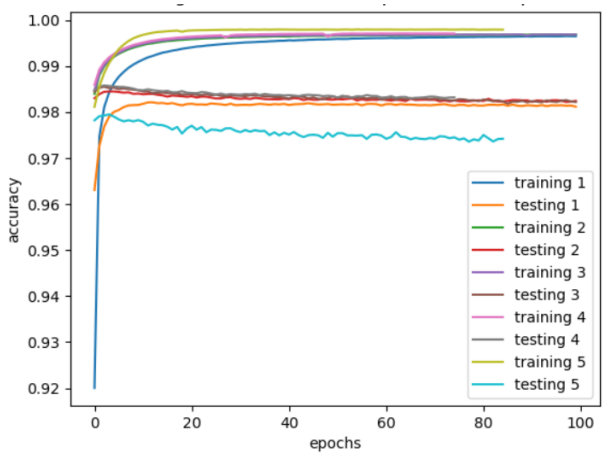
5. HUẤN LUYỆN MÔ HÌNH

a) Cấu hình: BATCH_SIZE = 64; EPOCHS = 100; LATENT_DIM = 256; NUM_SAMPLES = 50000; BREAK_CHAR = '\t' ; ENDSEQ_CHAR = '\n' ; MIN_SEQ_LENGTH = 4; MAX_SEQ_LENGTH = 45

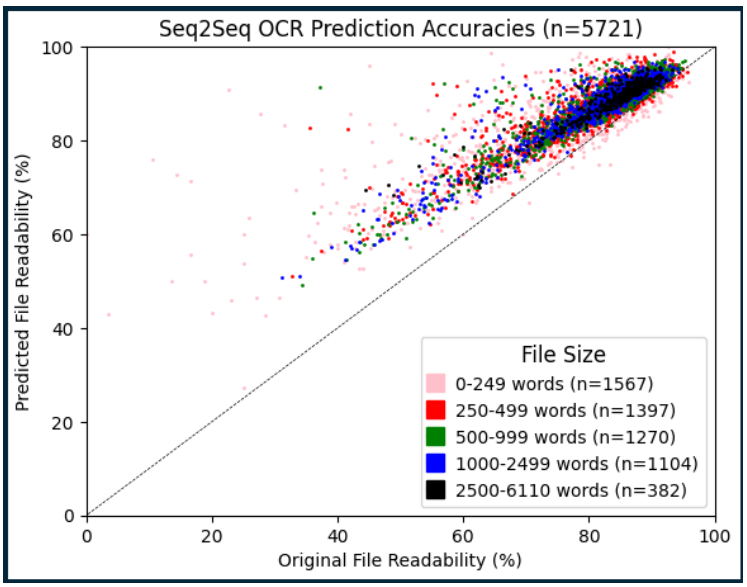
b) Kết quả huấn luyện



Hình 6. *Tần suất xảy ra lỗi trên các ký tự*



Hình 7. *Accuracy quá trình trainning Seq2Seq*

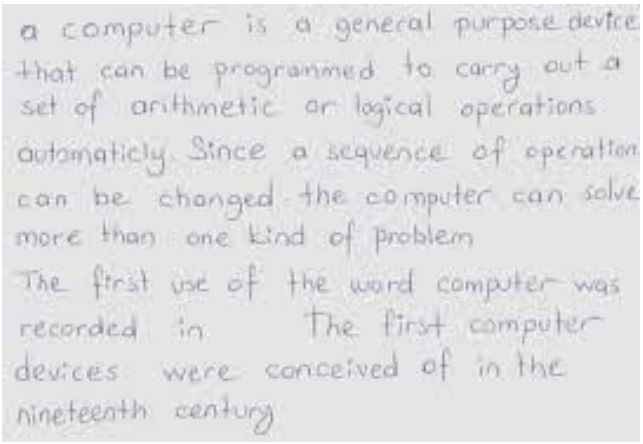


Hình 8. *Độ chính xác của mô hình Seq2Seq*

IV. KẾT QUẢ THỰC NGHIỆM

1. THỬ NGHIỆM TRÊN 1 HÌNH ẢNH:

Để đưa ra kết quả cuối cùng, 2 giai đoạn đã được thực hiện: phát hiện lỗi và sửa lỗi. Đầu tiên, đưa một hình ảnh như hình bên dưới, để thực hiện tesseract ocr



Hình 9. *Mẫu thử nghiệm cho mô hình*

Kết quả như sau:

a computer is a genecal purpose device iat cxn be Programmed $9. carry oot & Guomatiely, Stee a sequence of eperaltin can be changed the computer can salve more than one Kind of problem The first ise of Phe wor computer was recorded 4 The first compoler dewvices were conceived of in the nineteenth eenhing.

Thông qua phương pháp nhận diện lỗi bên trên đã nêu thu được kết quả phát hiện lỗi như sau:

a computer is a **genecal** purpose device **iat cxn** be programmed carry **oot guomatiely stee** a sequence of **eperaltin** can be changed the computer can salve more than one kind of problem the first **ise** of **phe wor** computer was recorded the first **compoler dewvices** were conceived of in the nineteenth **eenhing**

Phát hiện lỗi: sau khi nhận được kết quả từ tesseract ocr , tiến hành loại bỏ các ký tự không hợp lệ (không phải là số hoặc ký tự) trong văn bản bằng cách dò từng từ, nếu từ đó không nằm trong bộ dữ liệu, xây dựng sẵn bao gồm: một từ điển khoảng 120.000 từ tiếng anh, danh sách tên người khoảng 50.000 tên riêng, danh sách tên các địa điểm, bộ từ điển số đếm, bộ từ điển dạng từ (từ ở dạng số nhiều, phân từ hai, danh, động tính, cụm động từ, cụm danh từ và cụm tính từ, v.v...).

Các từ không ghép với từ trước hoặc từ sau đó để tạo thành cụm từ có ý nghĩa được cho là sai và đánh dấu '[CLS]'. Sau đó đưa danh sách các từ này vào mô hình seq2seq đã train ở phần trên và bert để dự đoán. Mô hình seq2seq cho ra kết quả một từ thay thế cho từ bị sai và mô hình BERT dựa và ngữ cảnh các từ xung quanh từ sai để đề xuất ra 50 từ thay thế. Các kết quả này được lưu vào một danh sách nhằm tiến hành đánh giá để tìm ra kết quả tốt nhất.

Sửa lỗi Error! Reference source not found.Error! Reference source not found.**:** Tuy nhiên kết quả phát sinh ra nhiều lỗi OCR do quá trình tesseract, chủ yếu do công cụ tesseract bị nhầm một số kí tự ở trong từ. Vì vậy tiến hành sử dụng khoảng cách "Levenshtein" là một phép đo khoảng cách giữa hai chuỗi hoặc hai từ ngữ. Khoảng cách Levenshtein được tính bằng cách đếm số lượng các thao tác chỉnh sửa cần thiết để biến đổi một chuỗi thành chuỗi khác. Các thao tác chỉnh sửa bao gồm:

- insert: Thêm một ký tự vào chuỗi.
- delete: Xóa một ký tự khỏi chuỗi.
- replace: Thay thế một ký tự bằng một ký tự khác.

Khoảng cách Levenshtein đo lường mức độ tương đồng giữa hai chuỗi, với giá trị nhỏ hơn đại diện cho sự tương đồng cao hơn. Từ đó tính khoảng cách giữa các ứng viên thay thế như **Error! Reference source not found.** cho từ bị sai và chọn ứng viên có độ tương đồng về kí tự với từ bị sai cao nhất.

```
seq2seq result:  ntei
['common', 'single', 'special', 'general', 'simple', 'for', 'the', 'all', 'a', '"', 'self', 'whole', 'no', 'kind', 'one', 'fixed', 'do'
', 'small', 'social', 'human', 'of', 'different', 'with', 'regular', 'natural', 'full', 'multi', 'and', 'normal', 'mixed', 'practical',
Từ gần nhất  {'general'} với  genecal
```

Hình 10. *Kết quả chọn ứng viên thay thế cho từ bị sai*

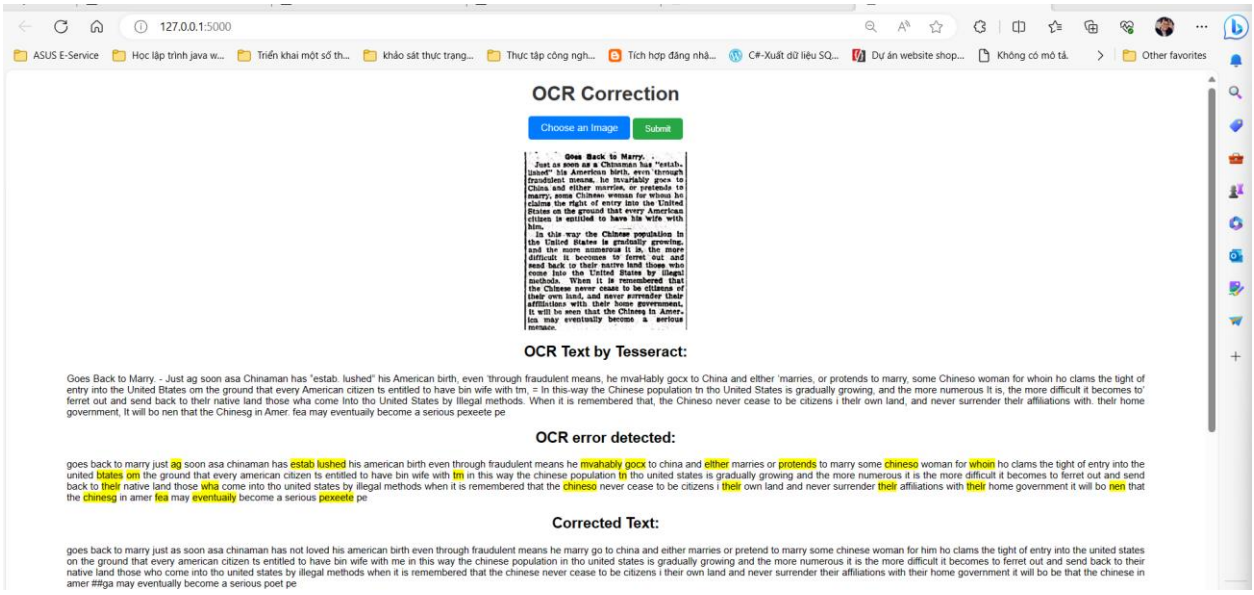
Như vậy từ “genecal” đã được thay thế bởi từ “general”, và thay hết các từ sai như sau:

Correcting:

- genecal ---> general
- iat ---> at
- cxn ---> can
- oot ---> out
- guomatiely ---> some
- stee ---> time
- eperaltin ---> operations
- ise ---> use
- phe ---> the
- wor ---> word
- compoler ---> common
- dewvices ---> devices
- eenhing ---> century

2. XÂY DỰNG GIAO DIỆN DEMO TRÊN WEB

Tiến hành xây dựng một web sử dụng framework flask của python cho phép tải lên một ảnh chứa ký tự. Sau đó sẽ tiến hành xử lý ảnh, nhận dạng văn bản trong ảnh bằng tesseract ocr, đưa vào mô hình để phát hiện và sửa lỗi. Kết quả thể hiện sửa lỗi bao gồm văn bản ban đầu, văn bản sau khi nhận diện lỗi, văn bản được sửa lỗi như **Error! Reference source not found.**.



Hình 11. *Giao diện chương trình demo*

3. KẾT QUẢ THỬ NGHIỆM VỚI DỮ LIỆU ẢNH ĐƯỢC SCANNED:

Sử dụng 25 hình ảnh có chữ để thực nghiệm, từ kết quả Bảng 2 cho thấy tỷ lệ sửa đúng của mô hình khá cao.

Bảng 2. *Kết quả thử nghiệm sửa lỗi với bộ ảnh tài liệu được scanned*

STT	Ảnh	Lỗi	Lỗi phát hiện được	Lỗi được sửa đúng	Tỷ lệ sửa đúng
1	1.jpg	13	12	10	76.92%
2	2.jpg	2	2	1	50.00%
3	3.jpg	5	4	3	60.00%
4	4.jpg	10	9	7	70.00%
5	5.jpg	7	6	5	71.43%
6	6.jpg	9	8	6	66.67%

7	7.jpg	15	14	11	78.57%
8	8.jpg	2	3	0	0.00%
9	9.jpg	6	5	4	80.00%
10	10.jpg	11	10	9	90.00%
11	11.jpg	8	7	6	85.71%
12	12.jpg	3	3	2	66.67%
13	13.jpg	4	4	3	75.00%
14	14.jpg	12	12	10	83.33%
15	15.jpg	6	6	5	83.33%
16	16.jpg	9	8	7	87.50%
17	17.jpg	5	5	4	80.00%
18	18.jpg	7	6	5	83.33%
19	19.jpg	8	8	7	87.50%
20	20.jpg	10	10	9	90.00%
21	21.jpg	4	4	3	75.00%
22	22.jpg	6	6	5	83.33%
23	23.jpg	7	7	6	85.71%
24	24.jpg	9	8	7	87.50%
25	25.jpg	5	5	4	80.00%
Tổng cộng		154	141	119	78.27%

V. KẾT LUẬN

Mô hình đã được xây dựng đơn giản bằng cách kết hợp seq2seq và BERT để đưa ra các dự đoán từ với độ chính xác tương đối, từ đó phát triển một chương trình nhằm mục đích giảm tải cho việc soát lỗi chính tả trong quá trình số hóa tài liệu từ dạnh ảnh sang dạng text nhằm tiết kiệm thời gian và chi phí để phân tích dữ liệu. Tuy nhiên, do hạn chế về việc thu thập dữ liệu, nên tập dữ liệu chưa đủ lớn để mô hình đạt độ chính xác cao. Khi kết quả tesseract ocr gặp lỗi từ bị dính liền, mô hình chưa sửa được lỗi. Hy vọng sẽ cải tiến mô hình và thu thập thêm để mở rộng tập dữ liệu để giải quyết vấn đề này trong tương lai.

VI. TÀI LIỆU THAM KHẢO

[1] Haithem Afli, Loïc Barrault, and Holger Schwenk (2016), OCR error correction using statistical machine translation, Int. J. Comput. Ling. Appl, Vol. 7, No. 1, pp. 175-191.

[2] Haithem Afli, Zhengwei Qiu, Andy Way, và Páraic Sheridan (2016), Using SMT for OCR error correction of historical texts, Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC'16), pp. 962-966.

[3] Mayce Al Azawi và Thomas M. Breuel (2014), Context-dependent confusions rules for building error model using weighted finite state transducers for OCR post-processing, Proceedings of the 2014 11th IAPR International Workshop on Document Analysis Systems IEEE, pp. 116-120.

[4] Chantal Amrhein và Simon Clematide (2018), Supervised OCR error detection and correction using statistical and neural machine translation methods, J. Lang. Technol. Comput. Ling, Vol. 33, No. 1, pp. 49-76.

[5] Richard C. Angell, George E. Freund, và Peter Willett (1983), Automatic spelling correction using a trigram similarity measure, Inf. Process. Manage, Vol. 19, No. 4, pp. 255-261.

[6] Axel Jean-Caurant, Nouredine Tamani, Vincent Courboulay, và Jean-Christophe Burie (2017), Lexicographical-based order for post-OCR correction of named entities, Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR'17) IEEE, Vol. 1, pp. 1192-1197.

[7] Mayce Al Azawi, Marcus Liwicki, và Thomas M. Breuel (2015), Combination of multiple aligned recognition outputs using WFST and LSTM, Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR'15), pp. 31-35.
[8] Lloyd Allison và Trevor I. Dix (1986), A bit-string longest-common-subsequence algorithm, Inform. Process. Lett, Vol. 23, No. 5, pp. 305-310.

OPTICAL CHARACTER ERROR RECOGNITION AND CORRECTION

Le Thi Bao Tran

ABSTRACT— The Optical character recognition (OCR) system helps identify characters contained in image documents. However, poor-quality images and limitations of text error detection and cleaning methods lead to inaccurate results. To improve the output quality of text, in this paper, I propose a new approach to detecting and correcting OCR errors using a machine learning model that combines BERT and seq2seq, and then uses distance calculation algorithm to solve optimization problems. Through the effective setting of algorithm parameters, my model can be implemented with high-quality candidate generation and error correction. We trained the model on a dataset with 1000 images collected from Google, then also built a website for testing. Experimental results show that the proposed method is more outstanding than traditional methods.

Keywords—Tesseract OCR, BERT, seq2seq, detection, correction.

Lê Thị Bảo Trân – Giảng viên khoa Công nghệ thông tin – Huflit.
Lĩnh vực nghiên cứu: các phương pháp học máy cho nhận dạng, ứng dụng nhận dạng tài liệu vào các phần mềm quản lý.