ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

BÁO CÁO
ĐỒ ÁN CHUYÊN NGÀNH

XÂY DỰNG HỆ THỐNG SỐ HÓA VÀ QUẢN LÝ VĂN BẢN

Ngành: Khoa học Máy Tính

HỘI ĐỒNG: Hội đồng 8
GVHD1: PGS.TS Trần Minh Quang
GVHD2: ThS. Bùi Tiến Đức
TKHĐ: ThS. Nguyễn Thanh Tùng
—o0o—
SVTH1: Lê Phước Đạt (2011060)
SVTH2: Nguyễn Nhật Hạ (2011158)

TP. HỒ CHÍ MINH, 01/2024

Xác nhận của giảng viên hướng dẫn

Báo cáo Đồ án Chuyên ngành - đề tài "Xây dựng hệ thống số hóa và quản lý văn bản" do hai sinh viên Lê Phước Đạt và Nguyễn Nhật Hạ thực hiện, được xác nhận bởi Giảng viên hướng dẫn Phó giáo sư - Tiến sĩ Trần Minh Quang theo yêu cầu của Khoa Khoa học và Kỹ thuật Máy tính.

....., Ngày.....Tháng.....Năm.....

Giảng viên hướng dẫn

(Ký và ghi rõ họ tên)

Lời cam đoan

Chúng tôi xin cam đoan Đồ án Chuyên ngành này là công trình nghiên cứu của riêng chúng tôi dưới sự hướng dẫn của Phó giáo sư - Tiến sĩ Trần Minh Quang và Thạc sĩ Bùi Tiến Đức. Các nguồn tham khảo, tài liệu và số liệu được sử dụng cho quá trình phân tích, nhận xét, đánh giá do chính chúng tôi thu thập từ nhiều nguồn khác nhau và đã được trích dẫn rõ ràng, đầy đủ trong phần tài liệu tham khảo để đảm bảo quyền lợi của các tác giả và nguồn thông tin. Bên cạnh đó, chúng tôi đã tham khảo và sử dụng các nhận xét, đánh giá và số liệu từ các tác giả và cơ quan tổ chức khác. Tất cả các công trình đó đều được chú thích nguồn gốc một cách đúng đắn, đảm bảo tính minh bạch và tránh vi phạm quyền tác giả. Nếu phát hiện có bất kì sự gian lận hoặc vi phạm nào, chúng tôi xin hoàn toàn chịu trách nhiệm, trường Đại học Bách Khoa Thành phố Hồ Chí Minh sẽ không liên quan đến những vi phạm tác quyền, bản quyền do chúng tôi gây ra trong quá trình thực hiện Đồ án này. Chúng tôi đã tuân thủ các nguyên tắc cũng như quy định về nghiên cứu khoa học, đạo đức và tác quyền.

Chúng tôi đã tự mình thu thập, kiểm tra và xử lý dữ liệu từ các nguồn khác nhau để đảm bảo tính chính xác và đáng tin cậy của thông tin được trình bày. Tất cả những nhận xét, ý kiến và kết luận còn lại trong Đồ án này đều là trung thực và phản ánh đúng những nỗ lực nghiêm túc của chúng tôi.

Sinh viên thực hiện đề tài

Lê Phước Đạt - Nguyễn Nhật Hạ

Lời cảm ơn

Trong quá trình thực hiện Đồ án Chuyên ngành, chúng tôi đã nhận được sự hỗ trợ từ rất nhiều bên. Đầu tiên và quan trọng nhất là hai giảng viên hướng dẫn trực tiếp của nhóm, Phó giáo sư - Tiến sĩ Trần Minh Quang và Thạc Sĩ Bùi Tiến Đức. Hai thầy là người đã theo dõi sát sao, tận tâm hướng dẫn và hỗ trợ chúng tôi trong quá trình thực hiện Đồ án này. Nhờ những nhận xét và lời khuyên của hai thầy đã giúp chúng tôi đi đúng hướng và kịp thời khắc phục những lỗi sai. Ngoài ra, chúng tôi còn nhận được rất nhiều lời khuyên có ích và sự hỗ trợ của bạn bè và người thân.

Chúng tôi xin chân thành cảm ơn những sự hỗ trợ, định hướng và giúp đỡ của hai thầy, cùng với tất cả các giảng viên, bạn bè và gia đình đã đóng góp ý kiến, cho lời khuyên, động viên và giúp đỡ chúng tôi trong suốt quá trình thực hiện đề tài này. Những đóng góp của họ là nguồn động lực to lớn để chúng tôi vượt qua những thách thức và khó khăn, chúng đều đã góp phần quan trọng vào quá trình và kết quả của Đồ án này.

Trong quá trình thực hiện dự án, chúng tôi phải đối mặt với một số hạn chế và giới hạn nhất định, cùng với ràng buộc về tài nguyên và thời gian, những điều đó đã ảnh hưởng đến việc tiến hành các thí nghiệm và phân tích chi tiết hơn. Tuy nhiên, chúng tôi đã cố gắng tối đa để đảm bảo tính chính xác và đáng tin cậy của kết quả đạt được.

Sinh viên thực hiện đề tài

Lê Phước Đạt - Nguyễn Nhật Hạ

Tóm tắt

Trong quản lý tài liệu hiện nay, phương pháp lưu trữ truyền thống hay lưu trữ vật lý đã dần trở nên lỗi thời và tồn đọng rất nhiều điểm yếu. Các hệ thống lưu trữ và quản lý văn bản hiện nay cũng còn những hạn chế nhất định. Qua đó, cho thấy rõ tính cấp thiết của đề tài "Xây dựng hệ thống số hóa và quản lý văn bản".

Thông qua đồ án này, chúng tôi đã nghiên cứu các giải pháp khả thi để từ đó rút ra ưu nhược điểm của chúng và đề xuất giải pháp để xây dựng được một hệ thống có thể giảm bớt các hạn chế đang tồn tại, từ đó giúp cho người dùng có trải nghiệm tốt hơn.

Cụ thể, hệ thống sẽ sử dụng cấu trúc dữ liệu kết hợp giữa khung được hệ thống cung cấp sẵn theo miền và hỗ trợ người dùng tự tạo cấu trúc phù hợp với nhu cầu. Bên cạnh đó, công nghệ Langchain và Chat Generative Pre-training Transformer (Chat GPT) cũng sẽ được sử dụng để hỗ trợ trong việc phân loại văn bản. Ngoài ra, hệ thống còn sử dụng Tesseract để làm công cụ nhận diện ký tự quang học (OCR) và Elasticsearch để lưu trữ và tìm kiếm văn bản.

Bố cục của đề tài

- **Chương 1: Giới thiệu**
  Đặt vấn đề và nêu tính cấp thiết của đề tài. Trình bày mục tiêu, phạm vi nghiên cứu và kế hoạch thực hiện đề tài.
- **Chương 2: Phân tích các hệ thống tương tự**
  Phân tích và nêu những ưu nhược điểm của các hệ thống phổ biến có mặt trên thị trường hiện nay.
- **Chương 3: Cơ sở lý thuyết**
  Trình bày cơ sở lý thuyết xoay quanh đề tài.
- **Chương 4: Phân tích và giải pháp**
  Phân tích các phương pháp và đưa ra giải pháp áp dụng cho hệ thống.
- **Chương 5: Phát triển hệ thống**
  Thiết kế và nêu những công nghệ sẽ sử dụng để phát triển hệ thống.
- **Chương 6: Tổng kết**
  Nhắc lại tính cấp thiết của đề tài, đánh giá kết quả và phương hướng phát triển trong tương lai.

Mục lục

Danh sách hình vẽ

Danh sách bảng

1 Giới thiệu

1.1 Đặt vấn đề

Trong quản lý tài liệu hiện nay, phương pháp lưu trữ truyền thống hay lưu trữ vật lý đã dần trở nên lỗi thời và tồn đọng rất nhiều điểm yếu như: Cần nhiều diện tích không gian để lưu trữ, khó khăn trong vấn đề bảo quản, dễ gây ra thất thoát và hư hỏng, cần nhiều nhân lực để quản lý, gây khó khăn trong việc phân loại và tìm kiếm tài liệu,... Vì vậy, ở thời điểm hiện tại cùng với sự phát triển của khoa học công nghệ, nhu cầu số hóa và quản lý tài liệu số đang ngày càng được lan rộng và chú trọng. Do đó, trên thị trường cũng đã xuất hiện khá nhiều các hệ thống hỗ trợ số hóa và quản lý tài liệu từ quy mô nhỏ cho cá nhân cho đến quy mô lớn cho doanh nghiệp như Google Drive, Microsoft SharePoint hay DocuWare. Tuy nhiên, chúng vẫn có những hạn chế nhất định trong việc phân loại, quản lý và tìm kiếm tài liệu.

Những vấn đề mà một hệ thống số hóa và quản lý tài liệu (Electronic Document Management System, viết tắt là EDMS) cần giải quyết chính là những điểm yếu mà phương pháp lưu trữ tài liệu bằng giấy và những hạn chế của các hệ thống đang có mang lại. Cụ thể như sau[1]:

- Thiếu không gian lưu trữ: Việc lưu trữ tài liệu giấy có thể chiếm rất nhiều không gian. Ngoài ra, khi lưu trữ số cũng đòi hỏi có lượng bộ nhớ đủ lớn. Lượng tài liệu sẽ ngày càng tăng theo thời gian đòi hỏi phải luôn sẵn sàng mở rộng không gian lưu trữ khi cần thiết.

- Tài liệu dễ bị hư hỏng: Khi lưu trữ tài liệu giấy sẽ luôn tồn tại những rủi ro khách quan như ẩm thấp, thiên tai, hỏa hoạn hoặc những lý do chủ quan như bị đánh cắp gây hư hại trầm trọng dẫn đến mất các tài liệu quan trọng, hơn hết nếu như không có bản sao lưu nào thì một khi các tài liệu bị hư hại không thể nào khôi phục lại dữ liệu nữa. Bên cạnh tài liệu giấy, các hệ thống lưu trữ số cũng có thể gây lỗi dữ liệu khi tải lên hệ thống mà không thông báo cho người dùng, điều này sẽ làm ảnh hưởng lớn tới việc lưu trữ.

- Vận chuyển tài liệu: Với số lượng tài liệu giấy khổng lồ thì việc vận chuyển tài liệu tới một địa điểm khác là một vấn đề khó giải quyết, tốn thời gian và tài nguyên, ngoài ra trong quá trình vận chuyển có thể gây ra thất thoát tài liệu.

- Bảo mật: Tài liệu giấy là một trong những rủi ro bảo mật thông tin lớn nhất vì dễ bị mất, xử lý sai, bị hỏng và dễ bị truy cập trái phép. Điều này khiến cho các thông tin nhạy cảm của doanh nghiệp dễ gặp rủi ro nghiêm trọng.

- Phân loại và tìm kiếm tài liệu: Với lượng tài liệu giấy khổng lồ thì việc tìm kiếm một tài liệu nào đó dù đã có đủ hết thông tin cũng là một việc vô cùng khó khăn

và tốn nhiều thời gian, còn nếu không có đủ lượng thông tin cần thiết về tài liệu thì việc tìm kiếm gần như là không thể hoặc sẽ cần rất nhiều thời gian và tài nguyên con người dù cho hệ thống lưu trữ đã được cấu trúc phân loại để hỗ trợ tìm kiếm.

- Hạn chế trong làm việc: Khi hợp tác giữa người với người trong môi trường lớn, nếu sử dụng tài liệu giấy thì phải in đủ số bản sao để phát cho từng người và các bản sao đó tồn tại một cách riêng biệt, vậy nên khi có sự thay đổi trên tài liệu thì mọi người sẽ rất khó để theo dõi cũng như cập nhật kịp thời, gây ảnh hưởng tới công việc và thời gian làm việc.

1.2 Mục tiêu của đề tài

Qua thực trạng nêu trên, chúng tôi nhận thấy nhu cầu cấp thiết của xã hội về một hệ thống số hòa và quản lý tài liệu có thể khắc phục được điểm yếu của việc lưu trữ truyền thống và các hệ thống hiện đang có mặt trên thị trường. Do đó, trong đồ án này chúng tôi sẽ nghiên cứu và đưa ra các giải pháp để xây dựng nên một "Hệ thống số hóa và quản lý tài liệu" có thể khắc phục hoặc giảm bớt được những hạn chế đó. Mục tiêu cụ thể của đề tài như sau:

- Tìm kiếm và đưa ra giải pháp xây dựng cấu trúc lưu trữ tài liệu sao cho có thể phân loại rõ ràng các tài liệu mà không phụ thuộc quá nhiều vào người dùng.
- Tìm kiếm và đưa ra giải pháp lưu trữ tài liệu sao cho tốn ít dung lượng nhất có thể để có thể tối đa hóa khối lượng lưu trữ.
- Tìm kiếm và đưa ra giải pháp lưu trữ để cải thiện chức năng tìm kiếm tài liệu, tạo điều kiện tìm kiếm nhanh chóng và đầy đủ cho người dùng.

1.3 Phạm vi nghiên cứu của đề tài

Chúng tôi hướng đến phát triển một hệ thống lưu trữ và quản lý tài liệu có thể sử dụng cho cả cá nhân hay cơ quan doanh nghiệp có quy mô vừa và nhỏ. Do đó, hệ thống này có thể hỗ trợ cả đơn và đa người dùng.

Về phạm vi công nghệ, đề tài sẽ tập trung nghiên cứu các hệ thống lưu trữ đang được sử dụng rộng rãi ở các doanh nghiệp, các phương pháp liên quan tới lưu trữ và tìm kiếm tài liệu đã và đang được nghiên cứu trong thời gian gần đây, các hệ thống đám mây cung cấp không gian lưu trữ, các công nghệ nhận dạng ký tự quang học (Optical Character Recognition, viết tắt là OCR).

Về phạm vi đối tượng, đề tài sẽ tập trung vào hai nhóm tài liệu và người dùng Tiếng Việt và Tiếng Anh, trong đó, nhóm tài liệu Tiếng Việt sẽ được chúng tôi chú trọng và tập trung phát triển hơn.

1.4 Kế hoạch thực hiện đề tài

Chúng tôi đã thực hiện kế hoạch theo các bước sau để hoàn thành mục tiêu đã đề ra:

- **Bước 1:** Nghiên cứu và phân tích các hệ thống tương tự đang có mặt trên thị trường. Từ đó đưa ra những điểm mạnh và yếu của các hệ thống trên.
- **Bước 2:** Nghiên cứu và phân tích các giải pháp xây dựng cấu trúc và tìm kiếm dữ liệu cho dạng tài liệu văn bản gần đây. Từ đó, tiến hành so sánh và rút ra được ưu nhược điểm của các phương pháp đó, để có thể chọn được phương pháp tối ưu nhất cho hệ thống.
- **Bước 3:** Nghiên cứu và phân tích các giải pháp lưu trữ hiện nay. Từ đó, có thể so sánh và chọn phương pháp tối ưu nhất về công nghệ và kinh tế cho hệ thống.
- **Bước 4:** Qua các nghiên cứu và lựa chọn giải pháp phù hợp đã nêu trên, tiến hành liên kết các phần và thiết kế một hệ thống theo mục tiêu đã đề ra.
- **Bước 5:** Nghiên cứu và xác định những công nghệ sẽ áp dụng để xây dựng hệ thống.

2 Phân tích các hệ thống tương tự

2.1 Google Drive

Google Drive là dịch vụ lưu trữ và đồng bộ hóa tập tin được tạo bởi Google giúp người dùng có thể lưu trữ tập tin trên đám mây, chia sẻ tập tin, và chỉnh sửa tài liệu, văn bản, bảng tính, và bài thuyết trình với cộng tác viên[2]. Bên cạnh đó, công cụ lưu trữ dữ liệu trực tuyến (online) này cũng cho phép người dùng có thể chỉnh sửa trực tiếp một vài tài liệu như bảng tính (Google Sheets), tài liệu văn bản (Google Docs), biểu mẫu (Google Forms), các bảng thuyết trình (Google Slides).

Một số ưu điểm nổi bật của Google Drive có thể kể đến như sau:

- Tất cả đều trên đám mây (cloud): Tất cả các văn bản có thể được truy cập từ các thiết bị có kết nối mạng, điều này tạo điều kiện cho làm việc từ xa, người dùng không cần thiết phải tải (download) bất kỳ ứng dụng nào về máy. Mặc dù trên cloud nhưng những thay đổi được tự động lưu dưới thời gian thực.
- Kiểm soát phiên bản: Google Drive hỗ trợ lưu trữ các phiên bản của văn bản giúp việc cộng tác trên quy mô lớn dễ dàng thực hiện hơn. Người dùng có thể chuyển lại các phiên bản trước của văn bản.
- Chức năng tìm kiếm: Google hỗ trợ khả năng tìm kiếm mạnh mẽ, đặc biệt ở Google Drive. Người dùng có thể nhanh chóng tìm kiếm và truy xuất văn bản bằng các từ khóa, tên văn bản và thậm chí cả nội dung của văn bản.

Tuy nhiên Google Drive vẫn còn một số nhược điểm cần được khắc phục, như:

- Khả năng bảo mật: Vì tất cả dữ liệu của Google được lưu trên cloud, nên một số doanh nghiệp đặc biệt là các doanh nghiệp làm việc với dữ liệu nhạy cảm sẽ cần khả năng bảo mật cao hơn cũng như có thể toàn quyền kiểm soát dữ liệu, lúc này các giải pháp tại chỗ (on-premises) sẽ tối ưu hơn.
- Vấn đề quản lý quyền: Về cơ bản, Google Drive còn cung cấp khá ít quyền và đối với một số tổ chức lớn yêu cầu nhiều vai trò truy cập hơn, sử dụng Google Drive có thể sẽ mang lại một số bất tiện về vấn đề chia sẻ và quản lý các quyền.
- Hạn chế trong việc quản lý văn bản: Google Drive chưa hỗ trợ một số tính năng quản lý văn bản như một DMS (Document Management System) chuyên dụng, chẳng hạn như quản lý siêu dữ liệu (metadata) và tạo các luồng làm việc tự động.
- Hạn chế trong lưu trữ dữ liệu: Cấu trúc thư mục trong Drive được thực hiện theo hình thức phân cấp và không được kiểm tra trùng lặp, vì vậy cùng một văn bản có thể xuất hiện tại nhiều vị trí gây ra dư thừa dữ liệu.

2.2 Microsoft SharePoint

Microsoft SharePoint là nền tảng giúp cho doanh nghiệp có thể cộng tác làm việc trên nền tảng web. Được Microsoft cho ra đời lần đầu vào năm 2001 với mục đích chủ yếu là để lưu trữ và quản lý mọi tài liệu của doanh nghiệp, nhưng SharePoint ngày càng được phát triển để sử dụng một cách linh hoạt trong việc giao tiếp và nâng cao hiệu quả cộng tác (collaboration)[3].

Sử dụng SharePoint, người dùng có thể tạo các trang web nội bộ, hoạt động giống như bất kỳ một trang web nào khác, dành cho đội nhóm hoặc quy mô toàn công ty với tính bảo mật cực kỳ cao. Dựa vào các trang web này, các thành viên trong tổ chức có thể truy cập, chia sẻ và chỉnh sửa các tài liệu một cách dễ dàng.

Microsoft SharePoint có một số điểm mạnh tiêu biểu như sau:

- Khả năng quản lý văn bản mạnh mẽ: SharePoint cung cấp nhiều tính năng nổi bật như quản lý phiên bản của văn bản, checkin/checkout, quản lý metadata, và khả năng quản lý văn bản thông qua văn bản (file) và thư mục (folder).
- Tính tích hợp: SharePoint được tích hợp với các ứng dụng của Microsoft Office giúp việc tạo, chỉnh sửa và cộng tác của các tài liệu Word, Excel và PowerPoint trở nên dễ dàng hơn. Ngoài ra, SharePoint có thể tích hợp với nhiều hệ thống và dịch vụ của bên thứ ba cũng như bộ sản phẩm của riêng Microsoft, cung cấp một giải pháp toàn diện cho việc quản lý tài liệu.
- Chức năng tìm kiếm: SharePoint cho phép người dùng tìm tài liệu nhanh chóng bằng cách sử dụng từ khóa, metadata và thậm chí tìm kiếm theo văn bản trong tài liệu.
- Kiểm soát truy cập: SharePoint cung cấp quyền và kiểm soát truy cập chi tiết, cho phép quản trị viên xác định ai có thể xem, chỉnh sửa và xóa tài liệu.

Mặc dù là một nền tảng cộng tác mạnh mẽ, Sharepoint vẫn bọc lộ một số điểm yếu như sau:

- Phức tạp trong sử dụng: Việc thiết lập và cấu hình SharePoint có thể tương đối phức tạp với các công ty không có nguồn lực IT chuyên dụng. Việc điều chỉnh và quản lý SharePoint yêu cầu chuyên môn về kỹ thuật.
- Đào tạo nhân sự: Người dùng, đặc biệt là những người mới sử dụng SharePoint, có thể gặp khó khăn khi sử dụng đầy đủ các tính năng của nó.
- Vấn đề về hiệu suất: Các thư viện SharePoint lớn với nhiều tài liệu đôi khi có thể gặp sự cố về hiệu suất, dẫn đến thời gian tải và kết quả tìm kiếm chậm hơn.

- Giá cả: Sử dụng SharePoint yêu cầu người dùng đăng ký Microsoft 365, và có thể phát sinh chi phí cho các dịch vụ như: Lưu trữ, tùy chỉnh theo doanh nghiệp và các chức năng nâng cao khác. Ngoài ra, phiên bản on-premises còn yêu cầu đầu tư đáng kể về cơ sở hạ tầng.

- Hạn chế trong lưu trữ dữ liệu: Tương tự như Drive, cấu trúc thư mục của Sharepoint được thực hiện theo hình thức phân cấp và không được kiểm tra trùng lặp, vì vậy cùng một văn bản có thể xuất hiện tại nhiều vị trí gây ra dư thừa dữ liệu.

2.3 DocuWare

DocuWare là công ty chuyên cung cấp phần mềm quản lý tài liệu và quy trình làm việc cho các doanh nghiệp trên toàn cầu. DocuWare ở đây được hiểu như Document Management Software, thông qua tên công ty cũng thấy được sản phẩm tiên phong của công ty chính là phần mềm giúp quản lý và số hóa tài liệu DocuWare. DocuWare giúp tự động hóa các tác vụ thủ công, giảm sự phụ thuộc vào tệp giấy và cung cấp quyền truy cập an toàn vào thông tin và tệp mọi lúc, mọi nơi. Sử dụng giải pháp quản lý tài liệu trên đám mây và phần mềm tự động hóa quy trình làm việc, DocuWare cho phép số hóa, bảo mật và làm việc với các tài liệu kinh doanh, sau đó tối ưu hóa các quy trình hỗ trợ hoạt động cốt lõi của doanh nghiệp. Với 35 năm phát triển, DocuWare cung cấp một hệ thống quản lý tài liệu ổn định, an toàn và hữu dụng, do đó, dần được khách hàng tin tưởng và sử dụng rộng rãi trên thị trường.[4]

DocuWare tập chung vào các cơ chế giúp bảo vệ toàn vẹn tài liệu khi một hệ thống bị tác động bởi nhiều người dùng. Cụ thể, DocuWare có những điểm mạnh tiêu biểu như sau:

- Có nhiều gói dịch vụ với nhiều mức giá khác nhau phù hợp với nhiều nhu cầu sử dụng.

Gói	DocuWare Cloud 4	DocuWare Cloud 15	DocuWare Cloud 40	DocuWare Cloud 100
Người dùng tối đa	4	15	40	100
Dung lượng tối đa	20 GB	50 GB	500 GB	1,000 GB
Giá sử dụng (trên tháng)	$300	$1,500	Liên hệ	Liên hệ

Bảng 2.3.1: Bảng giá các gói dịch vụ của DocuWare [5, 6]

- Không có chi phí cài đặt: Vì sử dụng đám mây để lưu trữ tài liệu, nên các hoạt động với hệ thống sẽ sử dụng hoàn toàn trực tuyến. DocuWare không yêu cầu người dùng phải cài đặt bất cứ gì, chỉ cần có thiết bị có thể kết nối vào Internet.

- Sử dụng cơ chế thời gian thực hiệu quả giữa các tài khoản: Vì sử dụng cơ chế này, nên việc sai lệch và hiểu lầm giữa các người dùng được giảm bớt, giúp cho công việc được trôi chảy hơn.

- Hỗ trợ phân quyền cho quản trị viên (admin) và người dùng thông thường (user): Quản trị viên có khu vực để lưu trữ tài liệu riêng, các người dùng thường khác sẽ không có quyền được truy cập vào các tài liệu đó.

- Có một thư mục lưu trữ riêng và bảo mật nhất tên "contract" chỉ có quản trị viên mới được truy cập, thuận lợi cho các doanh nghiệp, tổ chức cần lưu trữ những tài liệu mật như hợp đồng và các giấy tờ quan trọng.

- Khi tải lên tài liệu từ máy hoặc sử dụng máy scan, nếu quá trình tải lên không gặp vấn đề gì, thì tài liệu sẽ hiển thị trên một khu vực tên "tray" thay vì tài lên thẳng hệ thống chung. Qua đó người dùng có thể tiền chỉnh sửa, ghi chú lại hoặc xóa tài liệu nếu cần trước khi lưu trữ (store) lên hệ thống chung. Sau khi đã lưu lên hệ thống thì người dùng sẽ không được tùy ý tác động lên tài liệu nữa. Nếu muốn tác động lên bất cứ tài liệu nào trên hệ thống cần phải gửi các yêu cầu (request) cho người có trách nhiệm để được cấp phép. Riêng quyền xóa tài liệu ra khỏi hệ thống, chỉ có quản trị viên mới được thực hiện quyền này.

- Hỗ trợ tìm kiếm thông qua nội dung tập tin hoặc vị trí folder giúp cho việc tìm kiếm trở nên dễ dàng hơn.

- DocuWare hỗ trợ sử dụng công nghệ OCR để tìm kiếm toàn văn bản (full-text search) thông qua nội dung tài liệu. Đặc biệt, ngoài Tiếng Anh, DocuWare hỗ trợ công nghệ này cho rất nhiều ngôn ngữ trong đó có Tiếng Việt.

Bên cạnh những điểm mạnh, DocuWare cũng tồn tại những điểm yếu như sau:

- Hỗ trợ ứng dụng máy tính để phục vụ việc quét (scan) và tải lên nhiều tài liệu cùng lúc nhưng ứng dụng sử dụng không được mượt và có hiện tượng giật.

- Các tính năng chỉnh sửa và quét phải dùng ứng dụng máy tính để truy cập vào máy quét hoặc các phần mềm chỉnh sửa được cài đặt trên máy tính chứ không truy cập được trực tiếp từ trình duyệt (browser).

- Hạn chế trong tìm kiếm toàn văn bản: Công nghệ OCR không hoạt động tốt với các phông chữ (font) lạ và nền (background) phức tạp. Ngoài ra, dù công nghệ tìm kiếm toàn văn bản cho ra kết quả khá chính xác nhưng chỉ hỗ trợ dạng tìm kiếm theo từng từ một (word-by-word) nên sẽ làm hạn chế và đôi khi kết quả trả ra sẽ không đúng theo mong muốn của người dùng.

2.4 Nhận xét và đánh giá

Qua phân tích về ba hệ thống quản lý được sử dụng rộng rãi trên, nhìn chung các hệ thống đều thực hiện tốt việc lưu trữ và tìm kiếm căn bản nhưng bên cạnh đó vẫn tồn tại những điểm yếu cần khắc phục.

2.4.1 Điểm mạnh

Các EDMS áp dụng hiện nay đều khắc phục được đa số các điểm yếu của phương pháp lưu trữ truyền thống mang lại cụ thể như:

- Không gian lưu trữ lớn và dễ dàng mở rộng: Các hệ thống đều cung cấp những gói có mức dung lượng khác nhau, tùy vào nhu cầu sử dụng mà có thể lựa chọn mức dung lượng phù hợp.

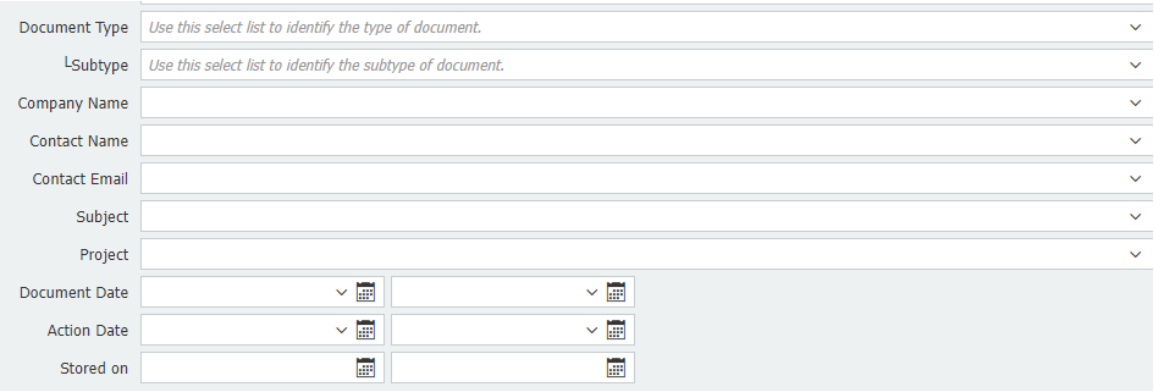
- Không gian lưu trữ trên đám mây: Toàn bộ tài liệu sẽ được lưu trữ trên đám mây của hệ thống, do đó, người dùng không phải lo về vấn đề phần cứng không đủ đáp ứng. Ngoài ra, việc lưu trữ trên đám mây còn hỗ trợ người dùng có thể tiếp cận từ nhiều nơi khác nhau mà không còn cản trở về mặt địa lý và hỗ trợ tiếp cận từ đa người dùng (multiple people) hay còn gọi là hệ thống quản lý tài liệu mở (Public Document Management System, viết tắt là Public DMS) giúp liên kết nhiều người dùng với nhau.

- Tài liệu được cập nhật trên đám mây thời gian thực (real-time) hoặc gần thời gian thực (near real-time): Giúp cho việc hoạt động của Public DMS trở nên dễ dàng và thuận tiện. Mọi người dùng đều được cập nhật tài liệu kịp thời và hạn chế ảnh hưởng tới tiến độ và chất lượng công việc.

- Các phiên bản tải lên được kiểm soát chặt chẽ: Các mục dữ liệu khi được tạo hoặc cập nhật đều được cập nhật phiên bản, hỗ trợ sao lưu lại các phiên bản cũ để người dùng có thể khôi phục nếu cần.

- Hỗ trợ phân quyền: Các EDMS hiện nay đa số đều là Public DMS do đó thông thường các hệ thống đều hỗ trợ phân quyền hoạt động. Người quản trị cao nhất của hệ thống (Super Administrator, viết tắt là super admin) có thể dễ dàng chặn một người dùng truy cập vào một tài liệu hoặc một nhóm tài liệu và nhóm tài liệu đó sẽ bị ẩn đi trên giao diện của người dùng đó. Việc này không làm ảnh hưởng tới cấu trúc của toàn bộ dữ liệu và trải nghiệm của người dùng.

- Tìm kiếm: Hỗ trợ tìm kiếm từ nhiều nguồn như thông qua vị trí tên file, tên văn bản, các thuộc tính liên quan tới mục dữ liệu, nội dung văn bản. Người dùng có thể tìm kiếm thủ công bằng cách tìm kiếm qua các thư mục, sử dụng các thông tin về thư mục muốn tìm để truy cập chính xác vị trí của dữ liệu muốn tìm, giảm bớt thời gian tìm kiếm. Hoặc sử dụng các chức năng hỗ trợ tìm kiếm có sẵn của hệ thống. Tương tự như việc tự tìm kiếm thủ công thông qua các thư mục, người dùng có thể nhập thông tin về thư mục muốn truy xuất vào bảng để tiến hành tìm kiếm. Chức năng này giúp giảm thời gian tìm kiếm và hỗ trợ truy xuất dữ liệu nếu người dùng không nắm đầy đủ thông tin hoặc có thông tin không rõ ràng về dữ liệu muốn tìm kiếm. Ngoài ra, một số hệ thống còn hỗ trợ tìm kiếm toàn văn bản (full-text search), nhằm tìm kiếm thông qua nội dung của văn bản lưu trữ.



Hình 2.4.1: Chức năng tìm kiếm của DocuWare

2.4.2 Điểm yếu

Ngoài những điểm mạnh đã khắc phục được các điểm yếu của phương pháp lưu trữ vật lý đã nêu trên, các EDMS vẫn còn hạn chế ở một số mặt. Các hệ thống hiện nay đều đòi hỏi người dùng phải tự cấu trúc dữ liệu, các cơ chế hỗ trợ xây dựng đều khá đơn giản và trực quan. Tuy nhiên, khi có một nguồn dữ liệu khổng lồ thì việc có thể cấu trúc tài liệu cho phù hợp lại trở thành việc khó khăn.

- Cứng nhắc, thiếu tính linh hoạt: Khi lượng tài liệu tăng cao và cấu trúc trở nên đồ sộ, các cấu trúc có thể trở nên lỗi thời và không còn đáp ứng được nhu cầu của người dùng thì việc sắp xếp lại các file và thay đổi cả cấu trúc sẽ trở nên cực kì khó khăn. Sự cứng nhắc này sẽ ảnh hưởng đến trải nghiệm của người dùng.

- Trùng lặp, chồng chéo và dư thừa dữ liệu: Việc tồn tại các thư mục chỉ chứa một thư mục con, hoặc tồn tại những thư mục con, tập tin có nội dung giống nhau trong các thư mục cha khác nhau thường xuyên xảy ra, điều này gây trùng lặp, chồng chéo thông tin gây lãng phí tài nguyên lưu trữ một cách không cần thiết, không tạo ra được sự nhất quán của cấu trúc và còn làm ảnh hưởng đến quá trình truy xuất dữ liệu.

- Xây dựng cấu trúc mang tính chủ quan của người dùng: Mỗi người đều có cách suy nghĩ và nhận định khác nhau về một vấn đề nào đó, vì vậy, việc xây dựng cấu trúc dữ liệu phụ thuộc hoàn toàn vào yếu tố con người sẽ gây nên sai lệch, hiểu lầm và khó khăn trong việc tìm kiếm cho người dùng khác. Vấn đề này nhận thấy rõ nhất ở các Public DMS, thông thường để giải quyết, các tổ chức phải ra quy định sử dụng chặt chẽ cho mọi người dùng và dẫn tới điểm yếu *Cứng nhắc, thiếu tính linh hoạt* đã nêu trên.

- Hạn chế trong việc thể hiện ngữ cảnh: Ngữ cảnh chủ yếu được thể hiện thông qua vị trí, vậy nên nếu không có quy ước về đặt tên hay có thêm các thông tin về tài liệu và thư mục được lưu trữ bên ngoài thì sẽ tạo nên sự hạn chế và làm ảnh hưởng đến quá trình tìm kiếm.

- Giao diện không thân thiện với người dùng: Các EDMS phổ biến đều mắc một điểm yếu là giao diện tương đối khó dùng và cần một khoảng thời gian để làm quen, đặc biệt đối với những người không am hiểu quá nhiều về máy tính và thao tác với các ứng dụng. Việc thể hiện cấu trúc thư mục lên giao diện thường được vẽ theo dạng cây nhưng chủ yếu nhìn khá rối (đặc biệt với các cấu trúc lớn) và gây cản trở trong việc tìm kiếm thủ công.

- Không hỗ trợ trong việc xây dựng một cấu trúc mới: Đối với nhiều người dùng phổ thông hoặc dùng với quy mô nhỏ, việc phải xây dựng một cấu trúc quản lý lớn từ một trang trắng sẽ tương đối khó khăn. Các hệ thống thường không hướng dẫn người dùng nên cấu trúc dữ liệu như thế nào cho hợp lý và về lâu dài sẽ dễ dẫn tới điểm yếu *Trùng lặp, chồng chéo và dư thừa thông tin* như đã nói ở trên.

- Hạn chế trong tìm kiếm: Mặc dù hỗ trợ tìm kiếm thủ công thông qua cấu trúc thư mục, tìm kiếm thông qua thông tin căn bản của dữ liệu và tìm kiếm toàn văn bản nhưng các hệ thống vẫn gặp một số vấn đề như phụ thuộc quá nhiều vào người tạo dựng cấu trúc khiến cho việc tìm kiếm trở nên khó khăn, hay tìm kiếm toàn văn bản chỉ hỗ trợ tìm theo từng từ một (word-by-word) nên sẽ cho ra kết quả tìm kiếm không đúng như kì vọng. Ngoài ra, các hệ thống đều không hỗ trợ tìm kiếm gần nghĩa, hoặc theo nhóm phù hợp với thông tin cung cấp, đòi hỏi người dùng phải sử dụng thông tin chính xác của tài liệu muốn tìm kiếm, điều này làm hạn chế trải nghiệm của người dùng.

Bên cạnh đó, việc Tiếng Việt là ngôn ngữ đặc biệt (ngôn ngữ có dấu) khiến cho các hệ thống này còn có thêm những điểm yếu khác.

- Hạn chế khi sử dụng các ngôn ngữ đặc biệt: Đây là một điểm yếu xuất hiện ở các EDMS hướng tới người dùng sử dụng các ngôn ngữ đặc biệt như ngôn ngữ có dấu (tiếng Việt), ngôn ngữ tượng hình (tiếng Trung),...

  - Với điểm yếu *Hạn chế trong việc thể hiện ngữ cảnh* đã có sẵn thì lại càng khó hơn khi sử dụng những ngôn ngữ này. Việc chỉ thể hiện ngữ cảnh thông qua vị trí của mục dữ liệu khiến cho việc đặt tên cho một folder trở nên rất quan trọng. Các EDMS đều không đặt ra quy tắc đặt tên cho các dữ liệu hay các thư mục trong cấu trúc nhằm giúp cho người dùng có thể tự do thiết kế cấu trúc nhưng lại vô hình chung dẫn tới hạn chế trong việc thể hiện ngữ cảnh.
  - Với các ngôn ngữ có nhiều ý nghĩa khác nhau còn làm ảnh hưởng và có thể gây nên điểm yếu *Trùng lặp, chồng chéo và dư thừa thông tin* khi tạo thêm các mục thông tin trống một cách không cần thiết. Bên cạnh đó khi sử dụng các ngôn ngữ này đôi lúc sẽ gây nên tình trạng không truy cập được thông tin vì các hệ thống không hiểu và không xử lý được các ngôn ngữ này gây ra lỗi.
  - Để khắc phục điểm yếu này, các công ty ở Việt Nam thường sẽ phải tự quy định về cách đặt tên tập tin và thư mục theo quy tắc chung để tránh làm hỏng

toàn bộ cấu trúc dữ liệu, phần lớn các công ty đều quy định sử dụng tiếng Anh, tiếng Việt không dấu hoặc sử dụng bộ quy tắc đặt tên theo ký hiệu đặc biệt của công ty để đặt tên cho các tập tin hay thư mục. Điều này làm giảm trải nghiệm người dùng và việc phụ thuộc vào yếu tố con người khiến cho rủi ro trong việc cấu trúc tăng cao, dễ làm hỏng toàn bộ cấu trúc tổng thể.

- Hạn chế chức năng tìm kiếm: Việc sử dụng ngôn ngữ có dấu như Tiếng Việt để cấu trúc dữ liệu làm hạn chế quá trình tìm kiếm và trích xuất văn bản.
  - Việc cấu trúc dữ liệu có thể gặp nhiều vấn đề khi sử dụng ngôn ngữ có dấu làm cho việc tìm kiếm thủ công trở nên khó khăn và cũng ảnh hưởng lớn tới các chức năng tìm kiếm của hệ thống, điều này đặc biệt nghiêm trọng khi cấu trúc bị phá hủy bới hệ thống không giải mã được các ngôn ngữ này gây ra lỗi trong quá trình lưu trữ.
  - Nhiều EDMS không hỗ trợ tìm kiếm có dấu và phải tìm kiếm không dấu khiến cho số lượng tìm kiếm ra bị dư thừa và chất lượng tìm kiếm không được đảm bảo.

3 Cơ sở lý thuyết

3.1 Cấu trúc thư mục

3.1.1 Cấu trúc phân cấp (Hierarchical)

Cấu phúc phân cấp sử dụng cấu trúc cây để sắp xếp các mục thông tin. Một lá (node) trong cây biểu thị một mục thông tin (ở đây là một file) hoặc một tập hợp các mục (một thư mục tệp). Các hệ thống sử dụng cấu trúc này cho phép người dùng tạo các thư mục và thư mục con để tạo điều kiện thuận lợi cho việc phân loại, quản lý và truy xuất thông tin.

Cấu trúc phân cấp tương đối phổ biến và được sử dụng rộng rãi bởi các hệ điều hành, các trình quản lý văn bản thông dụng bởi vì những ưu điểm sau:

- Cấu trúc quen thuộc với người dùng: Ta dễ dàng bắt gặp kiểu cấu trúc này ở khắp mọi nơi như trường học, thư viện, công ty, chính phủ vì nó thuận lợi cho việc phân cấp và phân loại. Vì vậy cấu trúc này cũng được rất nhiều sản phẩm về quản lý văn bản sử dụng vì độ tiện dụng và nhằm hướng tới sự thân thiện với người dùng.
- Trực quan và đơn giản: Các mục thông tin liên quan đến nhau được được tổ chức theo logic và ràng buộc mối quan hệ (cha-con) nên dễ dàng được nắm bắt.
- Dễ dàng mở rộng và sắp xếp: Vì thực hiện theo dạng cây nên việc thêm một node vào "cây" cấu trúc sẽ dễ dàng và không bị ảnh hưởng đến cấu trúc tổng thể.
- Tái sử dụng cấu trúc: Nếu có sẵn một phân cấp phù hợp ta có thể tái sử dụng cấu trúc có sẵn để thêm vào dữ liệu mới mà không cần phải cấu trúc lại từ đầu.
- Hỗ trợ việc tìm kiếm và truy xuất thông tin: Dựa vào mô hình phân cấp, Hierarchical cho phép người dùng giảm không gian tìm kiếm và loại bỏ mọi sự mơ hồ của một thuật ngữ có thể đề cập đến các ngữ cảnh khác nhau.

Cấu trúc phân cấp cho phép người dùng có thể dễ dàng cấu trúc dữ liệu dễ dàng nhờ sự đơn giản và trực quan của nó nhưng khi có một nguồn dữ liệu khổng lồ thì việc có thể cấu trúc thông tin phù hợp lại trở thành việc khó khăn.

- Thiếu tính linh hoạt: Điều này không gây quá nhiều khó khăn vào thời gian đầu khi hình thành một cấu trúc mới, nhưng về sau khi cấu trúc đã trở nên to và đồ sộ, khi các cấu trúc hiện có trở nên lỗi thời và không còn đáp ứng được nhu cầu của người dùng thì việc sắp xếp lại các file và thay đổi cả cấu trúc sẽ trở nên cực kì khó khăn. Sự cứng nhắc này sẽ ảnh hưởng đến việc lưu trữ một số loại file nhất định.
- Trùng lặp, chồng chéo và dư thừa thông tin: Việc tồn tại các thư mục chỉ chứa một thư mục con, hoặc tồn tại những thư mục con giống nhau trong các thư mục

cha khác nhau thường xuyên xảy ra khi dùng Hierarchical, điều này gây trùng lặp, chồng chéo thông tin gây lãng phí tài nguyên lưu trữ một cách không cần thiết, không tạo ra được sự nhất quán của cấu trúc và còn làm ảnh hưởng đến quá trình truy xuất dữ liệu.

- Hạn chế trong việc thể hiện ngữ cảnh: Ngữ cảnh trong cấu trúc này chủ yếu được thể hiện thông qua vị trí của node, vậy nên nếu không có quy ước về đặt tên hay có thêm các thông tin về node được lưu trữ bên ngoài thì sẽ tạo nên sự hạn chế và làm ảnh hưởng đến quá trình tìm kiếm.

3.1.2 Cấu trúc phẳng (Flat)

Trong khi cấu trúc phân cấp phổ biến trong quản lý thông tin, việc tổ chức các mục thông tin thành một cấu trúc phẳng gần đây đã trở nên phổ biến. Trong cấu trúc phẳng, người dùng gán thẻ hoặc thuộc tính cho các mục thông tin. Các thẻ này được sử dụng để nhóm hoặc truy xuất các mục thông tin, cung cấp quyền truy cập liên kết vào các mục (Dourish và cộng sự, 1999, 2000; Gifford và cộng sự, 1991; Gopal và Manber, 1999). Cách tiếp cận này bây giờ được gọi là Tagging – gắn thẻ.

Cấu trúc thư mục phẳng nhìn chung có thể khắc phục một số điểm yếu của cấu trúc phân cấp, chẳng hạn như:

- Linh hoạt trong sắp xếp: Tagging giúp linh hoạt để sắp xếp các mục thông tin. Người dùng có thể phân loại một mục thông tin thành nhiều danh mục bằng cách gán nhiều thẻ cho mục đó. Dẫn đến việc nhóm và thay đổi nhóm các mục thông tin có thể được thực hiện một cách linh hoạt.

- Thể hiện được ngữ cảnh của thông tin: Tagging tạo thành hệ thống các từ khóa, chúng liên kết với nhau giúp giảm bớt gánh nặng trong việc tạo danh mục để phân loại các mục thông tin, thông qua các thẻ cũng có thể biết được nhiều thông tin ngữ cảnh của dữ liệu.

- Hỗ trợ trong việc tìm kiếm dữ liệu: Tagging tìm kiếm dữ liệu thông qua các từ khóa của thẻ được gắn chung với mục dữ liệu, các thẻ sẽ có mối liên quan với nhau giúp hỗ trợ tìm kiếm dữ liệu nhanh chóng hơn.

Bên cạnh đó, cấu trúc phẳng vẫn có một số nhược điểm lớn:

- Dữ liệu không có tính nhất quán: Việc tự do liên kết nhiều thẻ với một mục thông tin dẫn đến sự không nhất quán trong việc gán thẻ. Sự không nhất quán này ngăn người dùng truy xuất tất cả các mục có liên quan trong bộ sưu tập cùng một lúc.

- Khó khăn trong việc xác định phạm vi của từ khóa: Nếu hệ thống không có chức năng gắn thẻ tự động hoặc hỗ trợ gắn thẻ thủ công thì sẽ khiến người dùng cảm

thấy khó khăn trong việc xác định từ khóa khi gắn thẻ và làm nhiễu dữ liệu, tạo ra các từ khóa không liên quan hoặc quá rộng.

- Khó khăn khi tìm kiếm dữ liệu nếu không được cấu trúc tốt: Nếu không tạo ra các thẻ đủ tốt gây mất nhất quán trong dữ liệu, thì điểm mạnh của Tagging sẽ trở thành điểm yếu khi việc truy xuất sẽ gặp khó khăn khi dữ liệu bị nhiễu trong quá trình tìm kiếm.

3.2 Nhận dạng ký tự quang học (OCR)

3.2.1 Định nghĩa và công dụng

Nhận dạng ký tự quang học (Optical Character Recognition, viết tắt là OCR) là quá trình chuyển đổi một hình ảnh văn bản thành định dạng văn bản mà máy có thể đọc được. Nếu bạn quét một tài liệu nào đó, máy tính sẽ lưu bản quét đó dưới dạng tệp hình ảnh. Vì là hình ảnh nên không thể sử dụng trình soạn thảo văn bản để chỉnh sửa, tìm kiếm hoặc đếm số từ. Vậy nên công nghệ OCR có thể giúp chuyển đổi hình ảnh thành tài liệu văn bản (text), trong đó phần nội dung sẽ được lưu trữ dưới dạng dữ liệu văn bản[7].

3.2.2 Những phương pháp nhận dạng

- **Phương pháp so khớp mẫu (Template Matching Approach)**: Cách thức hoạt động của phương pháp này 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 pháp này hoạt động tốt đối với hình ảnh quét từ tài liệu được đánh máy bằng phông chữ đã biết, thường được sử dụng cho các dữ liệu dạng in, có một định dạng và phông chữ thống nhất không bị sai lệch quá nhiều.

- **Phương pháp phân tích cấu trúc (Structure Analysis Method)**: Phương pháp so khớp mẫu chỉ thích hợp cho việc nhận dạng các ký tự in. Tuy nhiên, nhu cầu sử dụng công nghệ OCR cho các tập dữ liệu viết tay rất phổ biến. Sự biến đổi hình dạng của các ký tự viết tay quá lớn nên khó có thể tạo nên một khuôn mẫu cho chúng. Do đó phương pháp phân tích cấu trúc ra đời để ứng dụng vào nhận dạng ký tự viết tay. Phương pháp này 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.[7, 8]

3.2.3 Các bước hoạt động của OCR

Công cụ OCR thường sẽ hoạt động theo các bước sau:

- **Bước 1. 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.

- **Bước 2. 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.

- **Bước 3. Nhận dạng văn bản:** Sử dụng hai phương pháp so khớp mẫu và phân tích cấu trúc đã nêu ở phần trên để tiến hành nhận dạng văn bản.

- **Bước 4. 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.

3.2.4 Những loại công nghệ OCR

Các nhà khoa học dữ liệu phân loại những công nghệ OCR khác nhau dựa trên mục đích sử dụng và ứng dụng của chúng. Tiêu biểu ta có thể phân loại thành các dạng sau đây[7]:

- Phần mềm nhận dạng ký tự quang học đơn giản: Một công cụ OCR đơn giản hoạt động bằng cách lưu trữ nhiều khuôn thức hình ảnh văn bản và phông chữ khác nhau dưới dạng mẫu. Phần mềm OCR sử dụng các thuật toán so khớp mẫu để so sánh các hình ảnh văn bản, theo từng ký tự một, với cơ sở dữ liệu nội bộ. Nếu hệ thống so khớp văn bản theo từng từ một thì sẽ được gọi là nhận dạng từ quang học. Giải pháp này có những hạn chế vì số lượng phông chữ và kiểu chữ viết tay là gần như vô hạn, cũng như không thể ghi lại hay lưu trữ tất cả kiểu loại trong cơ sở dữ liệu được.

- Phần mềm nhận dạng ký tự thông minh: Những hệ thống này sử dụng các phương thức nâng cao để đào tạo máy hoạt động giống như con người bằng cách sử dụng máy học. Một hệ thống máy học được gọi là mạng nơ-ron phân tích văn bản qua nhiều cấp độ, xử lý hình ảnh lặp đi lặp lại. Hệ thống sẽ tìm kiếm các thuộc tính hình ảnh khác nhau, chẳng hạn như nét cong, nét thẳng, nét giao nhau và nét vòng, đồng thời kết hợp kết quả của tất cả các cấp độ phân tích khác nhau này để cho ra

kết quả cuối cùng. Các hệ thống này tuy phải xử lý hình ảnh theo từng ký tự một nhưng quá trình này vẫn diễn ra nhanh chóng, thu được kết quả chỉ trong vài giây.

- Nhận dạng từ thông minh: Hệ thống nhận dạng từ thông minh hoạt động theo nguyên tắc giống như hệ thống nhận dạng ký tự thông minh, nhưng sẽ xử lý theo các từ thay vì xử lý theo từng ký tự.
- Nhận dạng ký hiệu quang học: Nhận dạng ký hiệu quang học xác định logo, hình mờ và các biểu tượng văn bản khác trong tài liệu.

3.3 Xử lý ngôn ngữ tự nhiên

3.3.1 Các bước xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (natural language processing - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ-công cụ hoàn hảo nhất của tư duy và giao tiếp[9]. Quá trình xử lý ngôn ngữ tự nhiên thường được thực hiện qua 5 bước:

- Phân tích hình thái - Trong bước này từng từ sẽ được phân tích và các ký tự không phải chữ (như các dấu câu) sẽ được tách ra khỏi các từ. Trong tiếng Anh và nhiều ngôn ngữ khác, các từ được phân tách với nhau bằng dấu cách. Tuy nhiên trong tiếng Việt, dấu cách được dùng để phân tách các tiếng (âm tiết) chứ không phải từ. Cùng với các ngôn ngữ như tiếng Trung, tiếng Hàn,tiếng Nhật, phân tách từ trong tiếng Việt là một công việc không hề đơn giản.
- Phân tích cú pháp - Dãy các từ sẽ được biến đổi thành các cấu trúc thể hiện sự liên kết giữa các từ này. Sẽ có những dãy từ bị loại do vi phạm các luật văn phạm.
- Phân tích ngữ nghĩa - Thêm ngữ nghĩa vào các cấu trúc được tạo ra bởi bộ phân tích cú pháp.
- Tích hợp văn bản - Ngữ nghĩa của một câu riêng biệt có thể phụ thuộc vào những câu đứng trước, đồng thời nó cũng có thể ảnh hưởng đến các câu phía sau.
- Phân tích thực nghĩa - Cấu trúc thể hiện điều được phát ngôn sẽ được thông dịch lại để xác định nó thật sự có nghĩa là gì.

Tuy nhiên, ranh giới giữa 5 bước xử lý này cũng rất mong manh. Chúng có thể được tiến hành từng bước một, hoặc tiến hành cùng lúc - tùy thuộc vào giải thuật và ngữ cảnh cụ thể.

3.3.2 Word Embedding trong xử lý ngôn ngữ tự nhiên

Word Embedding là một không gian vector dùng để biểu diễn dữ liệu có khả năng miêu tả được mối liên hệ, sự tương đồng về mặt ngữ nghĩa, văn cảnh (context) của dữ liệu. Không gian này bao gồm nhiều chiều và các từ trong không gian đó mà có cùng văn cảnh hoặc ngữ nghĩa sẽ có vị trí gần nhau[10].

Có 2 phương pháp chủ yếu được hay dùng để tính toán Word Embedding là Count based method và Predictive method. Cả hai cách này đều dựa trên một giả thuyết rằng những từ nào xuất hiện trong cùng một văn cảnh, một ngữ nghĩa sẽ có vị trí gần nhau trong không gian mới được biến đổi.

Một thuật toán nổi bật trong Count based method là TF-IDF (Term frequency-inverse document frequency). Thuật toán này được chia thành 2 phần, bao gồm TF và IDF[11]:

- TF là tần suất xuất hiện của một từ hoặc cụm từ trong một văn bản. Tần suất này có thể được định nghĩa bằng nhiều cách như: Số lần từ xuất hiện trong tài liệu (số liệu thô), số lần từ xuất hiện trong một phần của tài liệu chia cho tổng số từ trong tài liệu, sử dụng lược đồ chuẩn hóa logarit (ví dụ: log(1 + số liệu thô)) hoặc đơn giản hơn là đánh số 1 nếu xuất hiện và 0 nếu ngược lại.
- IDF được xác định như là mức độ phổ biến của một từ trong một kho văn bản. Với IDF, trọng số cho các từ được xuất hiện nhiều như giới từ, liên từ,... sẽ được giảm thiểu.
- Giá trị trọng số cuối cùng của từ được tính bằng cách lấy TF chia cho IDF.

Khác so với Count-based method, Predictive method tính toán sự tương đồng ngữ nghĩa giữa các từ để dự đoán từ tiếp theo bằng cách đưa qua một mạng neural network có một hoặc vài layer dựa trên đầu vào là các từ xung quanh (context word). Một context word có thể là một hoặc nhiều từ. Một số thuật toán tiêu biểu bao gồm:

- Bag of Words (BoW)
- Word2Vec

3.4 Kiến trúc Microservices

Microservice là một kỹ thuật phát triển phần mềm, một biến thể của kiến trúc hướng dịch vụ (SOA), nơi mà cấu trúc một ứng dụng như một tập hợp các dịch vụ được ghép lỏng lẻo với nhau. Trong kiến trúc microservice, các dịch vụ được xử lý tốt và các giao thức rất nhẹ. Lợi ích của việc phân tách một ứng dụng thành các dịch vụ nhỏ hơn là nó cải thiện tính mô đun. Điều này làm cho ứng dụng dễ hiểu, phát triển, thử nghiệm hơn và trở nên linh hoạt hơn đối với sự xói mòn kiến trúc. Nó giúp cho việc phát triển

song song bằng cách cho phép các nhóm nhỏ phát triển, triển khai và mở rộng quy mô dịch vụ tương ứng của họ một cách độc lập. Nó cũng cho phép kiến trúc của một dịch vụ riêng lẻ xuất hiện thông qua tái cấu trúc liên tục. Kiến trúc dựa trên microservice cho phép phân phối và triển khai liên tục[12].

Về cơ bản, để xem xét một hệ thống có phù hợp với kiến trúc micro-service, ta có thể đánh giá qua các đề mục sau [13]:

- Sử dụng quy trình Agile cho phát triển phần mềm.
- Yêu cầu tốc độ phát hành (release) phần mềm nhanh.
- Hệ thống có khả năng mở rộng tốt.
- Hệ thống có nhiều tên miền (domains) và miền con (subdomains).
- Tổ chức bao gồm nhiều nhóm nhỏ phát triển hệ thống.

4 Phân tích và giải pháp

4.1 Cấu trúc dữ liệu

Tuy rất trực quan và dễ sử dụng, nhưng hạn chế lớn nhất của hai kiểu cấu trúc phân cấp (Hierarchical) và cấu trúc phẳng (Flat) chính là tính phụ thuộc quá nhiều vào người xây dựng cấu trúc. Như đã nói mỗi người đều có thế giới quan và nhận định riêng, do đó việc xây dựng cấu trúc dữ liệu dựa vào quá nhiều yếu tố con người sẽ gây khó khăn cho những người dùng khác khi sử dụng và tìm kiếm, gây nên các điểm yếu như trùng lặp, chồng chéo dữ liệu, các dữ liệu giống nhau xuất hiện tại nhiều vị trí gây nên tốn kém không gian lưu trữ, ngoài ra, còn khó khăn trong việc tìm kiếm thủ công do mỗi người dùng sẽ hiểu tài liệu theo một cách khác nhau.

Vì những lý do đó, chúng tôi đề xuất một phương pháp sử dụng kết hợp cả hai cấu trúc dữ liệu cho hệ thống. Hệ thống sẽ phải xây dựng một khung cấu trúc đầy đủ theo chuẩn có sẵn cho từng miền (Domain) khác nhau để người dùng có thể sử dụng. Tuy nhiên, phạm vi chủ đề của tài liệu và văn bản rất rộng, một hệ thống gần như không thể tạo dựng một khung cấu trúc hoàn chỉnh cho toàn bộ các lĩnh vực. Vì vậy, hệ thống sẽ kết hợp sử dụng cấu trúc cây phân cấp (Hierarchical) và cấu trúc phẳng (Flat), nhằm hỗ trợ người dùng có thể tự tạo cấu trúc dữ liệu khác ngoài miền hệ thống hỗ trợ. Nói cách khác, cấu trúc dữ liệu của hệ thống sẽ sử dụng khung văn bản được tạo dựng sẵn, kết hợp với cho phép người dùng tạo dựng cấu trúc dữ liệu cho những lĩnh vực ngoài miền được hệ thống hỗ trợ. Cách thức hoạt động của mỗi phương thức cụ thể như sau:

4.1.1 Xây dựng khung cấu trúc quy chuẩn theo miền

Vì giới hạn về thời gian, chúng tôi sẽ chỉ phát triển hai miền chính là Văn bản hành chính công ty và thư viện sách. Mỗi miền này sẽ hoạt động riêng lẻ nhau.

Mỗi miền sẽ được xây dựng với những tiêu chí nhất định (sẽ được trình bày cụ thể ở phần sau). Dựa vào các tiêu chí đó, hệ thống có thể xác định được vị trí của tài liệu nằm ở đâu trên khung đã dựng sẵn. Bản chất về mặt trình bày (presentation) lên phần nhìn (view) cho người dùng, toàn bộ khung sẽ không hiển thị, chỉ khi tồn tại tài liệu ở những nhánh đó thì mới được hiển thị lên cho người dùng thấy. Do cơ chế này nên một tài liệu có thể xuất hiện ở nhiều nơi trên khung hệ thống.

Như đã nói, phương thức này sẽ giải quyết điểm yếu phụ thuộc vào người xây dựng cấu trúc, bởi vì khung đã dựng sẵn theo quy chuẩn, nên nhiều người dùng có thể sử dụng và không gây hiểu lầm giữa các người dùng với nhau. Ở các phần sau, chúng tôi sẽ nói rõ cách hoạt động của sự liên kết giữa tài liệu với miền văn bản đã dựng sẵn.

4.1.2 Cấu trúc dữ liệu do người dùng tự tạo

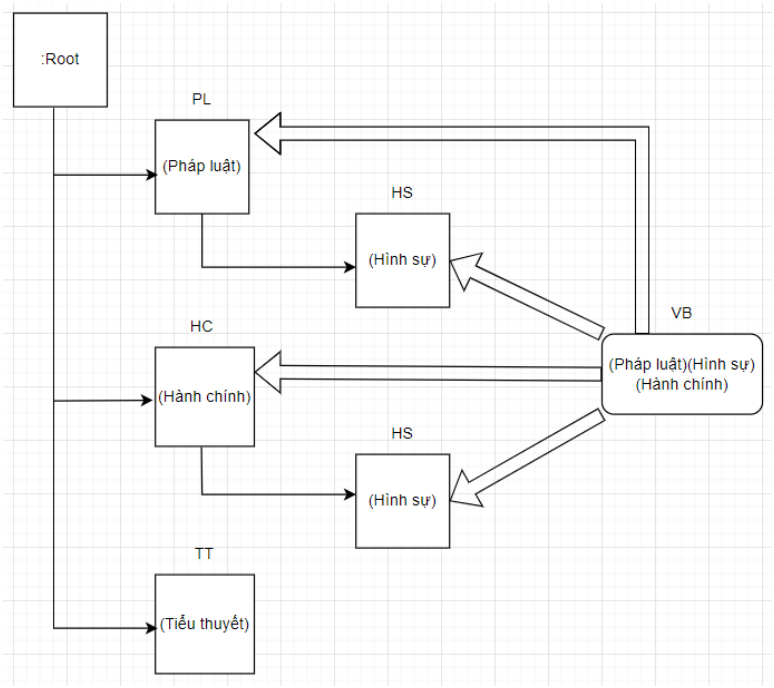
Đối với cấu trúc dữ liệu do người dùng tự tạo, hệ thống vẫn sẽ dùng cấu trúc phân cấp để xây dựng. Tuy nhiên, cơ chế thư mục sẽ được thay đổi và không hiện thực cứng nhắc như cấu trúc phân cấp thông thường.

Để thực hiện được yêu cầu này, các thư mục khi tạo sẽ được hệ thống yêu cầu xác định ít nhất một tiêu chí. Tiêu chí của thư mục sẽ được người dùng tạo, chẳng hạn như: Pháp luật, Hành chính,... Các tiêu chí này giúp xác định các văn bản nào thuộc phạm vi của thư mục nào. Nên chú ý rằng tiêu chí của thư mục một khi được tạo sẽ bị hạn chế thay đổi.

Tương tự, các tập tin khi được tải lên sẽ được hệ thống yêu cầu xác định ít nhất một tiêu chí. Các tiêu chí này như đã được đề cập bên trên, nhằm quy định các tập tin đó sẽ nằm trong thư mục nào. Một điểm cần lưu ý là các tiêu chí của tập tin có thể dễ dàng được thay đổi.

Các tiêu chí để xác định văn bản của thư mục con sẽ là tiêu chí của thư mục đó cộng với tiêu chí của các thư mục cha. Dưới đây là ví dụ của cấu trúc thư mục này, bao gồm:

- Tổng cộng 6 thư mục, bao gồm thư mục root, thư mục PL với tiêu chí Pháp Luật, thư mục con HS của thư mục PL với tiêu chí Hình sự, thư mục HC với tiêu chí Hành chính, thư mục con HS của thư mục HC với tiêu chí hình sự và thư mục TT với tiêu chí Tiểu thuyết.
- Văn bản VB với 3 tiêu chí: Pháp luật, Hình sự, Hành chính.



Hình 4.1.1: Ví dụ về xây dựng cấu trúc thư mục mới

Nếu người dùng chọn vào thư mục PL, HC thì VB phải thuộc 2 thư mục này vì VB có 2 chỉ tiêu: Pháp luật và Hành chính. Tương tự VB thuộc cả 2 folder HS vì văn bản này có các chỉ tiêu: (Pháp luật, hình sự) và (Hành chính, hình sự). Riêng thư mục TT thì VB không thuộc vì không có chỉ mục Tiểu thuyết.

Cách tiếp cận này mặc dù không thể đề xuất cấu trúc thư mục và phụ thuộc vào người dùng, tuy nhiên nó sẽ yêu cầu xác định chặt chẽ hơn từ người dùng nếu muốn tập tin vào đúng thư mục. Ngoài ra, nó sẽ xử lý trường hợp lặp và cách tiếp cận này giúp việc phân loại văn bản dễ dàng hơn.

4.2 Miền hỗ trợ tiêu chí

Vì để tối ưu trong quá trình hỗ trợ người dùng, hệ thống sẽ chỉ tập trung vào một số miền chính, bao gồm: Văn bản hành chính công ty và thư viện sách. Với các miền này, hệ thống sẽ tạo sẵn hai cấu trúc thư mục nhằm tối ưu việc đánh tiêu chí nếu người dùng có nhu cầu sử dụng các văn bản thuộc các lĩnh vực được nêu trên. Các bảng tiêu chí này sẽ mở rộng và hoàn thiện theo thời gian.

4.2.1 Văn bản hành chính công ty

Cấu trúc thư mục sẽ được chia thành 3 cấp, mỗi thư mục ở mỗi cấp tương ứng với một tiêu chí. Chi tiết cụ thể được thể hiện ở bảng bên dưới

Tiêu chí bậc 1	**Tiêu chí bậc 2**	**Tiêu chí bậc 3**
Hành chính	Nghị quyết	– Nghị quyết Đại hội đồng cổ đông – Nghị quyết Hội đồng quản trị
Hành chính	Quyết định	– Quyết định thôi việc – Quyết định bổ nhiệm – Quyết định phân công nhiệm vụ – Quyết định khen thưởng – Quyết định ban hành nội quy lao động – Quyết định cử nhân viên đi công tác – Quyết định thuyên chuyển công tác
Hành chính	Chỉ thị	–
Hành chính	Quy chế	– Quy chế Làm việc trong công ty – Quy chế Kỷ luật lao động – Quy chế tiền lương tiền thưởng cho người lao động
Hành chính	Quy định	– Quy định Công ty – Quy định Lao động
Hành chính	Thông cáo	– Thông cáo Báo chí

Hành chính	Thông báo	– Thông báo Truyền đạt – Thông báo Kết quả – Thông báo Thay đổi – Thông báo Nhiệm vụ – Thông báo hiệu lực
Hành chính	Kế hoạch	– Kế hoạch hành động – Kế hoạch chiến lược – Kế hoạch tài chính – Kế hoạch quản lý dự án – Kế hoạch thị trường – Kế hoạch marketing
Hành chính	Hướng dẫn	– Hướng dẫn Thủ tục – Hướng dẫn An toàn lao động – Hướng dẫn Cộng đồng – Hướng dẫn Kỹ thuật
Hành chính	Chương trình	– Chương trình đào tạo – Chương trình khuyến mãi – Chương trình hành động
Hành chính	Phương án	– Phương án kinh doanh – Phương án quản lý nhân sự – Phương án an ninh – Phương án sử dụng lao động – Phương án tài chính – Phương án phát triển sản phẩm – Phương án khẩn cấp
Hành chính	Đề án	– Đề án thành lập công ty – Đề án kinh doanh – Đề án tài chính – Đề án phát triển công ty – Đề án quản lý nhân sự – Đề án mở rộng thị trường – Đề án công nghệ thông tin
Hành chính	Dự án	– Dự án phát triển sản phẩm – Dự án phát triển hạ tầng – Dự án tổ chức sự kiện – Dự án quản lý nhân sự – Dự án cải tiến quy trình nghiệp vụ – Dự án tư duy sáng tạo
Hành chính	Báo cáo	– Báo cáo tài chính – Báo cáo tiến độ dự án

		– Báo cáo kết quả công việc – Báo cáo nhân sự – Báo cáo kế hoạch kinh doanh – Báo cáo nghiên cứu và phát triển – Báo cáo an toàn và sức khỏe nghề nghiệp – Báo cáo thị trường
Hành chính	Biên bản	– Biên bản cuộc họp – Biên bản họp nhóm làm việc – Biên bản họp đại hội cổ đông – Biên bản họp khẩn cấp – Biên bản đánh giá nhân sự – Biên bản họp đối tác – Biên bản họp nhóm dự án
Hành chính	Tờ trình	– Tờ trình đề xuất dự án – Tờ trình báo cáo sự kiện – Tờ trình thay đổi công nghệ – Tờ trình mở rộng kinh doanh – Tờ trình học bổng và đào tạo – Tờ trình mua sắm – Tờ trình xin ngân sách – Tờ trình thay đổi chính sách – Tờ trình xin quyết định quan trọng – Tờ trình nâng cao hiệu suất
Hành chính	Hợp đồng	– Hợp đồng lao động – Hợp đồng dịch vụ – Hợp đồng mua bán – Hợp đồng thuê mặt bằng – Hợp đồng nhượng quyền – Hợp đồng tài chính – Hợp đồng hợp tác – Hợp đồng chuyển giao công nghệ – Hợp đồng bảo hiểm – Hợp đồng quảng cáo – Hợp đồng sản xuất – Hợp đồng đối tác chiến lược
Hành chính	Công văn	– Công văn hướng dẫn – Công văn đôn đốc – Công văn chỉ đạo – Công văn đề nghị – Công văn phúc đáp

Hành chính	Công điện	–
Hành chính	Bản ghi nhớ	– Bản ghi nhớ làm việc – Bản ghi nhớ hợp tác
Hành chính	Bản thỏa thuận	– Bản thỏa thuận trả nợ – Bản thỏa thuận hợp tác kinh doanh – Bản thỏa thuận hợp tác ba bên – Bản thỏa thuận không thành công
Hành chính	Giấy ủy quyền	– Giấy ủy quyền cá nhân cho cá nhân – Giấy ủy quyền doanh nghiệp cho cá nhân – Giấy ủy quyền cá nhân cho doanh nghiệp – Giấy ủy quyền doanh nghiệp cho doanh nghiệp
Hành chính	Giấy mời	– Giấy mời họp – Giấy mời dự hội nghị – Giấy mời dự tiệc – Giấy mời làm việc
Hành chính	Giấy giới thiệu	– Giấy giới thiệu chuyển trường – Giấy giới thiệu rút tiền tại ngân hàng – Giấy giới thiệu đăng ký xe máy – Giấy giới thiệu người vào Đảng – Giấy giới thiệu công tác
Hành chính	Giấy nghỉ phép	– Giấy nghỉ phép có lương – Giấy nghỉ phép không lương – Giấy nghỉ phép năm – Giấy nghỉ phép có bệnh – Giấy nghỉ phép đặc biệt
Hành chính	Phiếu gửi	– Phiếu gửi hàng hóa – Phiếu gửi hàng – Phiếu gửi dịch vụ – Phiếu gửi trả hàng – Phiếu gửi thanh toán
Hành chính	Phiếu chuyển	– Phiếu chuyển hàng hóa – Phiếu chuyển khoản – Phiếu chuyển dịch vụ – Phiếu chuyển đổi sản phẩm – Phiếu chuyển nhân sự
Hành chính	Phiếu báo	– Phiếu báo nợ – Phiếu báo vật tư còn lại – Phiếu báo giá

		– Phiếu báo hàng hóa
Hành chính	Thư công	– Công thư cá nhân – Công thư kinh doanh – Công thư chính phủ

Bảng 4.2.1: Bảng tiêu chí của văn bản hành chính

4.2.2 Thư viện sách

Mục tiêu hướng tới của miền này là có thể phân loại sách ra theo thể loại và nhà xuất bản, qua đó người dùng có thể nhanh chóng tìm được cuốn sách mình cần tìm thông qua thông tin về thể loại cũng như nhà xuất bản. Bản mẫu có cấu trúc sau khi được thể hiện (visualize) cho người dùng sẽ có 4 bậc là: Thư viện sách – Phân loại sách – Nhà xuất bản – Thể loại. Với bậc Phân loại sách sẽ có 2 phân loại:

1. **Fiction book chính là sách hư cấu:** Là những câu truyện hư cấu hoặc giả tưởng được viết lên dưới bằng trí tưởng tượng của người tác giả.
2. **Non-fiction book chính là sách phi hư cấu:** Là những cuốn sách viết về những sự kiện, câu chuyện có thật, thể hiện quan điểm và ý tưởng của tác giả về những câu truyện trong thực tế.

Vì hệ thống hướng tới người dùng Việt Nam, nên mục nhà xuất bản sẽ phân loại theo các nhà xuất bản (NXB) của Việt Nam là:

1. NXB Bách Khoa Hà Nội
2. NXB Chính trị Quốc gia Sự thật
3. NXB Công thương
4. NXB Công an Nhân dân
5. NXB Dân trí
6. NXB Giao thông Vận tải
7. NXB Giáo dục Việt Nam
8. NXB Hàng hải
9. NXB Học viện Nông nghiệp
10. NXB Hồng Đức
11. NXB Hội Nhà năn
12. NXB Khoa học Tự nhiên và Công nghệ
13. NXB Khoa học và Kỹ thuật
14. NXB Khoa học Xã hội
15. NXB Kim Đồng
16. NXB Kinh tế thành phố Hồ Chí Minh
17. NXB Lao động
18. NXB Lao động – Xã hội
19. NXB Lý luận Chính trị
20. NXB Mỹ thuật
21. NXB Nông nghiệp
22. NXB Phụ nữ
23. NXB Quân đội Nhân dân
24. NXB Sân khấu
25. NXB Thanh niên

26. NXB Thông tin và Truyền thông
27. NXB Thông tấn
28. NXB Thế giới
29. NXB Thể thao và Du lịch
30. NXB Thống kê
31. NXB Thời đại
32. NXB Tri thức
33. NXB Tài chính
34. NXB Tài nguyên – Môi trường và Bản đồ Việt Nam
35. NXB Tôn giáo
36. NXB Tư Pháp
37. NXB Văn hóa – Thông tin
38. NXB Văn hóa dân tộc
39. NXB Văn học
40. NXB Xây dựng
41. NXB Y học
42. NXB Âm nhạc
43. NXB Đại học Công nghiệp Thành phố Hồ Chí Minh
44. NXB Đại học Cần Thơ
45. NXB Đại học Huế
46. NXB Đại học Kinh tế Quốc Dân
47. NXB Đại học Quốc gia Hà Nội
48. NXB Đại học Quốc gia Thành phố Hồ Chí Minh
49. NXB Đại học Sư phạm
50. NXB Đại học Thái Nguyên
51. NXB Đại học Vinh
52. NXB Đại học Sư phạm Thành phố Hồ Chí Minh
53. NXB Hà Nội
54. NXB Hải Phòng
55. NXB Nghệ An
56. NXB Phương Đông
57. NXB Thanh Hóa
58. NXB Thuận Hóa
59. NXB Trẻ
60. NXB Tổng hợp Thành phố Hồ Chí Minh
61. NXB Văn hóa – Văn nghệ Thành phố Hồ Chí Minh
62. NXB Đà Nẵng
63. NXB Đồng Nai

Trên đây là 63 nhà xuất bản do Cục Xuất bản, in và phát hành phê duyệt.[14] Bên cạnh đó để hỗ trợ ngoại văn sẽ có thêm mục số 34 là Ngoại văn và mục số 35 là NXB Khác.

Với thể loại sẽ có rất nhiều cách và tiêu chí để phân loại, mỗi nơi đều có những cách phân loại riêng biệt, ví dụ Reedsy Discovery phân loại theo 107 thể loại khác nhau, Amazon lên tới 16,000 thể loại. Nhưng để thống nhất và tránh thừa thải nhóm sẽ chỉ phân loại sách theo 33 thể loại chính nhất trong văn học, dưới đây nhóm sẽ đưa ra 33 thể loại cùng với các đặc điểm của nó để có thể xây dựng được khung tiêu chí và từ khóa

(keyword) liên quan.[15]

- Với fiction:

    1. **Hư cấu kỳ ảo (Fantasy):** Thể loại sách này đặc trưng bởi các yếu tố ma thuật hoặc siêu nhiên và thường được lấy cảm hứng từ thần thoại hoặc văn hóa dân gian.
    2. **Khoa học viễn tưởng (Science Fiction):** Gần giống với fantasy, Khoa học viễn tưởng cũng hướng tới các yếu tố siêu nhiên và hư cấu (Giả tưởng suy đoán – speculative fiction). nhưng khoa học viễn tưởng được phân biệt bởi mối bận tâm của nó với khoa học thực tế.
    3. **Phản địa đàng (Dystopian):** Là một thể loại khoa học viễn tưởng phổ biến mang đến một tầm nhìn ảm đạm và đáng sợ về tương lai: thường là một xã hội nghiệt ngã, sau một thảm họa, phải đối mặt với những thứ như chính phủ áp bức, công nghệ kiểu Black Mirror và sự tàn phá môi trường.
    4. **Hành động và Phiêu lưu (Action & Adventure):** Chứa đựng một cuộc hành trình đầy rủi ro và một chuỗi hành động ly kỳ.
    5. **Thần bí (Mystery) hay Trinh thám (Detective):** Có đặc điểm là cốt truyện hấp dẫn xoay quanh một bí ẩn, dẫn dắt giải được manh mối đó.
    6. **Kinh dị (Horror):** Là một thể loại biểu đạt với mục đích hoặc có khả năng tạo ra những cảm xúc tiêu cực như lo lắng, sợ hãi hay làm độc giả, khán giả giật mình bằng những nội dung kinh hoàng.
    7. **Hài hước (Comedy):** Là những tác phẩm gây cười cho người đọc.
    8. **Giật gân (Thriller & Suspense):** Thường được xây dựng với những yếu tố tạo cảm giác bất ngờ, hồi hộp, sợ hãi hay lo lắng cho khán giả.
    9. **Lịch sử hư cấu (Historical Fiction):** Là thể loại văn học trong đó cốt truyện hư cấu diễn ra trong bối cảnh các sự kiện lịch sử có thật cụ thể.
    10. **Tình cảm (Romance):** Mối quan hệ lãng mạn phải là điểm trung tâm của cốt truyện.
    11. **Tiểu thuyết giành cho phụ nữ (Women's Fiction):** Lấy phụ nữ làm trung tâm của câu chuyện.
    12. **Văn học đương đại (Contemporary Fiction):** Thể loại sách này đôi khi được gộp chung với những thể loại khác để cho biết rằng cuốn sách diễn ra ở thời điểm hiện tại.
    13. **Chủ nghĩa hiện thực huyền ảo (Magical Realism):** Chủ nghĩa này quan niệm rằng thực tại còn có cả đời sống tâm linh, niềm tin tôn giáo, các huyền thoại và truyền thuyết.

14. **Tiểu thuyết hình ảnh (Graphic novel):** Là một thể loại sách tiểu thuyết được viết và vẽ theo kiểu truyện tranh comics nhưng không phải là truyện tranh comics.
15. **Truyện ngắn (Short story):** Có xu hướng ngắn gọn, súc tích và hàm nghĩa hơn các câu truyện dài như tiểu thuyết.
16. **Thanh niên (Young Adult):** Dành cho độ tuổi từ 12-18.
17. **Người lớn (Adult):** Dành cho độ tuổi từ 18 trở lên.
18. **Trẻ em (Children):** Dành cho trẻ em dưới 12 tuổi.

- Với non-fiction:

1. **Hồi ký và tự truyện (Memoir & Autobiography):** Cung cấp một tường thuật chân thực về cuộc đời của tác giả.
2. **Tiểu sử (Biography):** Mô tả không chỉ những sự kiện cơ bản như giáo dục, công việc, các mối quan hệ và cái chết; mà còn miêu tả trải nghiệm của một người về những sự kiện cuộc sống của một người.
3. **Đồ ăn và thức uống (Food and Drink) hay sách nấu ăn (Cookbook)**
4. **Mỹ thuật và Tranh ảnh (Art and Photography)**
5. **Tự lực (Self-help):** Mục đích hướng dẫn độc giả giải quyết những vấn đề cá nhân.
6. **Lịch sử (History):** Trình bày những sự thật đã biết về một thời đại, sự kiện hoặc nhân vật lịch sử.
7. **Văn học du lịch (Travel):** Hồi ký và tạp chí du lịch
8. **Tội phạm có thật (True crime):** Tác giả xem xét một tội ác và nêu chi tiết hành động của những người liên quan và bị ảnh hưởng bởi các sự kiện tội phạm.
9. **Hài (Humor):** Hồi ký nhưng có khả năng gây cười.
10. **Tiểu luận (Esssay)**
11. **Hướng dẫn (Guide/How to)**
12. **Tâm linh (Religion Spirituality):** Mọi thứ liên quan đến chủ đề tôn giáo và tâm linh.
13. **Khoa học xã hội và nhân văn (Humanities & Social Sciences):** Triết học, lịch sử, văn học, ngôn ngữ, nghệ thuật, tôn giáo, âm nhạc hoặc thân phận con người.
14. **Khoa học và Công nghệ (Science & Technology)**
15. **Sách giáo khoa (Textbook)**

Tất cả các thể loại trên đều có thể xuất hiện trong bất cứ nhà xuất bản nào, miễn trong quá trình bắt dữ liệu có tồn tại thể loại nêu trên.

Từ những đặc điểm đã nêu ở trên xây dựng được bảng tiêu chí như sau:

Phân loại (Type)	**Thể loại (Genre)**	**Tiêu chí tiếng Việt (Vi)**	**Tiêu chí tiếng Anh (En)**
Truyện, sách hư cấu (Fiction)	Hư cấu kỳ ảo (Fantasy)	Ma thuật Siêu nhiên Kỳ ảo Thần thoại	Magic Supernatural Fantasy Myth
	Khoa học viễn tưởng (Science Fiction)	Khoa học Khoa học viễn tưởng Tương lai Công nghệ Siêu nhiên	Science Science Fiction (sci-fic) Future Technical, Technology Supernatural
	Phản địa đàng (Dystopian)	Phản địa đàng Tương lai Chính trị Sụp đổ Tàn phá Công nghệ Chiến tranh Khủng khiếp Vô cảm	Dystopian Future Political, politic Collapse Destroy Technical, Technology War, warfare Terrify, terrible Emotionless
	Hành động và Phiêu lưu (Action & Adventure)	Hành động Thử thách Trận đánh Phiêu lưu Khám phá Chuyến đi	Action Challenge Battle, combat, fight Adventure Explore, discover Trip
	Thần bí (Mystery) & Trinh thám (Dectective)	Thần bí Trinh thám Tội ác, tội phạm Cảnh sát Manh mối	Mystery Detective Crime Police Clue
	Kinh dị (Horror)	Kinh dị Lo lắng Sợ hãi Hoảng hốt	Horror Worry, anxiety Scare, fear Panic

	Giật mình Ma quỷ Máu Ma ám	Startle Ghost Blood Hauted
Hài hước (Comedy)	Hài hước Buồn cười Nói đùa	Comedy Funny Joke
Giật gân (Thriller & Suspense)	Giật gân Hồi hộp Bất ngờ, bất thình lình Lo lắng Sợ hãi	Thriller Suspense Unforeseen, surprise Worry, anxiety Scare, fear
Lịch sử hư cấu (Historical fiction)	Lịch sử Quá khứ Chính trị Sự kiện	Historical, history Past Political, politic Event
Tình cảm (Romance)	Tình cảm Tình yêu, yêu Cặp đôi	Romance Love Couple
Tiểu thuyết giành cho phụ nữ (Women's Fiction)	Tiểu thuyết giành cho phụ nữ	Women's Fiction
Văn học đương đại (Contemporary	Đương đại Hiện tại	Contemporary Present
Fiction) Chủ nghĩa hiện thực huyền ảo (Magical Realism)	Chủ nghĩa hiện thực huyền ảo Tâm linh Tôn giáo Huyền thoại Hiện thực Thực tại	Magical Realism Spirituality Religion Legendary, legend Realistic Reality
Tiểu thuyết hình ảnh (Graphic novel)	Hình ảnh Tiểu thuyết hình ảnh	Graphic Graphic novel
Truyện ngắn (Short story)	Truyện ngắn	Short story
	Thanh niên	Young Adult

	Thanh niên (Young Adult)	Tuổi trẻ	Youth
	Người lớn (Adult)	Người lớn Dành cho người trên 18 tuổi	Adult Censored 18
	Trẻ em (Children)	Trẻ em, thiếu nhi, trẻ con	Children
Sách phi hư cấu (Non-fiction)	Hồi ký và tự truyện (Memoir & Autobiography)	Hồi ký Tự truyện Tiểu sử Cuộc đời Trải nghiệm Chuyến đi	Memoir Autobiography Biography Life, lifetime Experience Trip
	Tiểu sử (Biography)	Tiểu sử Công việc Cuộc đời Trải nghiệm	Biography Job, business Life, lifetime Experience
	Đồ ăn và thức uống (Food and Drink) hay sách nấu ăn (Cookbook)	Sách nấu ăn Thực đơn	Cookbook Menu
	Mỹ thuật và Tranh ảnh (Art and Photography)	Mỹ thuật Tranh ảnh Hình ảnh	Art Photography Picture, Image
	Tự lực (Self-help)	Tự lực Vấn đề cá nhân Giải quyết, xử lý	Self-help Individual issue Handle, solve
	Lịch sử (History)	Lịch sử Sự kiện Sự thật, có thật	History Event Reality
	Văn học du lịch (Travel)	Du lịch Hồi ký Khám phá Chuyến đi Phiêu lưu	Travel Memoir explore, discover Trip Adventure
	Tội phạm có thật (True crime)	Tội ác, tội phạm Có thật	Crime True, Reality

	Hài (Humor)	Hài hước Buồn cười Nói đùa Hồi ký	Comedy Funny Joke Memoir
	Tiểu luận (Essay)	Tiệu luận	Essay
	Hướng dẫn (Guide/How to)	Hướng dẫn	Guide, How to
	Tâm linh (Religion & Spirituality)	Tâm linh Tôn giáo	Spirituality Religion
	Khoa học xã hội và nhân văn (Humanities & Social Sciences)	Triết học Lịch sử Văn học Ngôn ngữ Tôn giáo Âm nhạc Con người Xã hội	Philosophy History Literature Language Religion Music Humanity Social
	Khoa học và Công nghệ (Science & Technology)	Khoa học Công nghệ Máy tính, điện toán Điện tử Hóa học Cơ khí	Science Technology Computer, Computing Electronic Chemistry Mechanic
	Sách giáo khoa (Textbook)	Sách giáo khoa	Textbook

Bảng 4.2.2: Bảng tiêu chí của miền thư viện sách

4.3 Đề xuất tiêu chí

4.3.1 Phương pháp sử dụng Langchain và GPT

ChatGPT hay với tên gọi đầy đủ là Chat Generative Pre-training Transformer - một chatbot do công ty khởi nghiệp OpenAI phát triển. ChatGPT có thể được hiểu đơn giản là một AI (trí thông minh nhân tạo). Điểm đặc biệt của AI này nằm ở "kho" kiến thức mà ChatGPT đã học được. Mặc dù là một chatbot rất mạnh, ChatGPT vẫn có một số điểm yếu như:

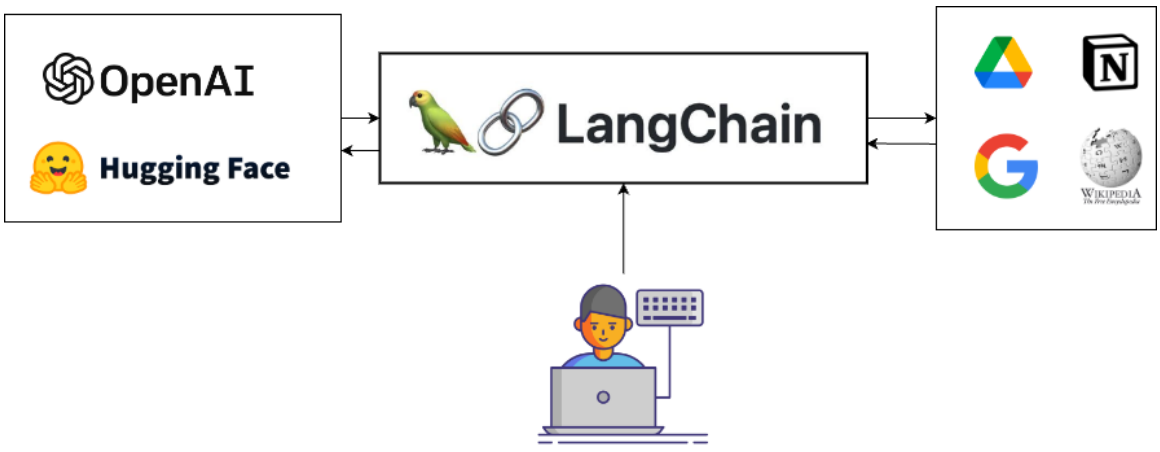
- Không có tính cập nhật: ChatGPT không thể trả lời các câu hỏi có tính cập nhập chẳng hạn như thời tiết, nhiệt độ hôm nay như thế nào? Bởi đơn giản ChatGPT

chỉ là một mô hình ngôn ngữ và chỉ có có thể trả lời được trên các thông tin nó đã được huấn luyện chứ không thể cập nhật dữ liệu một cách realtime.

- Không truy cập được vào các dữ liệu cá nhân: ChatGPT không thể trả lời được các thông tin mang tính cá nhân hoặc bảo mật, vì vậy mà khi người dùng muốn tìm giải pháp cho một số thông tin mang tính cá nhân hầu như là không thể.

Và cách để giải quyết các vấn đề trên của ChatGPT chính là Langchain. LangChain là một framework Python mã nguồn mở, cho phép lập trình viên phát triển ứng dụng do các mô hình ngôn ngữ lớn (large language model) cung cấp. Nó được sinh ra để tận dụng sức mạnh của các mô hình ngôn ngữ lớn LLM như ChatGPT, LLaMA... để tạo ra các ứng dụng trong thực tế. Những ứng dụng phổ biến của nó là chatbot, tổng kết, đặt câu hỏi và trả lời,... Dù mới được phát triển vào 10/2022 và vẫn được cập nhật liên tục hàng ngày nhưng Github Langchain đã nhận được rất nhiều chú ý từ cộng đồng lập trình viên[16].

Điểm nổi bật của Langchain là giúp người dùng không chỉ tương tác với các mô hình ngôn ngữ lớn mà còn cho phép ứng dụng của bạn tận dụng thêm các thông tin từ nhiều nguồn dữ liệu khác của bên thứ 3 như Google, Notion, Facebook... cũng như cung cấp các thành phần (component) cho phép sử dụng các mô hình ngôn ngữ trong nhiều tình huống khác nhau trên thực tế. Chức năng của Langchain được mô tả như hình bên dưới:



Hình 4.3.1: Langchain kết hợp ChatGPT

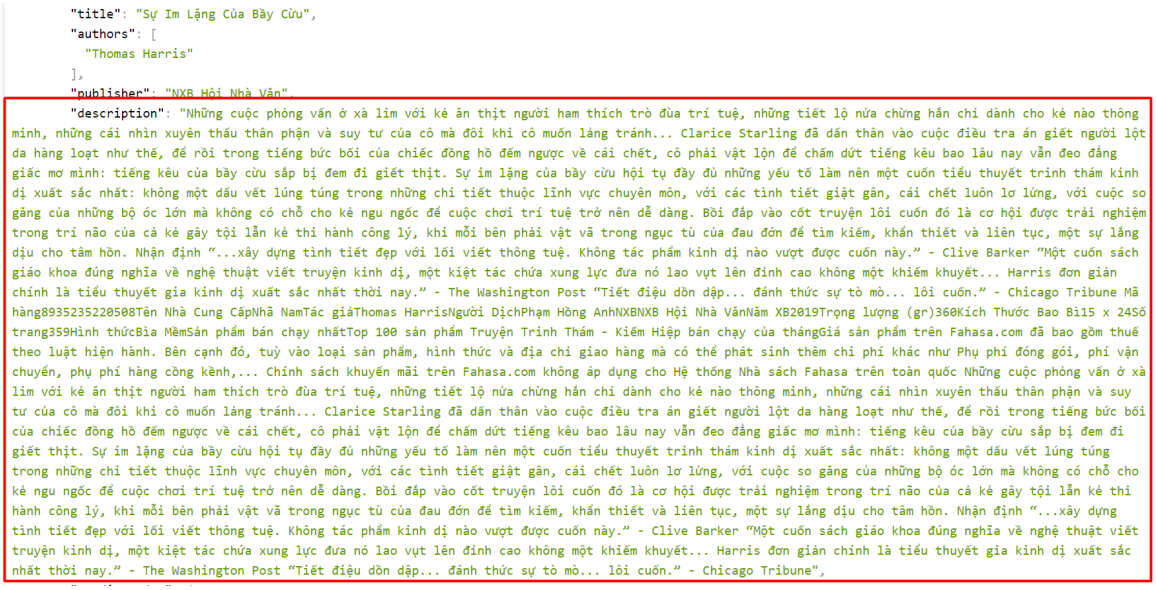
Hai ưu điểm chính của framework Langchain là:

- Cung cấp các component đa dạng: Langchain cung cấp một loạt các component cần thiết cho việc tương tác với các language model. Các component này được thiết kế dễ dàng sử dụng, mở rộng và tuỳ biến cho nhiều bài toán khác nhau.

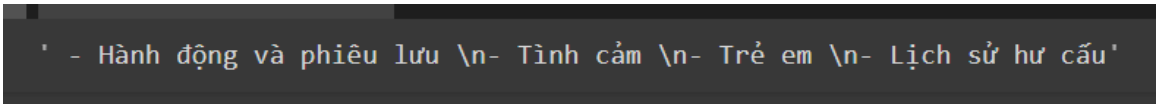
- Cung cấp các chuỗi (chain) cho các trường hợp cụ thể: Một chuỗi được hiểu là một chuỗi các component ghép nối lại với nhau theo thứ tự nhất định để từ đó có thể giải quyết được các trường hợp sử dụng trong thực tế. Các trường hợp mà Langchain cung cấp như trợ lý ảo, hỏi đáp dựa trên các tài liệu, chatbot, hỗ trợ truy vấn dữ

liệu bảng biểu, tương tác với các API, trích xuất đặc trưng của văn bản, đánh giá văn bản, tóm tắt văn bản.

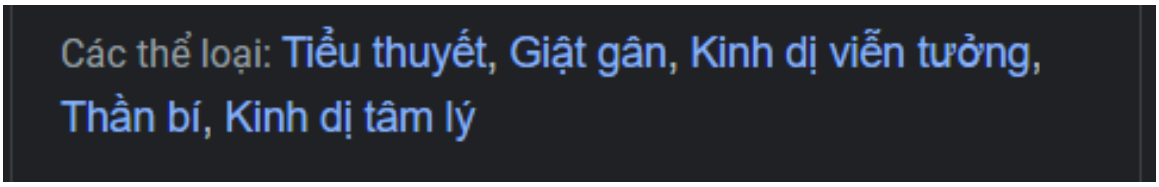
Từ những ưu điểm chính bên trên, việc áp dụng Langchain nhằm xác định tiêu chí của văn bản là khả thi. Hệ thống có thể đưa các file tiêu chí lên Google Drive và nhúng chúng vào câu lệnh truy xuất đến GPT engine thông qua Langchain. Sau đây là một số demo nhóm thực hiện với các tiêu chí được xác định tại mục 4.1 thông qua Langchain và GPT.



Hình 4.3.2: Một đoạn văn bản từ "Sự im lặng của bầy cừu" cần được đánh tiêu chí



Hình 4.3.3: Kết quả trả về từ Langchain



Hình 4.3.4: Phân loại từ nhà xuất bản

Hình 4.3.5: Một đoạn văn bản từ "Sự hiền hòa của sói" cần được đánh tiêu chí

Hình 4.3.6: Kết quả trả về từ Langchain

Hình 4.3.7: Phân loại từ nhà xuất bản

Từ kết quả trên, ta thấy được rằng Langchain và GPT cho ra kết quả rất tốt trên cả văn bản tiếng Việt và tiếng Anh.

4.3.2 Phương pháp sử dụng PhoBERT

Tuy có cùng kiến trúc Transformer nhưng khác với GPT, PhoBERT là mô hình được huấn luyện bằng các tập dữ liệu tiếng Việt từ các nguồn trên Internet. Nếu GPT tập trung vào việc sinh văn bản, thì với PhoBERT, mô hình này tập trung vào tác vụ mã hoá ngữ cảnh của các token trong đoạn văn bản.

Cách tiếp cận và áp dụng PhoBERT khá thân thiện, người dùng chỉ cần sử dụng một thư viện xử lý ngôn ngữ tự nhiên như Hugging Face Transformers hoặc PyTorch để tải và sử dụng mô hình. Người dùng có thể sử dụng mô hình PhoBERT đã được huấn luyện trước.

Cũng như các mô hình Transformer khác, PhoBERT không xử lý dữ liệu ở dạng ngôn ngữ con người. Người dùng cần chuẩn bị dữ liệu tiếng Việt và tiền xử lý nó sao cho phù hợp với định dạng đầu vào của PhoBERT. Đầu vào có thể là câu hoặc đoạn văn bản. Dữ

liệu đầu vào sẽ được mã hoá thành các token và chia thành các đoạn để giúp mô hình hiểu cấu trúc ngữ pháp của văn bản.

Mục tiêu huấn luyện của PhoBERT là tạo ra các biểu diễn vector ngữ cảnh cho mỗi token trong đoạn văn bản tiếng Việt đầu vào. Mỗi vector đại diện cho một khía cạnh nghĩa của từ đó trong ngữ cảnh của văn bản. Điều này cho phép hệ thống sử dụng các biểu diễn này cho các tác vụ ngôn ngữ như phân loại văn bản, gán nhãn, trích xuất thông tin, dịch thuật, và nhiều tác vụ khác.

Có thể thấy mô hình này có vẻ rất phù hợp cho bài toán chúng ta đang đề cập, nhưng để đáp ứng được độ chính xác đủ tốt cho mục đích sử dụng thì việc cần làm là nên huấn luyện PhoBERT dựa trên tập dữ liệu liên quan đến bài toán.

Giai đoạn huấn luyện mô hình nào cũng cần phải có dữ liệu. Có thể thấy, việc tìm kiếm và thu thập một lượng lớn dữ liệu huấn luyện chất lượng và đại diện cho ngôn ngữ tiếng Việt có thể là một thách thức. Đặc biệt là trong lĩnh vực bài toán, việc thu thập dữ liệu huấn luyện có thể gặp khó khăn trong việc đảm bảo tính đa dạng của các nguồn dữ liệu và đảm bảo chất lượng dữ liệu đối với các tác vụ cụ thể.

Đồng thời, quá trình huấn luyện PhoBERT có thể có thể mất nhiều thời gian và yêu cầu sử dụng GPU hoặc TPU mạnh để tăng tốc quá trình vì huấn luyện mô hình như PhoBERT yêu cầu một lượng lớn dữ liệu và tài nguyên tính toán phù hợp. Nhu cầu phần cứng đòi hỏi sẽ phải cao hơn rất nhiều, hoặc nếu dùng Colab thì một phiên làm việc được cung cấp có thể sẽ không đủ để huấn luyện mô hình.

Mặt khác, vì có cùng cấu trúc nên PhoBERT cũng có những nhược điểm tương tự như GPT. Ví dụ rõ hơn, một điểm yếu chí mạng của PhoBERT và các mô hình ngôn ngữ khác dựa trên Transformer là khả năng hiểu biểu đồ từ vựng. Mô hình sử dụng mã hóa từ vựng vào các vector nhúng, và việc xử lý các từ mới nằm ngoài từ vựng đã huấn luyện có thể gặp khó khăn. Vậy nên, mặc dù PhoBERT đã được huấn luyện trên dữ liệu tiếng Việt lớn, đôi khi dự đoán của mô hình vẫn có thể không chính xác hoặc không phù hợp với ngữ cảnh cụ thể. Điều này có thể xảy ra khi dữ liệu đầu vào không phổ biến hoặc không được đại diện đúng mức độ của ngôn ngữ tiếng Việt.

Qua đó, ta cần nhận thức rõ rằng việc huấn luyện và sử dụng mô hình này không phải là một quy trình đơn giản và không phải luôn tỉ lệ thuận với giá trị mà nó mang lại.

4.4 Giải pháp lưu trữ cấu trúc thư mục dạng cây

Như đã đề cập bên trên, hệ thống sẽ hiện thực theo hướng cấu trúc kết hợp, vì vậy một yêu cầu cần được giải quyết là phương pháp lưu trữ thư mục phân cấp. Phương

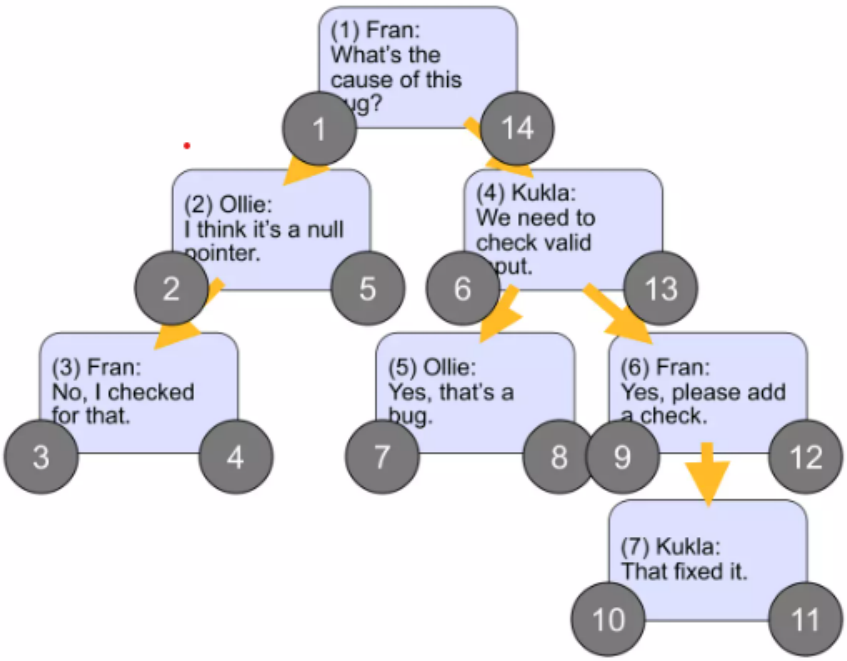
pháp này không chỉ tối ưu trong việc đảm bảo tốc độ truy xuất, mà còn đảm bảo dễ dàng trong việc thêm xóa sửa các node con.

Một phương án tiêu biểu nhất đó chính là danh sách liền kề (Adjacency list)[20]. Cách tiếp cận này được hiện thực bằng cách lưu parent_id trong mỗi node con. Điểm yếu lớn nhất của cách tiếp cận này chính là không xử lý được các cây với chiều sâu lớn vì phải thực hiện nhiều phép JOIN dẫn đến hiệu suất thấp.

Phương án tiếp cận thứ hai là lưu một chuỗi các cha (ancestor) cho các node. Phương án này tương đối cơ bản và sẽ phù hợp cho các ứng dụng muốn lưu trữ breadcrumb nhằm dễ dàng điều hướng. Tuy nhiên, vấn đề lớn nhất mà phương án này gặp phải là khi truy xuất các node con sẽ yêu cầu vét cạn tất cả cột của bảng, dẫn đến hiệu suất thấp.

Phương án thứ ba là tập lồng nhau (nested set). Tập lồng nhau sẽ lưu 2 giá trị trong mỗi node:

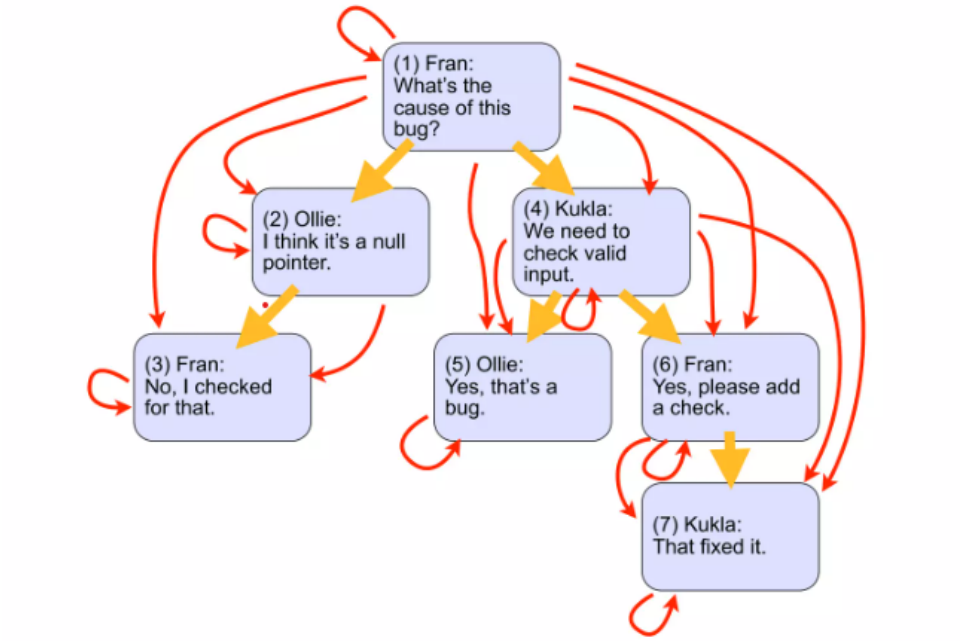
- Giá trị bên trái (left value) phải nhỏ hơn giá trị của tất cả node con.
- Giá trị bên phải (right value) phải lớn hơn giá trị của tất cả node con.
- Miền giá trị trái - phải bắt buộc nằm giữa miền giá trị của tất cả các node cha.



Hình 4.4.1: Cấu trúc tập lồng nhau

Tập lồng nhau giúp hỗ trợ tìm kiếm cấu trúc cây con nhanh chóng, nhưng thao tác tìm kiếm, thêm, xóa hay di chuyển node con đều rất khó khăn.

Phương án cuối cùng là bảng đóng (Closure table). Bảng đóng được hiện thực bằng cách cho mỗi node lưu đường dẫn tới từng node con và đường dẫn tới chính bản thân node đó.



Hình 4.4.2: Cấu trúc bảng đóng

Điểm yếu duy nhất của bảng đóng là phương án này phải lưu trữ 2 bảng, bao gồm bảng các node và bảng quan hệ cha - con giữa các node. Bảng quan hệ cha - con có thể yêu cầu tới $O(n^2)$ cột, tuy nhiên trong thực tế việc lưu trữ có thể ít hơn rất nhiều.

comment_id	author	comment
1	Fran	What's the cause of this bug?
2	Ollie	I think it's a null pointer.
3	Fran	No, I checked for that.
4	Kukla	We need to check valid input.
5	Ollie	Yes, that's a bug.
6	Fran	Yes, please add a check
7	Kukla	That fixed it.

ancestor	descendant
1	1
1	2
1	3
1	4
1	5
1	6
1	7
2	2
2	3
3	3
4	4
4	5
4	6
4	7
5	5
6	6
6	7
7	7

requires O(n²) rows

(but far fewer in practice)

Hình 4.4.3: Độ phức tạp không gian của Closure table

Lưu trữ theo hướng Closure Table mặc dù phải lưu 2 bảng, tuy nhiên việc truy vấn node con, truy vấn cây con, xóa node, thêm node và xóa cây con đều rất dễ thực hiện.

Design	Tables	Query Child	Query Subtree	Delete Node	Insert Node	Move Subtree	Referential Integrity
Adjacency List	1	Easy	Hard	Easy	Easy	Easy	Yes
Path Enumeration	1	Hard	Easy	Easy	Easy	Easy	No
Nested Sets	1	Hard	Easy	Hard	Hard	Hard	No
Closure Table	2	Easy	Easy	Easy	Easy	Easy	Yes

Hình 4.4.4: So sánh độ phức tạp giữa các phương pháp

4.5 Tìm kiếm tài liệu

4.5.1 Metadata

Trong hệ thống quản lý tài liệu, một điều kiện tiên quyết là phải tổ chức quản lý metadata cho các văn bản, thư mục nhằm kết xuất giao diện cho người dùng. Khi người dùng tương tác trên website, người dùng sẽ chỉ tương tác với metdata trong khi nội dung file thật sự sẽ được lưu trữ ở một dịch vụ riêng và chỉ được phục vụ khi người dùng tải xuống.

Việc tổ chức metadata không chỉ giúp giảm tải mà còn tăng khả năng tìm kiếm cho văn bản. Người dùng khi muốn tìm kiếm văn bản thường sẽ nhớ đến các đặc điểm nổi bật của văn bản như: Tên người sở hữu, ngày tạo, các tags,... Vì vậy nên việc hình thành một bản mẫu metadata càng quan trọng hơn.

Sau khi tham khảo các hệ thống như Google Drive, Microsoft SharePoint, Windows File System, cấu trúc file metadata sẽ được mô tả như sau:

- name: Tên của văn bản.
- size: Kích thước văn bản, được tính bằng byte.
- type: Loại văn bản dựa trên đuôi của file.
- created_date: Ngày upload lên hệ thống.
- author: Người upload văn bản lên hệ thống.
- criterions: Mảng các tiêu chí của văn bản như được giải thích bên trên.
- general_permission: Quyền tổng quát của file, quyết định lên trên nhiều người, chẳng hạn như file có quyền đọc cho tất cả mọi người, hoặc hạn chế ghi trên tất cả mọi người.

- permissions: Mảng các quyền cho từng cá nhân, chẳng hạn như người dùng với id = 1 có quyền đọc file.
- current_version: Phiên bản hiện tại của văn bản, chủ yếu nhằm hiện thực chức năng quản lý phiên bản.
- versions: Mảng các phiên bản hiện có của văn bản, mỗi phần tử sẽ lưu số thứ tự phiên bản, mã hash nhằm kiểm tra trùng văn bản và đường dẫn trên storage AWS S3 để có thể truy cập nếu cần thiết.
- star: Trường này nhằm đánh dấu nếu văn bản có được "gắn sao" hay không. Gắn sao là chức năng nhằm hightlight (làm nổi bật) văn bản để dễ truy cập.
- hash_value: Mã băm của văn bản nhằm thực hiện so sánh kiểm tra trùng văn bản trong hệ thống.
- description: Mô tả của văn bản.
- details: Giá trị JSON nhằm lưu các thuộc tính riêng (ngách) của văn bản nếu có. Chẳng hạn như nếu văn bản Hành chính có các thuộc tính như: Ngày ký, tiêu đề, người viết đơn, chữ ký, ... thì các thuộc tính riêng này sẽ được lưu trong trường details của metadata.

Folder metadata của văn bản sẽ khá tương tự như metadata của văn bản. Tuy nhiên ngoài một số trường nên có thì metadata của thư mục sẽ giảm bớt các trường như: type, tags, current_version, versions, hash_value và details bởi vì các trường này như được giải thích bên trên sẽ không cần khi hiện thực thư mục. Cấu trúc folder metadata sẽ được mô tả như sau:

- name: Tên của thư mục.
- size: Kích thước thư mục, được tính bằng byte và bằng tổng các file thuộc thư mục với kích thước các thư mục con.
- created_date: Ngày tạo thư mục.
- author: Người tạo thư mục.
- criterions: Mảng các tiêu chí của thư mục nhằm xác định văn bản thuộc thư mục.
- general_permission: Quyền tổng quát của thư mục, quyết định lên trên nhiều người, chẳng hạn như thư mục có quyền đọc cho tất cả mọi người, hoặc hạn chế chỉnh sửa trên tất cả mọi người.
- permissions: Mảng các quyền cho từng cá nhân, chẳng hạn như người dùng với id = 1 có quyền đọc thư mục.

- star: Trường này nhằm đánh dấu nếu thư mục có được "gắn sao" hay không. Gắn sao là chức năng nhằm hightlight (làm nổi bật) các thư mục được yêu thích.
- description: Mô tả của thư mục.

4.5.2 Giải pháp hiệu quả cho vấn đề tìm kiếm

Việc sử dụng một công cụ tìm kiếm mạnh là hết sức cần thiết trong hệ thống quản lý văn bản nhằm đảm bảo trả về dữ liệu kịp thời. Trong một số trình quản lý văn bản nhu Google Drive hoặc SharePoint, việc tìm kiếm trả về dữ liệu nhanh chóng, giúp trải nghiệm người dùng rất tốt.

Một giải pháp nổi bật cho vấn đề này chính là Elasticsearch (ES) - Một công cụ tìm kiếm dựa trên nền tảng Apache Lucene. Nó cung cấp một bộ máy tìm kiếm dạng phân tán, có đầy đủ công cụ với một giao diện web HTTP có hỗ trợ dữ liệu JSON. Elasticsearch được phát triển bằng Java và được phát hành dạng nguồn mở theo giấy phép Apache[17].

Các đặc điểm của Elasticsearch bao gồm:

- Elasticsearch là một search engine.
- Elasticsearch được kế thừa từ Lucene Apache.
- Elasticsearch thực chất hoạt động như 1 web server, có khả năng tìm kiếm nhanh chóng (near realtime) thông qua giao thức RESTful.
- Elasticsearch có khả năng phân tích và thống kê dữ liệu.
- Elasticsearch chạy trên server riêng và đồng thời giao tiếp thông qua RESTful do vậy nên nó không phụ thuộc vào ngôn ngữ của client hay hệ thống. Từ đó, việc tích hợp Elasticsearch vào bất kì hệ thống nào đều rất dễ dàng, người dùng chỉ cần gửi request http lên là Elasticsearch sẽ trả về kết quả.
- Elasticsearch là 1 hệ thống phân tán và có khả năng mở rộng tuyệt vời (horizontal scalability). Khi được thêm node, Elasticsearch sẽ có cơ chế tự động mở rộng.
- Elasticsearch là dự án mã nguồn mở được phát triển bằng Java.

Elasticsearch được sử dụng bởi nhiều hệ thống lớn: Facebook, Quora, Netflix,... Việc được sử dụng rộng rãi như vậy chứng minh rằng Elasticsearch thật sự là một công cụ mạnh mẽ và hoàn toàn phù hợp với yêu cầu tìm kiếm văn bản. Một số khái niệm cơ bản trong công cụ này là:

- Document: Là một thực thể (object) JSON với một số dữ liệu. Đây là đơn vị thông tin cơ bản trong ES. Hiểu 1 cách cơ bản thì đây là đơn vị nhỏ nhất để lưu trữ dữ liệu trong Elasticsearch.

- Index: Là một khái niệm gần giống như đánh chỉ mục trong MySQL. Tuy nhiên Elasticsearch sử dụng một cấu trúc được gọi là inverted index. Nó được thiết kế để cho phép tìm kiếm toàn văn, một tính năng cực kì mạnh mẽ và là chế độ tìm kiếm nổi bật nhất trong Elasticsearch.

- Shard: Là đối tượng của Lucene, là tập con các Documents của một Index. Một Index có thể được chia thành nhiều Shard. Mỗi node bao gồm nhiều Shard. Chính vì thế Shard mà là đối tượng nhỏ nhất, hoạt động ở mức thấp nhất, đóng vai trò lưu trữ dữ liệu. Hệ thống gần như không bao giờ làm việc trực tiếp với các Shard vì Elasticsearch đã hỗ trợ toàn bộ việc giao tiếp cũng như tự động thay đổi các Shard khi cần thiết. Có 2 loại Shard là: Primary shard và replica shard.

- Node: Là trung tâm hoạt động của Elasticsearch vì Node lưu trữ dữ liệu, tham gia thực hiện đánh chỉ mục của Cluster cũng như thực hiện các thao tác tìm kiếm. Mỗi Node được định danh bằng 1 tên duy nhất.

- Cluster: Chức năng chính của Cluster đó chính là quyết định xem Shards nào được phân bổ cho Node nào và khi nào thì di chuyển các Cluster để cân bằng lại Cluster.

4.6 Lưu trữ văn bản

4.6.1 Tái sử dụng hệ thống thư mục của hệ điều hành (OS)

Tái sử dụng hệ thống thư mục của hệ điều hành chắc hẳn sẽ là lựa chọn xuất hiện đầu tiên của nhiều lập trình viên khi gặp yêu cầu về hiện thực hệ thống file. Thông thường, các hệ thống file của các hệ điều hành lớn như Windows, MacOS, Ubuntu,... đã được phát triển khá hoàn thiện. Vậy đơn giản chúng ta chỉ cần tái sử dụng chúng là đã xem như là hoàn thiện được hệ thống lưu trữ file? Vấn đề không chỉ nằm ở đó, và không phải nằm ở hệ thống file.

Nếu chúng ta hiện thực theo cách tiếp cận này, việc hiện thực hệ thống sẽ tương đối dễ dàng. Chúng ta chỉ cần host một server với hệ điều hành bản thân cảm thấy tối ưu, hiện thực lưu trữ trên hệ thống ấy và gọi tới khi cần. Như vậy thì chỉ cần một máy với dung lượng đủ (đủ theo nhu cầu hệ thống) thì sẽ được xem như là hoàn thành hệ thống file. Tuy nhiên, khả năng cao hệ thống này sẽ không có tính khả dụng (availability) cao, bởi vì chỉ với 1 server chạy dịch vụ lưu trữ file, chẳng hạn gặp những lúc lưu lượng truy cập cao hoặc bị tấn công DOS / DDOS, gần như dịch vụ này sẽ bất khả dụng, và cả hệ thống xem như là vô dụng.

Ngoài tính availability, hệ thống xây dựng như vậy sẽ gặp các vấn đề về bảo mật (security) nếu lập trình viên không xử lý đúng cách hoặc dữ liệu không được mã hóa. Và nếu như có vấn đề gì xảy ra với phần cứng của server, tất cả dữ liệu của hệ thống sẽ coi

như mất.

Tính mở rộng cũng sẽ là vấn đề của cách tiếp cận này, vì khi hết dung lượng, cách giải quyết duy nhất là mở rộng theo chiều dọc, tức là thêm RAM và SSD / HDD vào máy chủ đang chạy.

Chúng ta có thể dễ dàng kết luận rằng cách hiện thực tái sử dụng này chỉ thích hợp cho các hệ thống nhỏ. Vậy với yêu cầu xây dựng hệ thống quản lý văn bản chuẩn doanh nghiệp, cách tiếp cận này là không phù hợp.

4.6.2 Hadoop HDFS

Hadoop HDFS gần như giải quyết được tất cả vấn đề trên, chủ yếu giải quyết dựa trên tinh thần phân tán dữ liệu của hệ thống này. Về cơ bản, mỗi khi có dữ liệu, hệ thống này sẽ thực hiện phân tán (replica) chúng giữa các node (máy tính) trong các rãnh (rack) khác nhau. Mỗi rack sẽ gồm nhiều node, và mỗi cluster sẽ gồm nhiều rack.

Mỗi khi phân tán dữ liệu, hệ thống sẽ tăng tính chịu lỗi (fault tolerant). Nếu dữ liệu trên một node có vấn đề, hệ thống thông qua master node - một node chính nhằm quản lý tất cả node khác – sẽ thực hiện điều hướng tới các node khác được replica dữ liệu ấy, giúp tăng tính availability và back – up dữ liệu. Hadoop HDFS còn sử dụng giao thức Keberos nhằm xác thực người dùng.

Ngoài ra, bằng cách cấp thêm dung lượng cho các node, hoặc thêm node vào cluster, HDFS có thể nhanh chóng mở rộng hệ thống file theo cả chiều ngang và dọc.

Nói chung, Hadoop HDFS sẽ là một lựa chọn tối ưu hơn so với việc tái sử dụng hệ thống file của OS vì HDFS sẽ khá đúng so với chuẩn doanh nghiệp. Tuy nhiên, một vấn đề gặp phải của HDFS là chúng ta sẽ cần nhiều node (máy tính) để hiện thực hệ thống file này.

Để có được nhiều node, chúng ta sẽ cần thuê nhiều tài nguyên khác nhau, ở mỗi tài nguyên có thể chạy thêm nhiều VMs. Việc hiện thực như vậy sẽ tương đối tốn kém về chi phí vận hành cũng như chi phí bảo trì, và sẽ là không phù hợp với các doanh nghiệp tầm nhỏ và trung trong việc duy trì hệ thống HDFS lên tới cả ngàn node.

Vậy ta cần một giải pháp vừa đảm bảo các điểm mạnh trên của HDFS, vừa khả thi về mặt kinh tế cho doanh nghiệp.

4.6.3 AWS S3

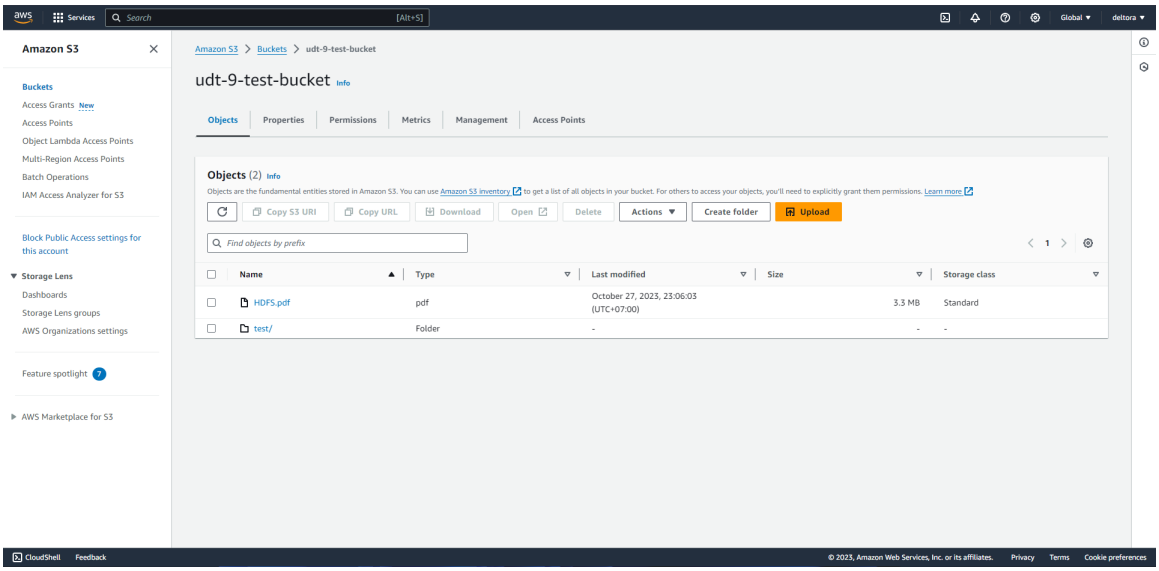
Với AWS S3, ta sẽ không chỉ giải quyết vấn đề của phương án đầu tiên, mà còn giải quyết được triệt để vấn đề của HDFS. S3 lưu trữ dữ liệu theo dạng đối tượng (object

storage), rất dễ mở rộng và cũng hỗ trợ cơ chế phân tán, tức đảm bảo các tính chất đã có của HDFS.

S3 đảm bảo các tính chất cơ bản của một hệ thống mạnh bao gồm: Availability, Security, tính chịu lỗi (Durability) và tính mở rộng (Scalability). Về Availability, S3 đảm bảo ở mức 99.9% đến 99.99%. Về Security, S3 cho phép người dùng cấu hình quyền, khả năng truy cập và mã hóa dữ liệu khi tạo bucket. Về Durability, S3 nổi tiếng với khả năng đảm bảo "11 9's", tức là 99.999999999%. Một ví dụ cụ thể là nếu hệ thống lưu 10000 object trên S3 thì sẽ phải cần 10 triệu năm để có thể xảy ra hư hại dữ liệu trên một object. Cuối cùng về Scalability, như đã trình bày ở trên, S3 được mở rộng theo hướng dọc, tức là thêm dữ liệu chứ không phải thêm node, và AWS đảm bảo khả năng lưu trữ vô tận cho hệ thống, chỉ cần hệ thống của bạn có dữ liệu.

Điểm nổi bật trên hết của S3 so với HDFS trong việc hiện thực hệ thống quản lý văn bản chính là chi phí thấp hơn rất nhiều. S3 chỉ tính chỉ tính chi phí theo không gian lưu trữ mà hệ thống sử dụng, thay vì trả theo cơ sở hạ tầng được thuê để chạy HDFS. Ngoài ra, chúng ta cũng không cần tốn chi phí để bảo trì nhiều máy như HDFS, vì AWS sẽ đảm bảo vấn đề này.

Nên lưu ý rằng, AWS cũng sẽ hỗ trợ lưu trữ theo folder trong từng bucket, và vì vậy hệ thống có thể hiện thực theo hướng cấu trúc cây trong cấu trúc object storage của S3. Bên cạnh đó, S3 hỗ trợ việc truy xuất dữ liệu thông qua REST API và khả năng tích hợp với nhiều dịch vụ khác của AWS chẳng hạn như AWS Lambda, giúp tăng tính mở rộng của hệ thống quản lý sau này.



Hình 4.6.1: Sử dụng Amazon S3

Sau đây là bảng giá lưu trữ một số gói dịch vụ S3. Gói S3 Standard là gói nhóm sẽ sử dụng:

Gói	S3 Standard	S3 Intelligent - Tiering
50 TB đầu tiên / tháng	$0.023 trên GB	$0.023 trên GB
450 TB đầu tiên / tháng	$0.022 trên GB	$0.022 trên GB
500 TB đầu tiên / tháng	$0.021 trên GB	$0.021 trên GB

Bảng 4.6.1: Bảng giá các gói dịch vụ của S3 [18]

4.7 Nhận dạng ký tự quang học (OCR)

4.7.1 Các công cụ OCR hiện nay

Hiện nay trên thị trường tồn tại rất nhiều phần mềm và công cụ OCR miễn phí và trả phí. Các phần mềm này có độ hoàn thiện tương đối tốt và trả ra kết quả với độ chính xác cao. Dưới đây là bảng so sánh nhanh các phần mềm OCR tốt nhất có trên thị trường.

Tên	Thị trường mục tiêu	Số lượng người dùng hỗ trợ	Xếp hạng Capterra (tổng thể)
Nanonets	Vừa và nhỏ	11-200 người dùng	4.9
DocSumo	Vừa và tập trung công ty tài chính	1-10 người dùng	4.6
DocSumo	Vừa và tập trung công ty tài chính	1-10 người dùng	4.6
Rossum	Từ nhỏ đến lớn	2-1000+ người dùng	4.6
Tesseract	Nhỏ và công ty khởi nghiệp	NA	NA
Ephesoft	Công ty tài chính, logistics, sức khỏe và giáo dục	2-1000+ người dùng	4.5
Adobe Acrobat	Vừa và nhỏ	NA	NA
Abbyy	Công ty sức khỏe, ngân hàng	NA	NA
Docparser	Nhỏ và công ty logistics, tài chính, IT	1-1000+ người dùng	4.8

Bảng 4.7.1: Bảng so sánh nhanh các phần mềm và công cụ phổ biến trên thị trường hiện nay [19]

Qua bảng so sánh trên ta có thể thấy được quy mô sử dụng của các phần mềm và công cụ OCR nổi bật nhất hiện nay. Tuy nhiên các phần mềm này đều phải trả phí với một mức giá khá cao (Khoảng $500 trở lên với giới hạn số trang được quét) trừ Tesseract. Tesseract là một phần mềm mã nguồn mở (outsource) và được cung cấp miễn phí. Vì vậy, chúng tôi quyết định sử dụng Tesseract cho hệ thống này.

Tesseract là nền tảng OCR giúp trích xuất văn bản và thông tin từ những hình ảnh hoặc tài liệu không ở định dạng văn bản. Tesseract là một thư viện mã nguồn mở hỗ trợ

chuyển đổi các tài liệu hiện có thành các tệp văn bản đểcó thể dùng để tìm kiếm hoặc chỉnh sửa.

Các tính năng nổi bật của Tesseract:

- Trích xuất văn bản từ hình ảnh bằng tính năng nhận dạng mẫu ký tự sắc nét.
- Tesseract hỗ trợ hơn 100 ngôn ngữ, trong đó có Tiếng Việt và đã được hỗ trợ từ các phiên bản đầu tiên.
- Có thể tự đào tạo Tesseract OCR bằng cách chạy hàng trăm, hàng nghìn ví dụ (hình ảnh và tài liệu) để huấn luyện.

Tesseract có những ưu điểm hơn so với các OCR mã nguồn mở khác vì khả năng phân tích và trích xuất có độ chính xác cao. Một trong những điểm mạnh của nó là khả năng tương thích với nhiều ngôn ngữ và framework. Trên hết Tesseract là một mã nguồn mở được cung cấp miễn phí phù hợp với mô hình nhỏ. Mô hình cơ sở của Tesseract có thể cho ra kết quả kém chính xác, nhưng điều này có thể cải thiện nó bằng cách huấn luyện và điều chỉnh các tham số.

Bên cạnh những ưu điểm, Tesseract cũng có nhược điểm. Khi làm việc với dữ liệu dạng bảng, Tesseract có thể gặp khó khăn vì thiếu ngữ cảnh hoặc tạo ra các từ ngẫu nhiên, vì vậy cần thực hiện tiền xử lý để đảm bảo tính chính xác của kết quả trả ra. Thiết lập metadata để xử lý các bảng thành từng vùng nhất định sẽ giúp cho Tesseract dễ dàng trích xuất hơn và cũng góp phần giảm thiểu sai số. Ngoài ra, nếu so với các sản phẩm thương mại, độ chính xác của Tesseract sẽ không mạnh bằng và có độ sai lệch nhiều hơn. Một trong những thách thức lớn nhất trong việc triển khai phần mềm OCR là ngôn ngữ. Mặc dù phần mềm có thể được đào tạo và tùy chỉnh để nhận dạng nhiều ngôn ngữ nhưng phần mềm vẫn gặp vấn đề khi nói đến một số ngôn ngữ nhất định, đặc biệt là các ngôn ngữ có dấu (như tiếng Việt), những ngôn ngữ tượng hình (như tiếng Trung) hay những ngôn ngữ đặc biệt như tiếng Ả Rập phải đọc từ phải sang trái, những ngôn ngữ này khi chạy đều có thể gây ra sự cố.

Phiên bản được áp dụng vào hệ thống là phiên bản mới nhất của Tesseract là phiên bản 4.0. Nó cung cấp tích hợp thêm trí tuệ nhân tạo (Artificial Intelligence, viết tắt là AI) thông qua Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), điều này hỗ trợ phát hiện, nhận biết đầu vào và trả ra kết quả chính xác hơn[19].

4.7.2 Phương án tiền xử lý OCR

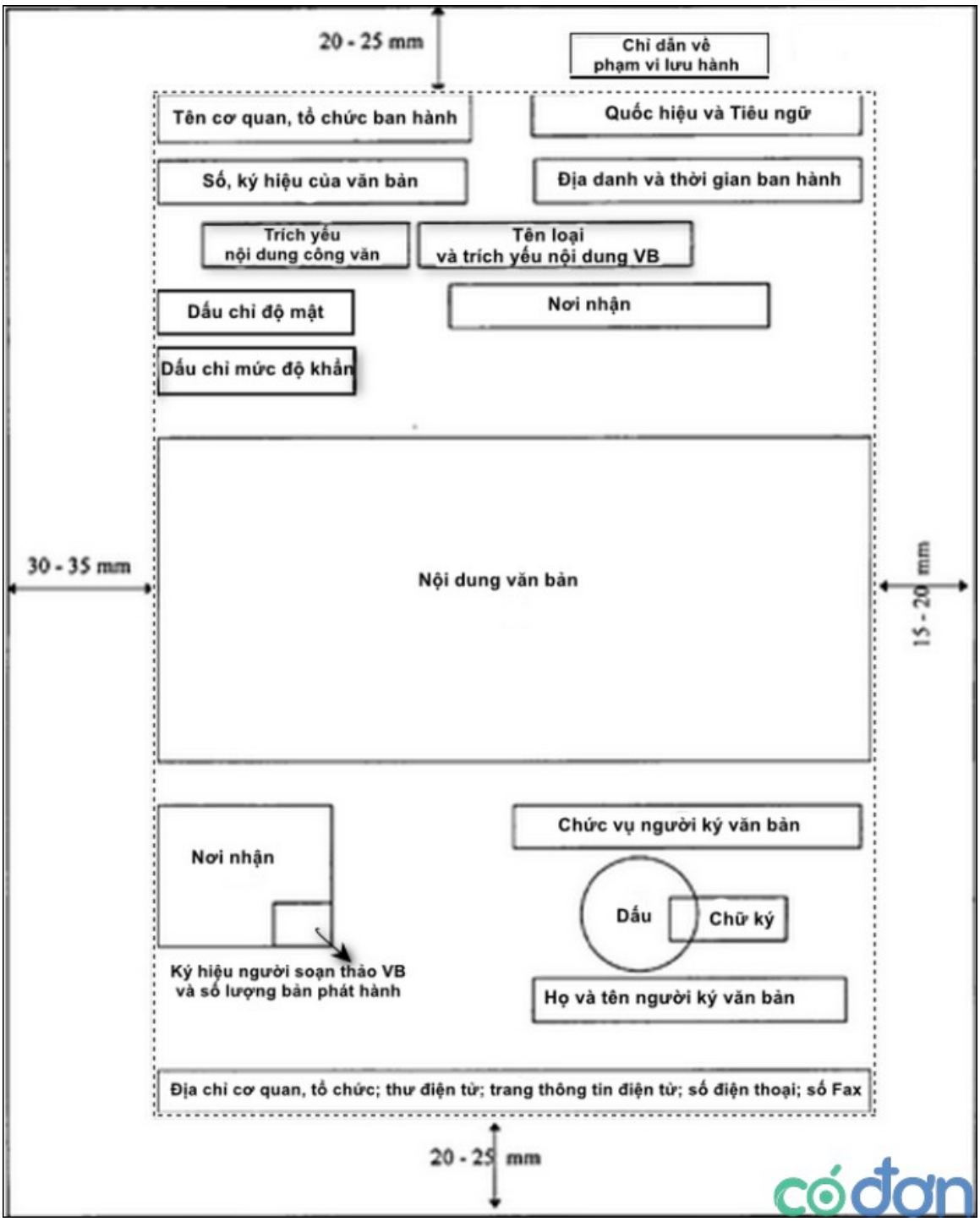
Một văn bản sau khi được quét, nếu như chỉ đơn thuần sử dụng OCR và lưu kết quả nhận được thì kết quả sẽ thường trả ra không được như mong muốn. Thông thường các tài liệu sẽ không có dạng văn bản liền mạch từ trên xuống, mà sẽ có dạng bảng phân

chia thành nhiều vùng (field) khác nhau trong cùng một hàng, đặc biệt là ở các trang bìa, trang tiêu đề, trang đầu và một số dạng tài liệu đặc biệt như chứng minh nhân dân (ID Card) hay sổ hộ chiếu. Do vậy các phần mềm OCR không thể cho ra kết quả một cách chính xác, điều này làm ảnh hưởng rất lớn tới chức năng tìm kiếm của EMDS. Vì vậy cần phải có giải pháp để xử lý vấn đề trên.

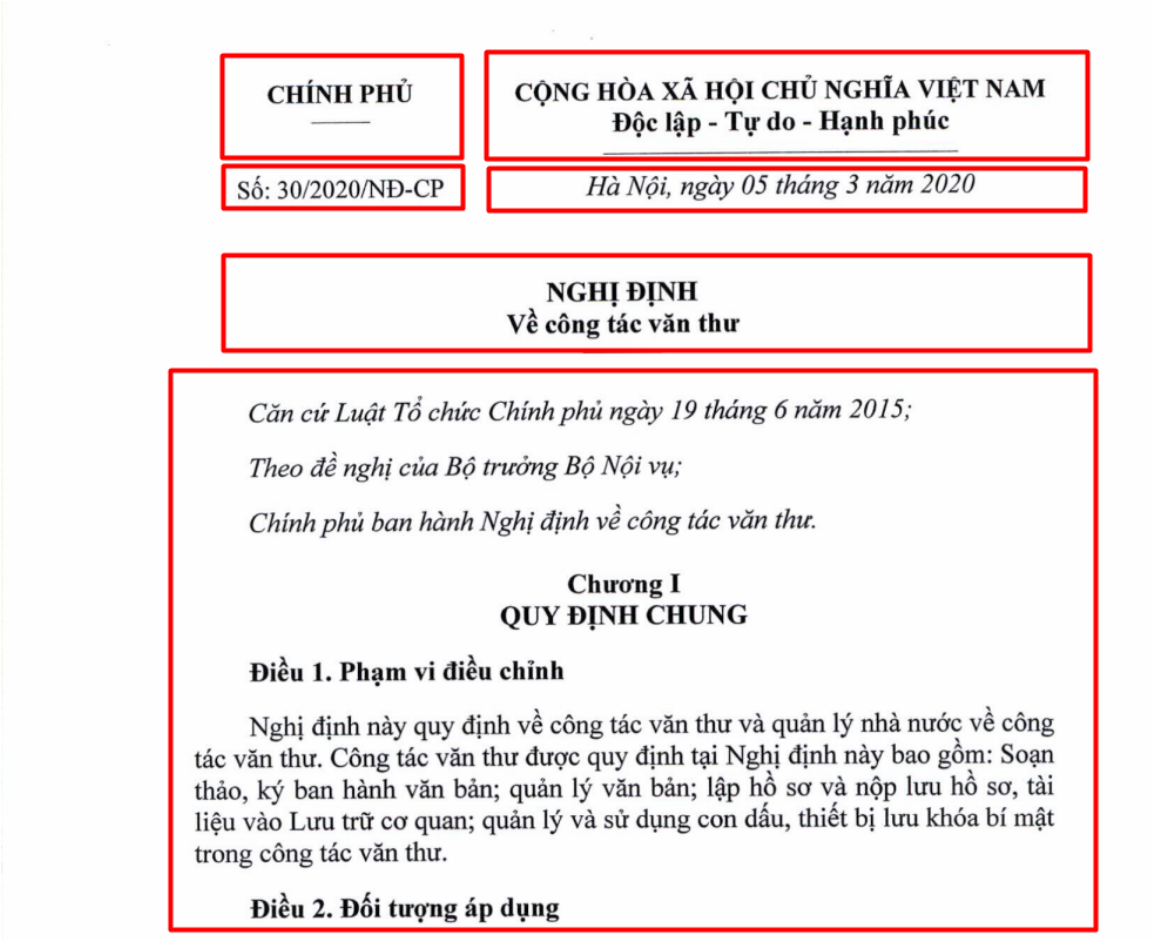
Nhìn chung có ba cách xử lý các văn bản dạng bảng như đã nêu ở trên:

1. **Sử dụng các mẫu (template) tương ứng với văn bản:** Điểm mạnh của phương pháp này là dễ thực hiện và phù hợp với các loại văn bản có một khuôn mẫu được quy định, đặc biệt với các loại có khuôn mẫu cố định giống nhau như chứng minh nhân dân hay sổ hộ chiếu. Tuy nhiên, đối với những loại không theo khuôn mẫu cố định thì sẽ có nhiều biến số gây sai lệch kết quả được trả ra. Ngoại trừ những khuôn mẫu tiêu biểu như phiếu chuyển tiền của ngân hàng, chứng minh nhân dân hay sổ hộ chiếu thì cũng tồn tại ít loại văn bản có một khuôn mẫu được quy định sử dụng rộng rãi. Các bước thực hiện phương án này:

    - **Bước 1:** Xác định loại văn bản cần được xử lý. Ví dụ: văn bản cần được xử lý là văn bản hành chính.
    - **Bước 2:** Chia hình ảnh thành những vùng theo template của loại văn bản được xử lý. Ví dụ: văn bản hành chính sẽ có template như sau.



Hình 4.7.1: Cách ghi số hiệu và ký hiệu văn bản đúng vị trí theo Nghị Định 30/2020/NĐ-CP



CHÍNH PHỦ

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Số: 30/2020/NĐ-CP

Hà Nội, ngày 05 tháng 3 năm 2020

NGHỊ ĐỊNH
Về công tác văn thư

Căn cứ Luật Tổ chức Chính phủ ngày 19 tháng 6 năm 2015;

Theo đề nghị của Bộ trưởng Bộ Nội vụ;

Chính phủ ban hành Nghị định về công tác văn thư.

Chương I
QUY ĐỊNH CHUNG

Điều 1. Phạm vi điều chỉnh

Nghị định này quy định về công tác văn thư và quản lý nhà nước về công tác văn thư. Công tác văn thư được quy định tại Nghị định này bao gồm: Soạn thảo, ký ban hành văn bản; quản lý văn bản; lập hồ sơ và nộp lưu hồ sơ, tài liệu vào Lưu trữ cơ quan; quản lý và sử dụng con dấu, thiết bị lưu khóa bí mật trong công tác văn thư.

Điều 2. Đối tượng áp dụng

Hình 4.7.2: **Ví dụ phân chia một văn bản thuộc kiểu văn bản hành chính**

Hình 4.7.3: **Các biến số gây sai lệch kết quả trả ra. Cùng một vùng nhưng độ dài văn bản khác nhau dẫn đến sai lệch**

- **Bước 3:** Tiến hành chạy OCR trên từng vùng đã quy định chia và lưu nó thành các biến riêng để thuận lợi cho việc tìm kiếm.

2. **Sử dụng biểu thức chính quy (Regular Expression, viết tắt là RegEx):** Điểm mạnh của phương pháp này là Khắc phục được điểm yếu của phương pháp sử dụng khuôn mẫu vì không còn phụ thuộc vào một khuôn cố định nữa. Thông qua đó còn có thể thêm những trường tùy theo mong muốn và nhu cầu mà không bị giới hạn. Tuy nhiên, vì mỗi biểu thức chính quy cũng đại diện cho mỗi loại văn bản khác nhau, do đó, phải xác định được mỗi loại văn bản sẽ có những trường nào và áp dụng biểu thức nào. Điều này làm giới hạn số tài liệu có thể quy định. Ngoài

ra, kết quả trả ra cũng phụ thuộc vào chất lượng của biểu thức chính quy được xây dựng. Các bước thực hiện phương án này:

- **Bước 1:** Xác định loại văn bản cần được xử lý. Ví dụ: văn bản cần được xử lý là văn bản hành chính.
- **Bước 2:** Chạy OCR trên tài liệu và trả về kết quả là một chuỗi liền nhau.
- **Bước 3:** Xác định các trường của tài liệu dựa trên công thức biểu thức chính quy tương ứng.

3. **Sử dụng phương pháp phân tích ngôn ngữ tự nhiên:** Phương pháp này sẽ sử dụng phân tích ngôn ngữ tự nhiên áp dụng học sâu (Deeplearning) để phân tích kết quả OCR của văn bản, qua đó có thể phân loại và cho ra được các kết quả phù hợp. Việc không phụ thuộc vào khuôn mẫu hay quy định nào giúp cho phương pháp này sẽ không bị giới hạn ở những miền nhất định nào. Tuy nhiên, việc có thể phân tích được một loại văn bản đòi hỏi rất nhiều thời gian để có thể xây dựng và luyện cho mô hình (model) có thể hiểu được các loại văn bản, trong quá trình sử dụng cũng sẽ tốn rất nhiều thời gian để quét và cho ra kết quả.

Vì phương pháp phân tích ngôn ngữ tự nhiên cho toàn văn bản vượt quá scope của dự án nên nhóm quyết định không sử dụng phương pháp này cho hệ thống. Với hai phương pháp còn lại, chúng tôi quyết định sẽ sử dụng kết hợp phương pháp sử dụng khuôn mẫu với phương pháp sử dụng biểu thức chính quy để có thể phát huy điểm mạnh của cả hai phương pháp.

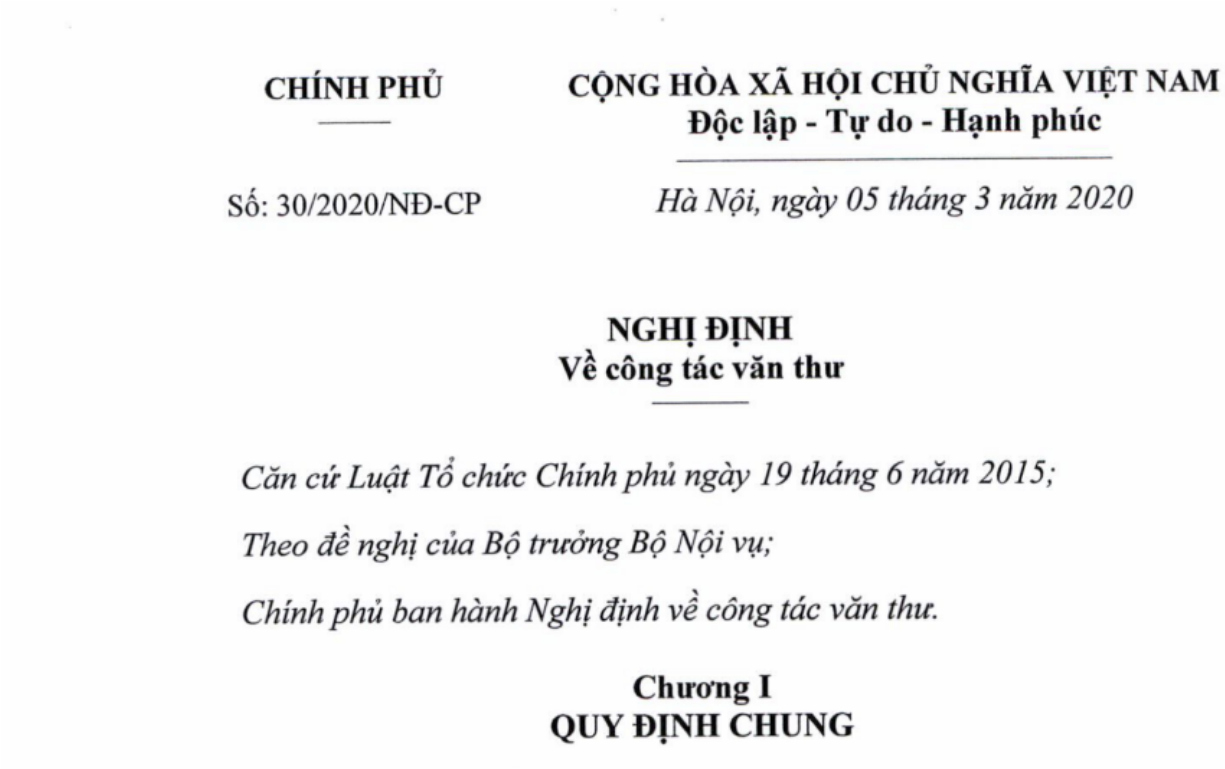
Với phương pháp sử dụng khuôn mẫu, dùng để chia ra những vùng rộng, đặc biệt với dạng bảng chia thành nhiều cột, nếu không thực hiện chia vùng sẽ gây ra sai lệch kết quả trả ra.

Với phương pháp sử dụng biểu thức chính quy, dùng để khắc phục các điểm yếu của phương pháp sử dụng khuôn mẫu.

Vì hệ thống của dự án được xây dựng chủ yếu hướng tới người dùng Việt Nam, nên các phần mềm OCR và biểu thức chính quy đều phải hiện thực trên tiếng Việt. Việc tạo biểu thức chính quy trên tiếng Việt đòi hỏi phải thực hiện tiền xử lý ngôn ngữ khá nhiều để có thể chuẩn hóa lại dấu câu, bảng mã gõ,... Vì vậy nhóm quyết định tiền xử lý văn bản thành chữ viết thường để hạn chế sự sai lệch kết quả khi tìm kiếm bằng biểu thức chính quy.

4.7.3 Thí nghiệm hoạt động của OCR

1. **Chỉ sử dụng phương pháp biểu thức chính quy**



CHÍNH PHỦ

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Số: 30/2020/NĐ-CP

Hà Nội, ngày 05 tháng 3 năm 2020

NGHỊ ĐỊNH
Về công tác văn thư

Căn cứ Luật Tổ chức Chính phủ ngày 19 tháng 6 năm 2015;

Theo đề nghị của Bộ trưởng Bộ Nội vụ;

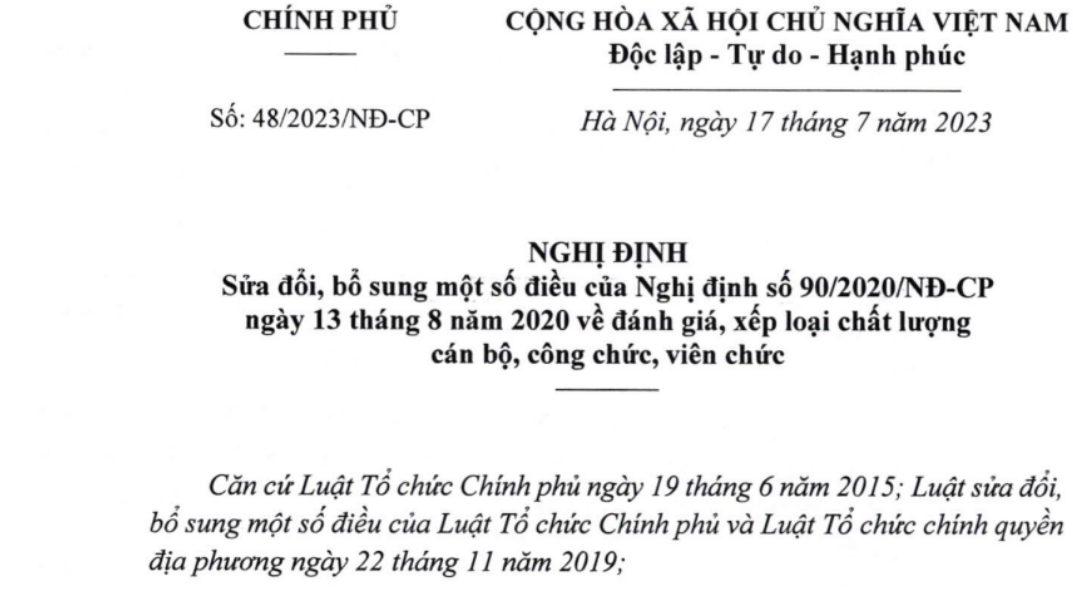
Chính phủ ban hành Nghị định về công tác văn thư.

Chương I
QUY ĐỊNH CHUNG

Hình 4.7.4: Văn bản Nghị định số 30/2020/NĐ-CP

```
{
co_quan : chính phủ,
quoc_hieu : cộng hòa xã hội chủ nghĩa việt nam,
tieu_ngu : độc lập - tự do - hạnh phúc,
so : 30/2020/nđ-cp,
dia_diem : hà nội,
ngay_thang : ngày 05 tháng 3 năm 2020,
loai_van_ban : nghị định,
tieu_de : về công tác văn thư,
noi_dung : căn cứ luật tả chức chính phủ ngày 19 thẳng 6 năm 2015;
theo đề nghị của bộ trưởng bộ nội vụ;
chính phủ ban hành nghị định về công tác văn thư.
chương i
```

Hình 4.7.5: Kết quả OCR của Nghị định số 30/2020/NĐ-CP



CHÍNH PHỦ

Số: 48/2023/NĐ-CP

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

Hà Nội, ngày 17 tháng 7 năm 2023

NGHỊ ĐỊNH
Sửa đổi, bổ sung một số điều của Nghị định số 90/2020/NĐ-CP ngày 13 tháng 8 năm 2020 về đánh giá, xếp loại chất lượng cán bộ, công chức, viên chức

Căn cứ Luật Tổ chức Chính phủ ngày 19 tháng 6 năm 2015; Luật sửa đổi, bổ sung một số điều của Luật Tổ chức Chính phủ và Luật Tổ chức chính quyền địa phương ngày 22 tháng 11 năm 2019;

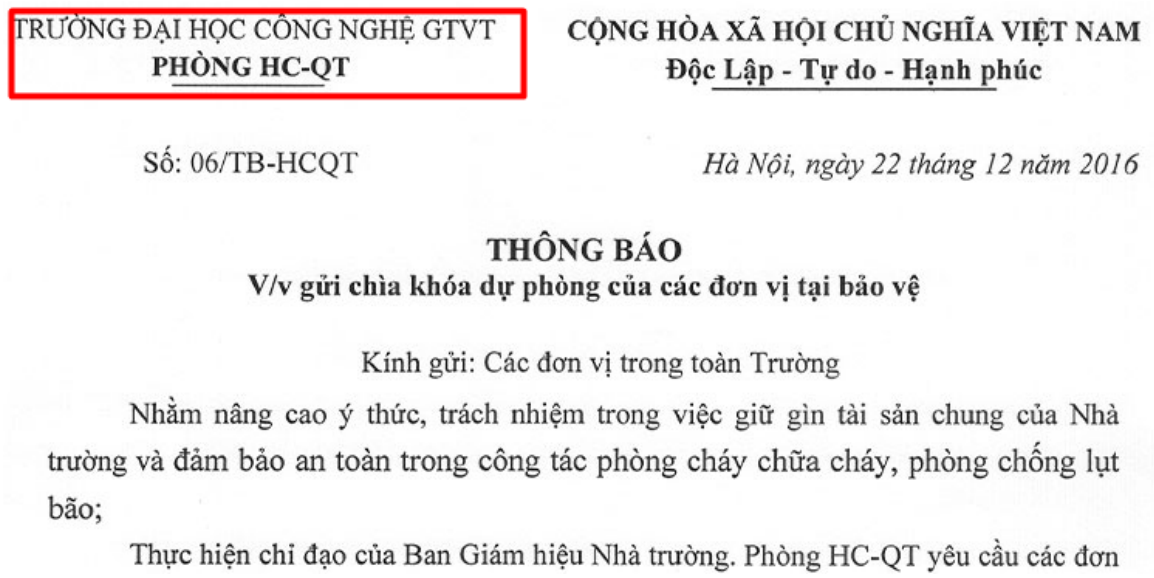
Hình 4.7.6: Văn bản Nghị định số 48/2023/NĐ-CP

```
{
co_quan : chính phủ,
quoc_hieu : cộng hòa xã hội chủ nghĩa việt nam,
tieu_ngu : độc lập - tự do - hạnh phúc,
so : 48/2023/nđ-cp,
dia_diem : hà nội,
ngay_thang : ngày 17 tháng 7 năm 2023,
loai_van_ban : nghị định,
tieu_de : sửa đổi, bổ sung một số điều của nghị định số 90/2020/nđ-cp ngày 13 tháng 8
năm 2020 về đánh giá, xếp loại chất lượng cán bộ, công chức, viên chức,
noi_dung : căn cứ luật tổ chức chính phủ ngày 19 tháng 6 năm 2015; luật sửa đổi,
bổ sung một số điều của luật tổ chức chính phủ và luật tổ chức chính quyền
```

Hình 4.7.7: Kết quả OCR của Nghị định số 48/2023/NĐ-CP

Cả hai Nghị định đều trả ra kết quả chính xác. Qua đó có thể thấy được sử dụng biểu thức chính quy đã khắc phục được nhược điểm đã nêu trên của phương pháp sử dụng khuôn mẫu.

2. **Sử dụng phương pháp khuôn mẫu kết hợp với biểu thức chính quy**



TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GTVT
PHÒNG HC-QT

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự do - Hạnh phúc

Số: 06/TB-HCQT

Hà Nội, ngày 22 tháng 12 năm 2016

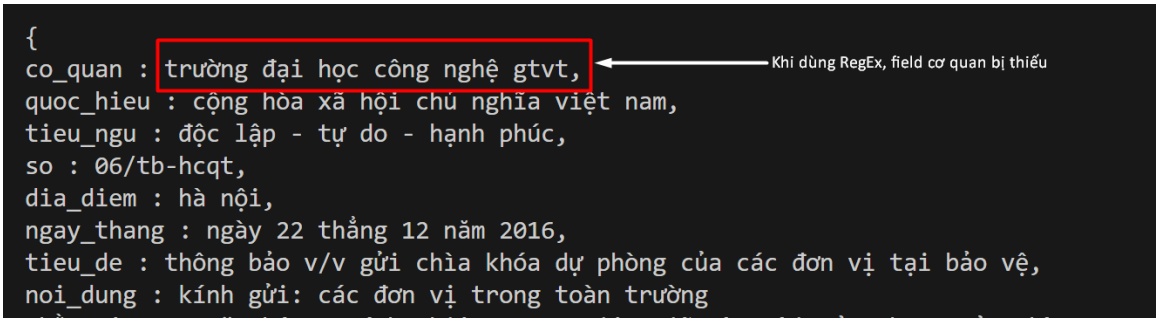
THÔNG BÁO
V/v gửi chìa khóa dự phòng của các đơn vị tại bảo vệ

Kính gửi: Các đơn vị trong toàn Trường

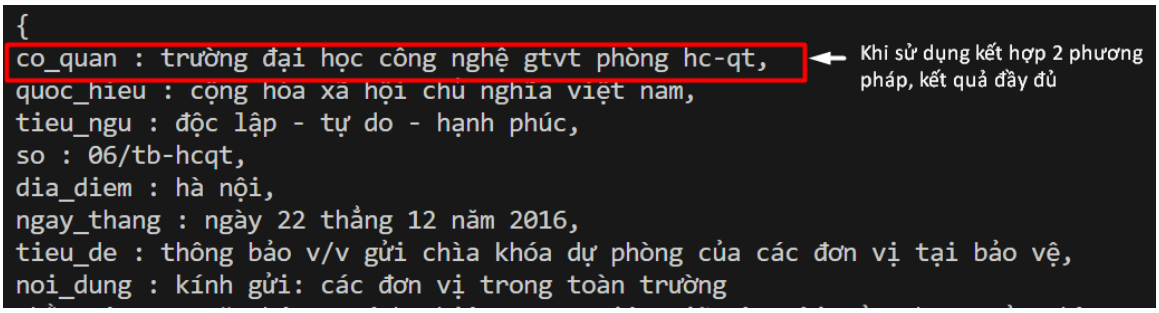
Nhằm nâng cao ý thức, trách nhiệm trong việc giữ gìn tài sản chung của Nhà trường và đảm bảo an toàn trong công tác phòng cháy chữa cháy, phòng chống lụt bão;

Thực hiện chỉ đạo của Ban Giám hiệu Nhà trường. Phòng HC-QT yêu cầu các đơn

Hình 4.7.8: Văn bản Thông báo của Trường Đại học Công nghệ - Giao thông vận tải



Hình 4.7.9: Kết quả khi chỉ sử dụng biểu thức chính quy



Hình 4.7.10: Kết quả sử dụng biểu thức chính quy và khuôn mẫu

Kết quả cho ra khi sử dụng kết hợp cả hai phương pháp chính xác hơn khi chỉ dùng biểu thức chính quy.

3. **Kết luận:** Sử dụng kết hợp hai phương pháp biểu thức chính quy và khuôn mẫu cho ra kết quả tốt hơn nếu chỉ sử dụng một trong hai. Hệ thống sẽ tập chung xây dựng khuôn mẫu và biểu thức chính quy cho hai miền văn bản hành chính và sách đã nêu ở phần trên để có thể cho ra kết quả chính xác nhất.

4.8 Kết luận

Về cấu trúc dữ liệu, phương pháp kết hợp đã tận dụng được các điểm mạnh của cấu trúc Hierarchical và Flat, đồng thời phát huy được giá trị của miền hành chính và thư viện sách mà đề tài hướng tới.

Về phương pháp đề xuất tiêu chí, Langchain và GPT đã thể hiện được ưu điểm vượt trội so với phương án sử dụng PhoBERT cả về mặt tầm vực, thời gian và dữ liệu của đề tài.

Về các phương pháp lưu trữ thư mục phân cấp, Closure Table có thể được xem như là phù hợp nhất vì tính đơn giản cho các tác vụ tạo, xóa node, tương ứng với bài toán tạo và xóa thư mục trong hệ thống.

Về phương pháp cho vấn đề tìm kiếm, Elasticsearch với khả năng truy xuất mạnh mẽ có thể xem như phù hợp cho yêu cầu của hệ thống.

Về nhận dạng ký tự quang học OCR, Tesseract là một công cụ đủ mạnh mẽ và phù hợp để áp dụng vào hệ thống. Kết quả cho ra tương đối chính xác và sẽ phát triển sau khi được huấn luyện qua tập dữ liệu lớn hơn. Để hỗ trợ công cụ trả ra kết quả tốt hơn, hệ thống sẽ sử dụng kết hợp hai phương pháp sử dụng khuôn mẫu và sử dụng biểu thức chính quy để tiền xử lý văn bản.

Về các phương pháp lưu trữ văn bản, với 3 cách tiếp cận như trên cùng với điểm mạnh-yếu của chúng, chúng ta dễ dàng kết luận với quy mô hiện thực quản lý văn bản cho một doanh nghiệp vừa, nhỏ, AWS S3 sẽ là lựa chọn tối ưu nhất không chỉ vì những tính chất nổi bật về kỹ thuật, mà còn về khả năng giảm thiểu chi phí của nó.

5 Phát triển hệ thống

5.1 Phân tích yêu cầu

5.1.1 Yêu cầu chức năng

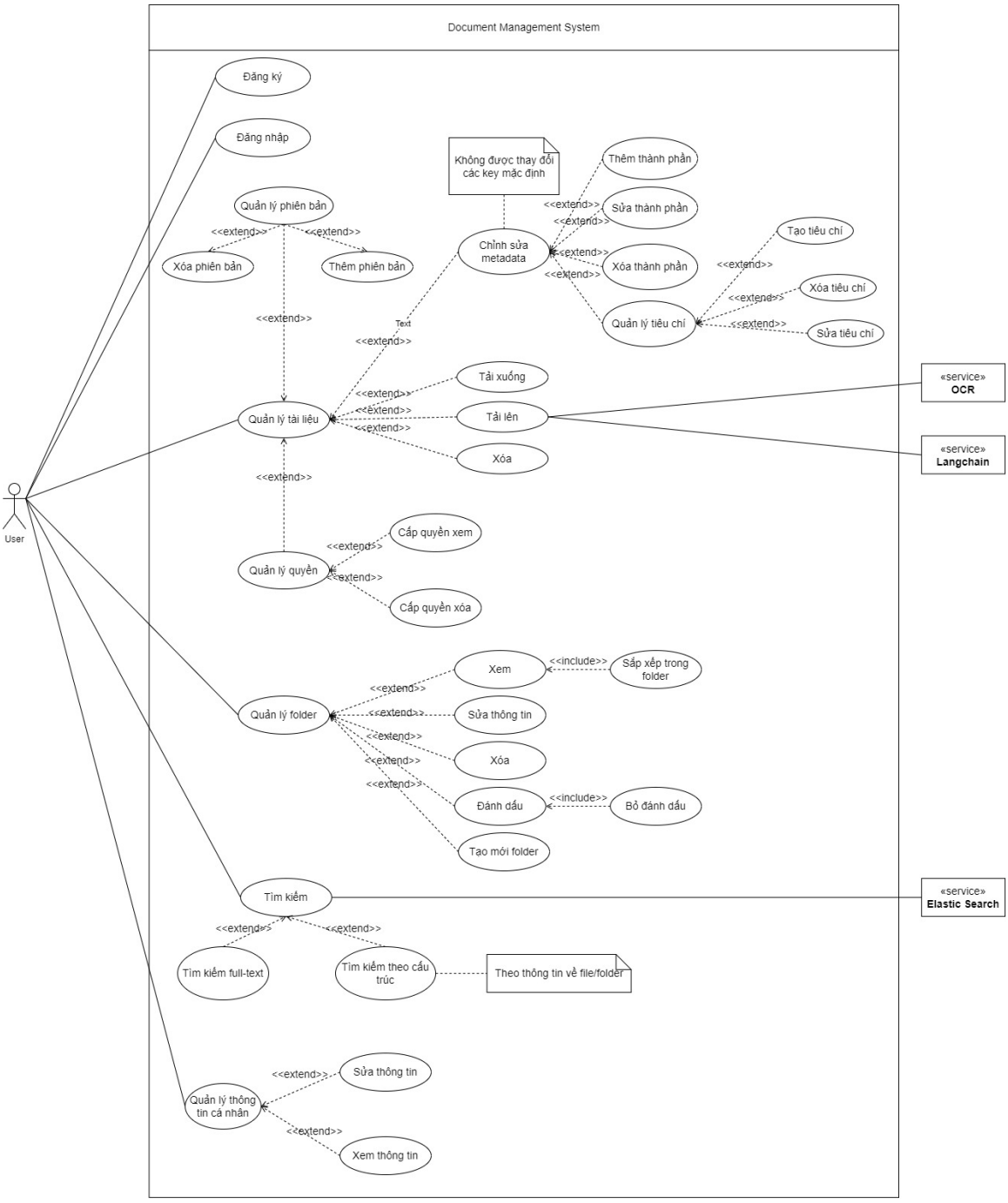
- Chức năng xem, xóa văn bản: Người dùng có thể xem hoặc xóa các văn bản mà mình có quyền.
- Chức năng thêm văn bản: Người dùng có thể thêm văn bản tại bất kỳ thư mục nào. Văn bản sẽ được quét OCR và trích xuất đề xuất tiêu chí từ Langchain. Người dùng sẽ phải bắt buộc tạo hoặc chọn tiêu chí đề xuất cho văn bản trong giai đoạn này.
- Chức năng quản lý phiên bản văn bản: Người dùng có thể dễ dạng thêm hoặc xóa các phiên bản khác nhau của văn bản.
- Chức năng quản lý tiêu chí cho văn bản: Người dùng có thể tạo, xóa và sửa tiêu chí của văn bản.
- Chức năng chia sẻ văn bản: Để có thể dễ dàng cộng tác hơn trong doanh nghiệp, các văn bản có thể được chia sẻ với nhau giữa những người có quyền thao tác trên văn bản.
- Chức năng tải xuống văn bản: Văn bản dễ dàng được tải xuống theo yêu cầu người dùng.
- Chức năng tạo thư mục: Người dùng có thể thêm thư mục tại bất kỳ vị trí nào, tuy nhiên khi thêm thì người dùng sẽ phải bắt buộc chọn tiêu chí cho thư mục này.
- Chức năng xem, xóa thư mục: Người dùng có thể xem hoặc xóa các thư mục mà mình có quyền.
- Chức năng đánh dấu thư mục: Người dùng có thể đánh dấu sao các thư mục quan trọng.
- Chức năng chia sẻ thư mục: Để có thể dễ dàng cộng tác hơn trong doanh nghiệp, các thư mục có thể được chia sẻ với nhau giữa những người có quyền thao tác trên thư mục.
- Chức năng đăng nhập: Người dùng có thể đăng nhập vào tài khoản của bản thân.
- Chức năng đăng ký: Người dùng có thể đăng ký mới tài khoản.
- Chức năng tìm kiếm: Người dùng có thể tìm kiếm văn bản thông qua các thông tin được nhập vào. Ngoài ra hệ thống sẽ tìm kiếm các văn bản khác liên quan đến từ khóa.
- Quản lý thông tin cá nhân: Người dùng có thể xem và chỉnh sửa thông tin của bản thân.

5.1.2 Yêu cầu phi chức năng

- Hệ thống tối giản các thao tác người dùng cần hiện thực để sử dụng một chức năng. Giao diện thân thiện với người dùng, màu sắc được phối hợp hài hòa, làm nổi bật thư mục và văn bản. Mỗi người dùng có thể dễ dàng sử dụng ứng dụng sau 5 phút làm quen.
- Hệ thống có thể cung cấp dịch vụ cho 100 người dùng cùng lúc. Ngoài ra, hệ thống có thể đáp ứng nhu cầu lưu trữ cho tối thiểu 1000 người dùng.
- Hệ thống có thể sử dụng hiệu quả trên điện thoại di động (Android, IOS), máy tính bảng hay máy tính bàn, laptop (Windows, Linux, Mac) với các trình duyệt (Chrome, Firefox, Safari, Opera).
- Mỗi lần nâng cấp, bảo trì hệ thống định kỳ (theo quý 3 tháng) không mất quá 30 phút. Thời gian khởi động lại hệ thống không quá 1 phút.
- Hệ thống sẽ hỗ trợ cả tiếng Anh và tiếng Việt.

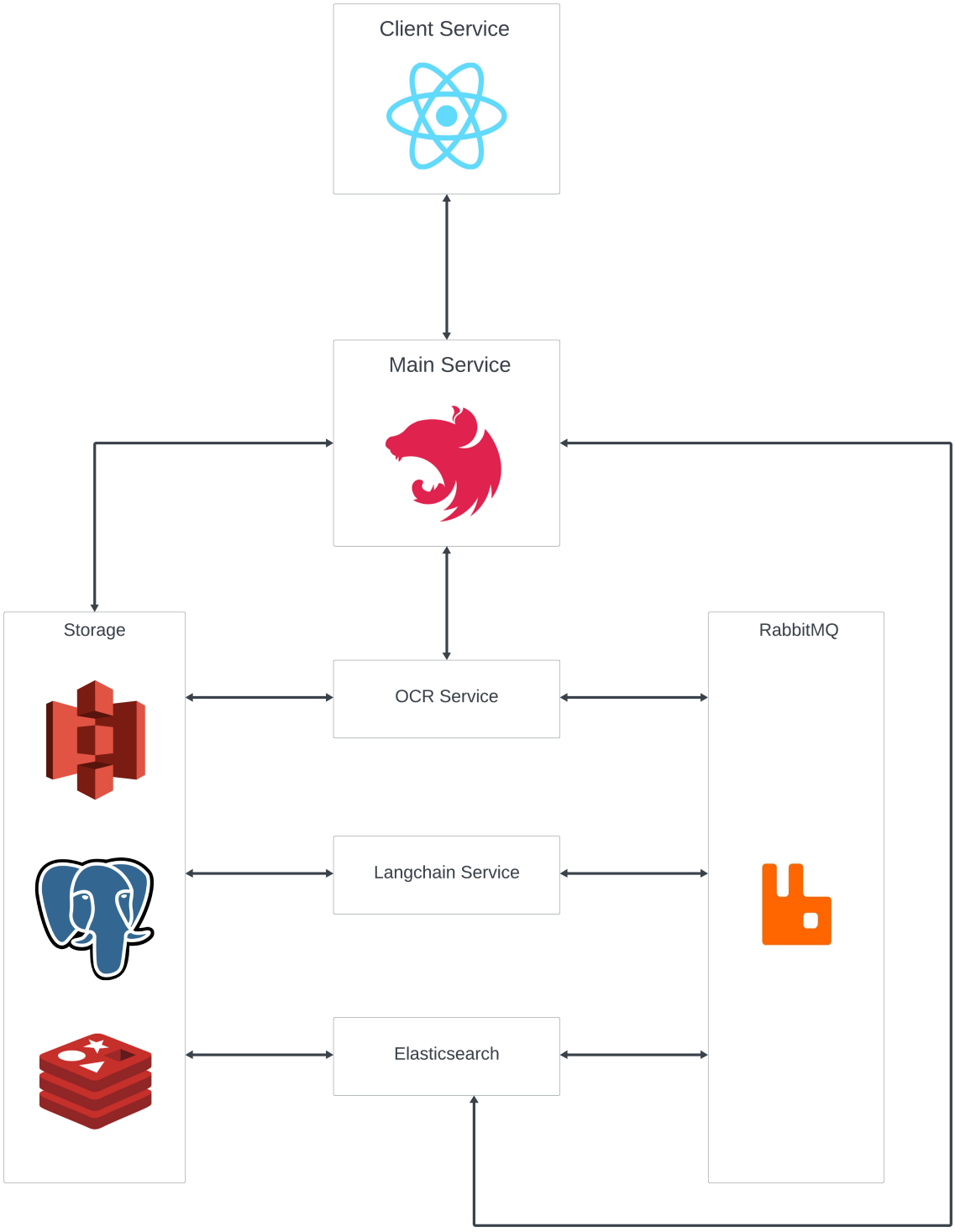
5.2 Thiết kế hệ thống

5.2.1 Lược đồ use-case



Hình 5.2.1: Usecase tổng quát của hệ thống

5.2.2 Kiến trúc hệ thống



Hình 5.2.2: Kiến trúc tổng quan của hệ thống

Giải thích kiến trúc hệ thống:

- Client Service: Dịch vụ nhằm cung cấp giao diện (Frontend) sử dụng chức năng hệ thống cho người dùng. Dịch vụ này sủ dụng Framework ReactJS.
- Main Service: Dịch vụ trung tâm (Backend) nhằm xử lý các yêu cầu từ người dùng và kết nối với các dịch vụ khác.

- Storage: Nơi lưu trữ dữ liệu của cả hệ thống. Tại đây bao gồm 3 dịch vụ khác nhau: Dịch vụ lưu trữ file, dịch vụ lưu trữ thông tin hệ thống và dịch vụ caching.
- RabbitMQ: Dịch vụ Microservices của toàn hệ thống. Các dịch vụ khác sẽ giao tiếp với nhau qua hệ thống hàng đợi của RabbitMQ.
- OCR Service: Dịch vụ xử lý ký tự quang học nhằm trích xuất thông tin từ hình ảnh.
- Langchain service: Dịch vụ trích xuất tiêu chí từ văn bản.
- Elasticsearch: Dịch vụ tìm kiếm của hệ thống.

5.3 Công nghệ sử dụng

5.3.1 ReactJS

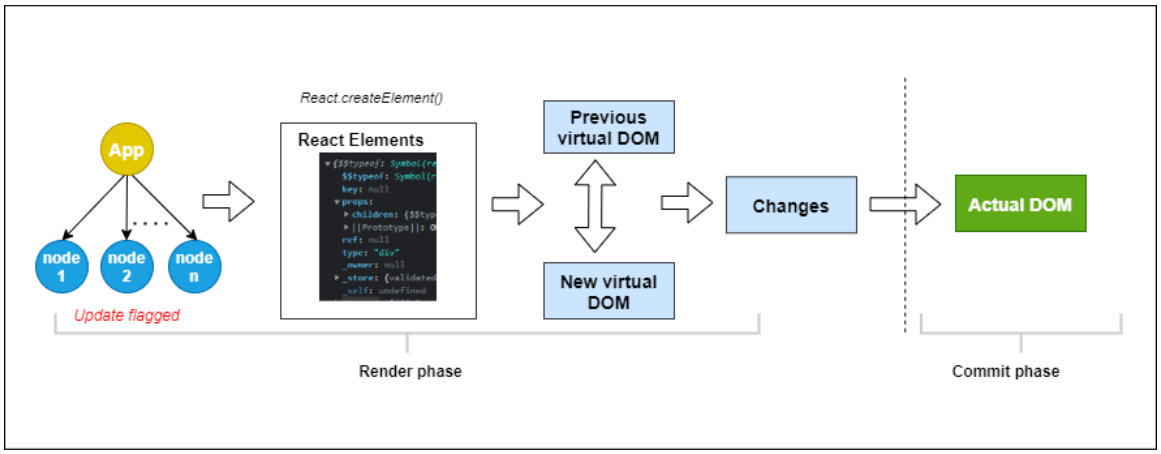
ReactJS là một thư viện JavaScript mã nguồn mở được Facebook xây dựng và phát triển. Thư viện này được sử dụng để tạo ra các trang web hấp dẫn với hiệu quả cao, tốc độ tải nhanh và tối giản mã nguồn. Mỗi trang web sử dụng ReactJS đảm bảo hiệu suất và khả năng mở rộng cao, thao tác thực hiện đơn giản[21].

ReactJS có nhiều tính năng hữu ích cho việc phát triển ứng dụng web, bao gồm:

- Components: ReactJS cho phép phát triển ứng dụng web theo mô hình component. Các component là các phần tử UI độc lập có thể được tái sử dụng trong nhiều phần khác nhau của ứng dụng.
- Virtual DOM: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM là một bản sao của DOM được lưu trữ trong bộ nhớ và được cập nhật một cách nhanh chóng khi có thay đổi, giúp tăng tốc độ và hiệu suất của ứng dụng.
- JSX: JSX là một ngôn ngữ lập trình phân biệt được sử dụng trong ReactJS để mô tả các thành phần UI. JSX kết hợp HTML và JavaScript, giúp cho việc viết mã dễ hiểu và dễ bảo trì hơn.
- State và Props: ReactJS cho phép quản lý trạng thái của các thành phần UI thông qua State và Props. State là trạng thái của một thành phần được quản lý bởi nó chính, trong khi Props là các giá trị được truyền vào từ bên ngoài để tùy chỉnh hoặc điều khiển hành vi của một thành phần.
- Hỗ trợ tốt cho SEO: ReactJS hỗ trợ tốt cho việc tối ưu hóa SEO. Với các thư viện như React Helmet, các nhà phát triển có thể quản lý các phần tử meta và title cho từng trang web, giúp tăng khả năng tìm kiếm và tăng cường trải nghiệm người dùng.

- Hỗ trợ đa nền tảng: ReactJS không chỉ được sử dụng để phát triển ứng dụng web, mà còn được sử dụng để phát triển ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

- Redux: Redux là một thư viện quản lý trạng thái cho các ứng dụng ReactJS. Nó giúp quản lý trạng thái của ứng dụng một cách chính xác và dễ dàng, đồng thời giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.



Hình 5.3.1: Cách kết xuất (render) của ReactJS

5.3.2 NestJS

NestJS là một framework mã nguồn mở để phát triển ứng dụng server-side (backend applications) bằng ngôn ngữ TypeScript hoặc JavaScript. Nó được xây dựng trên cơ sở của Node.js và sử dụng các khái niệm từ TypeScript để tạo ra một môi trường phát triển hiện đại và mạnh mẽ cho việc xây dựng các ứng dụng web và API.

Mục tiêu chính của NestJS là cung cấp một cấu trúc ứng dụng rõ ràng và dễ quản lý, giúp tăng tính bảo trì và sự tổ chức trong mã nguồn. Để đạt được điều này, NestJS triển khai mô hình kiến trúc lõi (core architecture) dựa trên các nguyên tắc của Angular, đặc biệt là sử dụng Dependency Injection (DI) và Modules (Các module).

Cấu trúc của NestJS được xây dựng dựa trên mô hình kiến trúc lõi (core architecture) giúp tạo ra một ứng dụng server-side (backend application) rõ ràng, dễ quản lý và dễ mở rộng. Cấu trúc NestJS thường được tổ chức thành các phần chính sau:

- Module (Các module): Module là một phần cơ bản trong cấu trúc NestJS. Mỗi ứng dụng NestJS bao gồm ít nhất một module gốc (root module) và có thể có nhiều module con. Module là nơi tổ chức các thành phần của ứng dụng như Controllers, Providers và các thành phần khác. Mỗi module đại diện cho một phần chức năng cụ thể của ứng dụng.

- Controller (Bộ điều khiển): Controllers là thành phần chịu trách nhiệm xử lý các yêu cầu HTTP từ phía client và trả về kết quả tương ứng. Controllers là nơi xử lý các request và trả về các response. Các phương thức của controller được chú thích (decorated) bằng các decorator như '@Get()', '@Post()', '@Put()', v.v., để chỉ định các route và phương thức HTTP tương ứng.

- Provider (Các nhà cung cấp): Providers là thành phần chịu trách nhiệm cung cấp các dịch vụ cho ứng dụng. Đây có thể là các service, repository, logger, v.v. Providers sử dụng dependency injection để chèn vào các thành phần khác và có thể được sử dụng bởi các controllers hoặc các providers khác.

- Middleware (Trung gian): Middleware là các hàm xử lý mà NestJS sử dụng để xử lý các yêu cầu HTTP trước khi chúng đến các route xử lý chính. Middleware có thể được sử dụng để thực hiện các thao tác chung như xác thực, ghi log, xử lý lỗi, v.v.

- Filter (Bộ lọc): Filters được sử dụng để xử lý các exception (ngoại lệ) xảy ra trong ứng dụng. Filters cho phép bạn xử lý và thay đổi response trước khi gửi về client khi có exception xảy ra.

- Guard (Bảo vệ): Guards được sử dụng để kiểm tra xem một yêu cầu có thể được xử lý hoặc không. Guards cho phép bạn thực hiện các kiểm tra xác thực hoặc kiểm tra quyền trước khi xử lý một yêu cầu.

- Interceptor (Bộ chặn): Interceptors là các hàm xử lý mà NestJS sử dụng để chặn và thay đổi response trước khi nó được gửi về client. Interceptors có thể được sử dụng để thực hiện các thao tác chung trên response trước khi nó đi ra ngoài.

- Exception (Ngoại lệ): Exception handling (xử lý ngoại lệ) là một phần quan trọng của cấu trúc NestJS. Exception handling cho phép bạn xử lý các exception xảy ra trong ứng dụng và trả về các thông báo lỗi thích hợp cho client.

Một trong những ưu điểm nổi bật của NestJS là việc hỗ trợ cả TypeScript và JavaScript. Lập trình viên có thể lựa chọn ngôn ngữ phù hợp với nhu cầu và kinh nghiệm của họ. TypeScript là một ngôn ngữ mở rộng của JavaScript, cung cấp kiểu dữ liệu tĩnh và các tính năng nâng cao, giúp mã nguồn dễ đọc và dễ bảo trì hơn.

NestJS triển khai mô hình kiến trúc lõi, trong đó mỗi ứng dụng bao gồm ít nhất một module gốc và có thể có nhiều module con. Mỗi module đại diện cho một phần chức năng cụ thể của ứng dụng, giúp mã nguồn trở nên rõ ràng, tổ chức tốt hơn và dễ quản lý.

5.3.3 RabbitMQ

Trước khi tìm hiểu RabbitMQ là gì, có 2 khái niệm cần làm rõ trước nhất là AMQP và Message Broker. AMQP hay Advanced Message Queuing Protocol là tên gọi dùng để

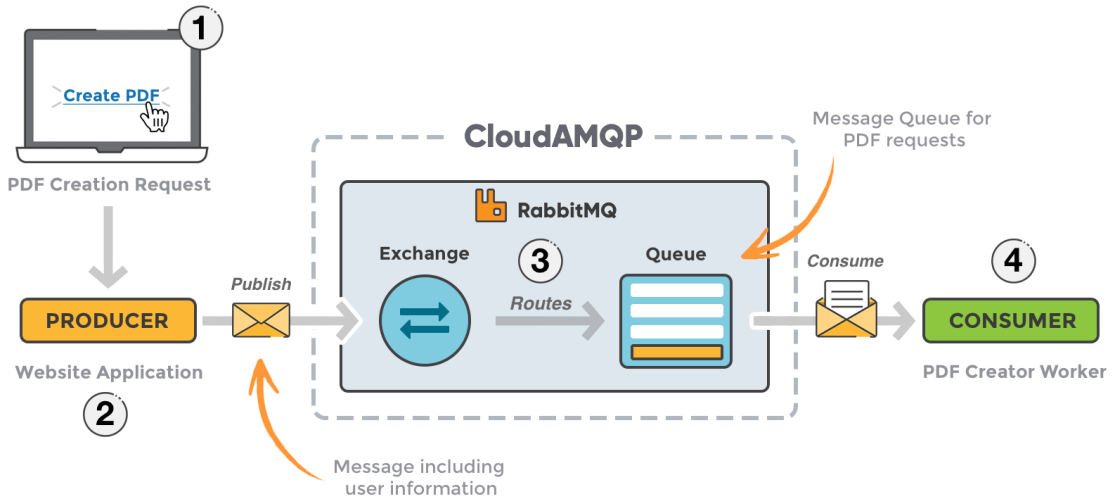
chỉ các giao thức xếp hàng tin nhắn nâng cao. Những giao thức này được ứng dụng cung cấp để để chuẩn mở cho các phần mềm trung gian hướng thông báo. Nhiệm vụ chính của AMQP là định hướng tin nhắn, xếp hàng, định tuyến, tăng độ tin cậy và bảo mật[22].

Message Broker là phần mềm dùng để giúp các ứng dụng khác nhau có thể giao tiếp với nhau một cách dễ dàng. Nó cũng được sử dụng như một ứng dụng trung chuyển tin nhắn và thông tin giữa các phần mềm khác nhau.

RabbitMQ là một Message Broker sử dụng giao thức AMQP để phục vụ cho hoạt động trao đổi tin nhắn giữa các ứng dụng với nhau. RabbitMQ được hiểu như một người vận chuyển các message và quản lý những tin nhắn trên hàng đợi.

RabbitMQ sử dụng ngôn ngữ lập trình Erlang. Mục đích của việc sử dụng ngôn ngữ này là giúp các lập trình viên trong nhiều lĩnh vực khác nhau có thể dễ dàng kết nối và giao tiếp với nhau. Erlang cũng giúp tạo lập nên một trạm trung gian để gửi và nhận các thông tin. Ngoài ra, nó cũng góp phần lưu trữ và bảo vệ an toàn cho các dữ liệu trong message. Nói chung, RabbitMQ được tạo ra với mục đích xử lý lượng lớn tin nhắn, thông điệp phức tạp giữa các ứng dụng với nhau. Nó giúp di chuyển, xử lý, biên dịch và lưu trữ các message.

RabbitMQ hoạt động như một bưu điện trung chuyển. Nhiệm vụ của nó là chờ người bán hàng mang hàng đến bưu cục và vận chuyển nó đến tay khách hàng. Khi một người có nhu cầu gửi thông tin, họ sẽ đẩy tin nhắn vào Message broker. Message broker sẽ tiếp nhận, lưu trữ bản sao và phiên dịch nếu cần thông tin đó. Rồi cuối cùng mới mang tin nhắn đến cho người dùng. Tại sao phải dùng RabbitMQ khi mà người gửi có thể trực tiếp gửi tin nhắn đến cho người nhận? Hệ thống chỉ có thể làm thủ công thế với những cuộc trao đổi 1:1. Khi một máy chủ cần phải gửi nhiều loại thông tin cho nhiều đối tượng khác nhau, RabbitMQ sẽ giúp tối ưu hóa quá trình này.



Hình 5.3.2: Cơ chế hoạt động của RabbitMQ

Một RabbitMQ sẽ bao gồm hai hoạt động chính đó là exchange và queue. Trong đó, exchange chịu trách nhiệm phân luồng thông tin thành các topic đã được cài trước khác nhau. Từ đó xác định đúng tin nhắn cho đúng đối tượng. Còn queue được hiểu như một danh sách chờ. Danh sách này bao gồm các tin nhắn được sắp xếp theo một thứ tự thời gian nhất định và lần lượt được gửi đi. Sau khi tin nhắn đã được gửi đi, nó tiếp tục phải chờ đợi nếu cho đến khi người nhận muốn lấy nó xuống. Tất nhiên là trong trường hợp người nhận cài đặt chế độ chờ thư như vậy trước.

RabbitMQ sở hữu rất nhiều tính năng ưu việt. Đó cũng là lý do khiến RabbitMQ lại được nhiều người sử dụng đến như vậy.

- Giao diện dễ sử dụng: Rabbit MQ sở hữu một bộ giao diện rất tối giản và cơ bản. Những mục chính và quan trọng được bố trí ở nơi dễ nhìn và dễ thao tác. Chính vì thế khi thực hiện các hoạt động trên RabbitMQ người dùng sẽ cảm thấy rất thoải mái và tiện lợi.

- Khả năng bảo mật tốt: Hệ thống lưu trữ của RabbitMQ có tính an toàn rất cao. Người dùng có thể yên tâm sử dụng nhiều tác vụ cùng lúc mà không cần lo đến việc bảo vệ dữ liệu.

- Tính linh hoạt cao: Các Message được thông qua router trước, sau đó mới đi đến queue. Nếu định tuyến sở hữu một mô hình phức tạp, người dùng có thể viết riêng các kiểu trao đổi như một plugin.

- Tạo sự liên kết chặt chẽ: RabbitMQ có khả năng tạo ra sự liên kết giữa các đối tượng với nhau. Nếu phải làm việc với các máy chủ không yêu cầu liên kết hoặc có hệ thống liên kết lỏng lẻo, RabbitMQ sẽ tiến hành gia cố lại các liên kết cho phù hợp với nhu cầu sử dụng của người dùng.

- Tối ưu hóa danh sách chờ: RabbitMQ là một công cụ gửi tin nhắn có sử dụng danh sách chờ. Bằng cách nhân bản nhiều queue ở những máy khác nhau trong quy trình truyền tin, người dùng có thể dễ dàng lấy lại dữ liệu khi máy chủ bị lỗi.

5.3.4 PostgreSQL

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ và đối tượng (object-relational database management system) miễn phí và mã nguồn mở tiên tiến nhất hiện nay. PostgreSQL đảm bảo khả năng mở rộng cao và tuân thủ các tiêu chuẩn kỹ thuật. Nó được thiết kế để xử lý một loạt các khối lượng công việc lớn, từ các máy tính cá nhân đến kho dữ liệu hoặc dịch vụ Web có nhiều người dùng đồng thời[23].

PostgreSQL tích hợp nhiều tính năng tuyệt vời giúp hỗ trợ nhà phát triển xây dựng ứng dụng đáp ứng các chức năng phức tạp, truy vấn nhanh chóng và bảo mật duy trì

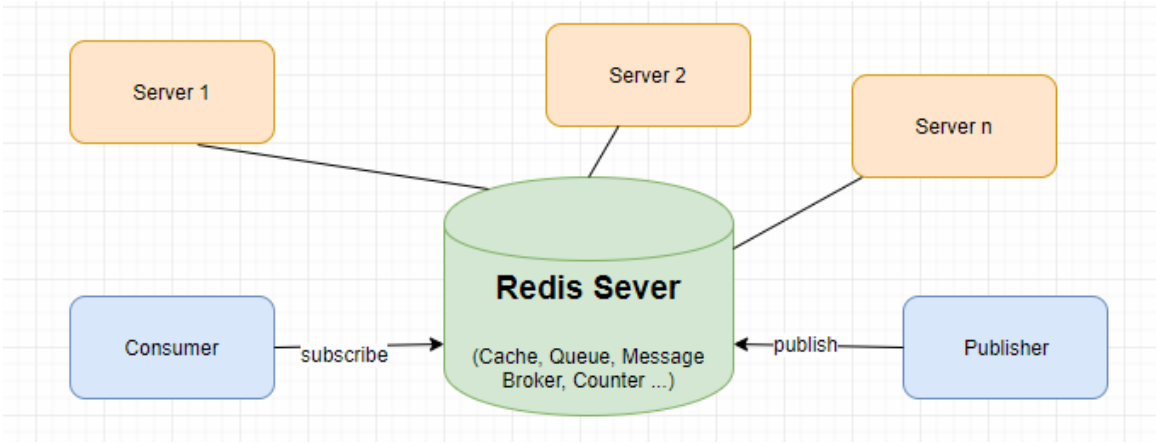
tính toàn vẹn và độ tin cậy. Để đáng tin cậy hơn, PostgreSQL cung cấp các tùy chọn bảo mật, xác thực và khôi phục thảm họa khác nhau. PostgreSQL được chứng minh là có khả năng mở rộng cao cả về số lượng dữ liệu và số lượng người dùng có thể thao tác cùng lúc.

Dưới đây mà một số tính năng nổi bật của PostgreSQL:

- Cung cấp nhiều kiểu dữ liệu: PostgreSQL cung cấp đa dạng kiểu dữ liệu như nguyên hàm (các nguyên số, boolean, số, chuỗi), cấu trúc (UUID, phạm vi,...), hình học, document,...
- Bảo đảm toàn vẹn dữ liệu: Dữ liệu trong PostgreSQL đảm bảo tính toàn vẹn bằng cách ràng buộc loại từ, Primary Keys, Foreign Keys, khóa khuyến nghị, khóa hàm số,...
- Tính năng thiết lập linh hoạt: Người dùng được thiết lập danh mục từ đơn giản đến phức tạp, tối ưu hóa tốc độ truy cập, hỗ trợ thống kê trên nhiều cột,...
- Chức năng bảo mật: PostgreSQL hỗ trợ xây dựng hàng rào bảo mật, xác thực mạnh (SCRAM-SHA-256, SSPI, LDAP, GSSAPI, Certificate,...), hệ thống kiểm soát truy cập kĩ càng, bảo mật cấp độ cột – hàng.
- Khả năng mở rộng: Người dùng thực hiện mở rộng hệ thống qua các phương pháp lưu trữ, ngôn ngữ thủ tục (PL / PGSQL, Python, Perl, và nhiều ngôn ngữ khác), PostGIS, kết nối cơ sở dữ liệu hoặc luồng khác với giao diện SQL chuẩn.
- Chức năng tìm kiếm văn bản: PostgreSQL cung cấp tính năng tìm kiếm văn bản đầy đủ, hệ thống hóa ký tự theo cách khoa học (thông qua ICU collations).

5.3.5 Redis

Redis (REmote DIctionary Server) là một mã nguồn mở được dùng để lưu trữ dữ liệu có cấu trúc, có thể sử dụng như một database, bộ nhớ cache hay một message broker. Redis ngoài tính năng lưu trữ KEY-VALUE trên RAM thì Redis còn hỗ trợ tính năng xắp xếp, query, backup dữ liệu trên đĩa cứng cho phép bạn có thể phục hồi dữ liệu khi hệ thống gặp sự cố... và có thể nhân bản (chạy nhiều Server Redis cùng lúc)[24].



Hình 5.3.3: Các ứng dụng của Redis

- Caching: Sử dụng làm bộ nhớ đệm. Chính tốc độ đọc ghi nhanh mà Redis có thể làm bộ nhớ đệm, nơi chia sẻ dữ liệu giữa các ứng dụng hoặc làm database tạm thời. Ngoài ra Redis có thể sử dụng để làm Full Page Cache cho website. Cũng vì tính nhất quán của Redis, cho dù restart Redis thì người dùng cũng không có cảm nhận chậm khi tải trang.

- Counter: Sử dụng làm bộ đếm. Với thuộc tính tăng giảm thông số rất nhanh trong khi dữ liệu được lưu trên RAM, sets và sorted sets được sử dụng thực hiện đếm lượt view của một website, các bảng xếp hạng trong game chẳng hạng. Redis hỗ trợ thread safe do đó nó có thể đồng bộ dữ liệu giữa các request.

- Publish/Suscribe (Pub/Sub): Tạo kênh chia sẻ dữ liệu. Redis hỗ trợ tạo các channel để trao đổi dữ liệu giữa publisher và subscriber giống như channel trong Socket Cluster hay topic trong Apache Kafka. Ví dụ: Pub/Sub được sử dụng theo dõi các kết nối trong mạng xã hội hoặc các hệ thống chat.

- Queues: Tạo hàng đợi để xử lý lần lượt các request. Redis cho phép lưu trữ theo list và cung cấp rất nhiều thao tác với các phần tử trong list, vì vậy nó còn được sử dụng như một message queue.

6 Tổng kết

6.1 Tính cấp thiết của đề tài

Công nghệ đang ngày càng phát triển, nên những phương pháp truyền thống thủ công dần được loại bỏ, trong đó có vấn đề lưu trữ và quản lý tài liệu văn bản. Các phương pháp truyền thống đã trở nên lỗi thời và tồn động rất nhiều điểm yếu. Do đó, nhu cầu về hệ thống số hóa và quản lý tài liệu ngày càng tăng cao. Trên thị trường hiện nay cũng đã xuất hiện rất nhiều hệ thống lưu trữ và quản lý tài liệu, tuy nhiên, trong quá trình nghiên cứu, chúng tôi càng thấy rõ thêm những hạn chế của các hệ thống lưu trữ và quản lý văn bản hiện nay. Qua đó càng cho thấy rõ tính cấp thiết của đề tài "Xây dựng hệ thống số hóa và quản lý văn bản".

6.2 Đánh giá kết quả

Trong giới hạn đồ án này, chúng tối đã hoàn thành được các mục sau:

- **Nghiên cứu và rút ra được giải pháp xây dựng cấu trúc dữ liệu:** Sử dụng cấu trúc dữ liệu kết hợp giữa xây dựng 2 miền hỗ trợ và cho người dùng tự xây dựng cấu trúc dữ liệu.
- **Xây dựng được tiêu chí cho 2 miền đã chọn**
- **Xây dựng được metadata cho tập tin và thư mục**
- **Nghiên cứu và rút ra được phương thức để có thể tương tác với văn bản:** Sử dụng Langchain và GPT để có thể liên kết được với bảng tiêu chí và rút ra được những từ khóa để lưu vào metadata hỗ trợ tìm kiếm.
- **Nghiên cứu cách sử dụng Elasticsearch:** Elasticsearch là một engine tìm kiếm mạnh mẽ, do đó, sẽ áp dụng nó để hỗ trợ việc lưu trữ và tìm kiếm.
- **Nghiên cứu và rút ra được giải pháp lưu trữ tập tin hợp lý:** Sử dụng AWS S3 để lưu trữ tập tin một cách linh động và hiệu quả.
- **Nghiên cứu và tìm kiếm được giải pháp sử dụng OCR:** Sử dụng phần mềm mã nguồn mở Tesseract làm công cụ OCR cho hệ thống.
- **Đưa ra được thiết kế hệ thống, mô tả sự liên kết các phần mềm và ứng dụng bên thứ ba với hệ thống:** Đưa ra được các công nghệ sẽ sử dụng để phát triển hệ thống

6.3 Hướng phát triển

Thông qua đồ án này, chúng tôi đã đề xuất được các phương pháp, công nghệ và công cụ sử dụng để xây dựng "Hệ thống số hóa và quản lý văn bản". Ở giai đoạn tiếp theo chúng tôi sẽ tập trung phát triển hệ thống. Cụ thể:

- Tiếp tục xây dựng khung tiêu chí để cho đầy đủ và tổng quát hơn.
- Xây dựng và hoàn thiện biểu thức chính quy cho hai miền đã xây dựng.
- Huấn luyện cho Tesseract OCR và Langchain GPT để có thể cho ra kết quả chính xác.
- Xây dựng hệ thống theo các phương pháp và công nghệ đã nghiên cứu.
- Tiếp tục nghiên cứu phương pháp tìm từ đồng nghĩa, đề xuất từ khi tìm kiếm.
- Tìm hiểu các vấn đề có thể gặp phải khi xử lý văn bản lớn giữa nhiều dịch vụ.

Tài liệu

[1] Ricoh.(2023). *"Nhược Điểm Quản Lý Hồ Sơ Vật Lý Và Tại Sao Chuyển Sang Hệ Thống Quản Lý Và Lưu Trữ Tài Liệu?"*. https://www.ricoh.com.vn/blogs-and-insights/he-thong-luu-tru-tai-lieu. Đã truy cập: 12-16-2023.

[2] Nguyễn Bùi Nhật Mỹ. *"Google drive là gì? Cách dùng các tính năng miễn phí tiện lợi của Google drive mà bạn chưa biết"*. https://www.dienmayxanh.com/kinh-nghiem-hay/google-drive-la-gi-cach-dung-cac-tinh-nang-mien-ph-1133563. Đã truy cập: 07-12-2023.

[3] Tuan Nguyen. *"SharePoint là gì? Tại sao nên sử dụng SharePoint trong doanh nghiệp?"*. https://fptshop.com.vn/tin-tuc/danh-gia/sharepoint-la-gi-tai-sao-nen-su-dung-sharepoint-trong-doanh-nghiep-147269. Đã truy cập: 07-12-2023.

[4] DocuWare. *"About DocuWare"*. https://start.docuware.com/about. Đã truy cập: 16-12-2023.

[5] DocuWare. *"DocuWare Cloud"*. https://start.docuware.com/docuware-cloud. Đã truy cập: 16-12-2023.

[6] TrustRadius. *"DocuWare Pricing Overview"*. https://www.trustradius.com/products/docuware/pricing. Đã truy cập: 16-12-2023.

[7] Amazon Web Services. *"OCR (Nhận dạng ký tự quang học) là gì?"*. https://aws.amazon.com/vi/what-is/ocr/. Đã truy cập: 16-12-2023.

[8] Mori, S., Suen, C. Y., & Yamamoto, K. (1992). *"Historical review of OCR research and development"*. IEEE, 80(7), pp. 1029 – 1058.

[9] Wikipedia. *"Xử lý ngôn ngữ tự nhiên"*. https://vi.wikipedia.org/wiki/X%E1%BB%AD_l%C3%BD_ng%C3%B4n_ng%E1%BB%AF_t%E1%BB%B1_nhi%C3%AAn. Đã truy cập: 08-12-2023.

[10] Bui Quang Manh. *"Word Embedding - Tìm hiểu khái niệm cơ bản trong NLP"*. https://viblo.asia/p/word-embedding-tim-hieu-khai-niem-co-ban-trong-nlp-1Je5E93G5nL. Đã truy cập: 08-12-2023.

[11] Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman. *"Mining of Massive Datasets"*. Cambridge University Press, 2014, pp. 1 – 27.

[12] Wikipedia. *"Microservices"*. https://vi.wikipedia.org/wiki/Microservices. Đã truy cập: 09-12-2023.

[13] Chellammal Surianarayanan, Gopinath Ganapathy, Pethuru Raj. *"Essentials of Microservices Architecture: Paradigms, Applications, and Techniques"*. CRC Press Taylor & Francis Group, 2020, pp. 1 – 108.

[14] Cục Xuất bản, in và phát hành. (25-06-2019). *"Danh sách nhà xuất bản"*. https://mic.gov.vn/solieubaocao/Pages/TinTuc/139248/Danh-sach-nha-xuat-ban.html. Đã truy cập: 19-11-2023.

[15] Reedysblog. (29-12-2020). *"The Ultimate List of Book Genres: 35 Popular Genres, Explained"*. https://blog.reedsy.com/book-genres/. Đã truy cập: 19-11-2023.

[16] Phạm Văn Toàn. *"Langchain #1 - Điểm qua các chức năng sừng sỏ nhất của Langchain"*. https://viblo.asia/p/langchain-1-diem-qua-cac-chuc-nang-sung-so-nhat-cua-langchain-mot-framework-cuc-ba-dao-khi-lam-viec-voi-llm-BQyJKmrqVMe. Đã truy cập: 09-12-2023.

[17] Viet Anh. *"Elasticsearch là gì?"*. https://viblo.asia/p/elasticsearch-la-gi-1Je5E8RmlnL. Đã truy cập: 09-12-2023.

[18] AWS. *"Amazon S3 pricing"*. https://aws.amazon.com/s3/pricing/. Đã truy cập: 09-12-2023.

[19] Swetha Kumaraswamy. (2023). *"8 Best OCR Software and Tools (Free + Paid) in 2023"*. https://happay.com/blog/best-ocr-software/. Đã truy cập: 16-12-2023.

[20] Bill Karwin. *"Models for Hierarchical Data with SQL and PHP"*. Percona Inc, pp. 6 – 69.

[21] Lê Hoàng. *"ReactJS là gì? Tất tần tật những điều căn bản về ReactJS"*. https://stringee.com/vi/blog/post/reactJS-la-gi. Đã truy cập: 10-12-2023.

[22] Khánh Kim. *"RabbitMQ là gì? Những thông tin cơ bản nhất cho người mới tìm hiểu"*. https://teky.edu.vn/blog/rabbitmq-la-gi/. Đã truy cập: 10-12-2023.

[23] Nguyễn Văn Thịnh. *"Tìm hiểu hệ quản trị cơ sở dữ liệu PostgreSQL"*. https://viblo.asia/p/tim-hieu-he-quan-tri-co-so-du-lieu-postgresql-m68Z0eLdlkG. Đã truy cập: 10-12-2023.

[24] Topdev. *"Redis là gì? Ưu điểm của nó và ứng dụng"*. https://topdev.vn/blog/redis-la-gi/. Đã truy cập: 10-12-2023.