LỜI CAM ĐOAN

Tôi xin cam đoan:

1. Những nội dung trong luận văn này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của Thầy **PGS. TS Đỗ Văn Nhơn**.
2. Mọi tham khảo trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian công bố.

Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo tôi xin chịu hoàn toàn trách nhiệm.

Tp. Hồ Chí Minh, ngày 18 tháng 07 năm 2022

Học viên thực hiện luận văn

Hà Hoài Nam

LỜI CẢM ƠN

Xin cho tôi được gửi lòng biết ơn đến Thầy **PGS.TS. Đỗ Văn Nhơn** – người đã hướng dẫn luận văn cho tôi. Trong suốt thời gian thực hiện luận văn, Thầy đã tận tình hướng dẫn và có những lời khuyên, những đóng góp rất quý báu, giúp cho tôi định hướng và hoàn thành các mục tiêu đề ra.

Tôi xin chân thành tỏ lòng biết ơn đến quý Thầy, Cô đã tận tình giảng dạy cho tôi trong suốt các năm học qua trong chương trình đào tạo Thạc sĩ Hệ thống thông tin, Khoa Sau Đại Học, Học Viện Bưu Chính Viễn Thông Thành Phố Hồ Chí Minh.

Cho tôi được gửi lòng biết ơn trân trọng đến những người lãnh đạo cơ quan, đã tạo điều kiện thuận lợi để tôi công tác và học tập.

Xin cảm ơn tất cả bạn bè đã động viên, giúp đỡ và đóng góp cho tôi nhiều ý kiến quý báu, qua đó giúp chúng tôi hoàn thiện hơn cho đề tài này.

Và cuối cùng, tôi cũng không quên gửi lời cảm ơn đến tác giả của các báo cáo nghiên cứu khoa học mà tôi đã tham khảo và tìm hiểu cho đề tài.

Luận văn đã hoàn thành với một số kết quả nhất định tuy nhiên vẫn không tránh khỏi thiếu sót. Kính mong sự cảm thông và đóng góp ý kiến từ quý thầy cô và các bạn.

Một lần nữa tôi xin chân thành cảm ơn!

Tp. Hồ Chí Minh, ngày 18 tháng 07 năm 2022

Học viên thực hiện luận văn

Hà Hoài Nam

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Viết tắt	**Diễn giải**
CRFs	Conditional Random Fields
NE	Named Entity
NER	Named Entity Recognition
VSM	Vector Space Model
OCR	Optical Character Recognition
IR	Information Retrieval
CSDL	Cơ sở dữ liệu
HTML	HyperText Markup Language
XML	Extensible Markup Language
CK_ONTO	Classed Keyphrase based Ontology
MEMM	Mô hình cực đại hóa Entropy
HMM	Mô hình Markov ẩn

DANH SÁCH CÁC BẢNG

DANH SÁCH CÁC HÌNH VẼ

MỞ ĐẦU

Ngày nay cùng với sự phát triển của internet thì dữ liệu của ngành công nghệ thông tin ngày càng gia tăng. Nhu cầu quản lý, chia sẻ, tìm kiếm thông tin trong ngành này cũng được đặt ra và đáp ứng một phần nhờ các công cụ tìm kiếm. Một số công cụ tìm kiếm nổi tiếng hiện nay như Google hay Yahoo đều có thể cho phép người dùng tìm kiếm dữ liệu có liên quan bằng cách nhập từ khóa và tìm những tài liệu có chứa từ khóa đó. Với phương pháp tìm như vậy thì kết quả tìm kiếm đôi khi chẳng liên quan gì đến cái mà người dùng muốn tìm, vì các công cụ tìm kiếm này không hiểu được ý nghĩa mà người dùng cần tìm. Như vậy các công cụ tìm kiếm thông tin về từ khóa sẽ không trả lời các câu hỏi tìm ẩn mà người dùng muốn tìm kiếm trên hệ thống.

Các hệ thống tìm kiếm này phần lớn vẫn dựa trên từ khóa và mức độ phổ biến của tài liệu. Một danh sách các từ khóa là dạng biểu diễn sơ lược nhất của nội dung, nghĩa là mỗi tài liệu được biểu diễn bởi một tập từ hay cụm từ được rút trích từ chính nội dung của tài liệu và do đó, cách biểu diễn này mang mức độ thông tin còn thấp. Do đó hệ thống tìm kiếm này có kết quả trả về không phải lúc nào cũng thỏa mãn yêu cầu tìm kiếm của người sử dụng, như là độ chính xác không cao khi kết quả trả về quá nhiều mà tỷ lệ số tài liệu hữu ích trên tổng số tài liệu trả về thấp, hoặc có thể không tìm thấy được những tài liệu liên quan khi chúng được mô tả với những từ khóa khác đồng nghĩa hoặc gần nghĩa với từ khóa mà người dùng tìm kiếm (độ bao phủ không cao) gây ra không ít khó khăn cho người sử dụng trong việc tìm kiếm chính xác thông tin mình cần

Như vậy làm thế nào để việc tìm kiếm của người sử dụng có hiệu quả hơn. Để giải quyết các vấn đề trên cần phải xây dựng một hệ thống cho phép tra cứu, tìm kiếm tài liệu theo đa dạng hơn không chỉ hỗ trợ tìm kiếm dựa trên từ khóa mà còn hỗ trợ tìm kiếm dựa trên tri thức của lĩnh vực hay theo ngữ nghĩa, trả về tập tài liệu kết quả đúng nhất với ý định của người dùng.

Ứng dụng đã được cài đặt, thử nghiệm tại Hệ thống quản lý văn bản Tic-Office của Hội nông dân tỉnh Tây ninh. Kết quả thực nghiệm bước đầu cho thấy giải pháp đã đề xuất là khả quan và có khả năng ứng dụng tốt.

Nội dung của luận văn được trình bày trong 5 chương, bao gồm:

Chương 1: Giới thiệu và khảo sát các hệ thống tìm kiếm thông tin, phân tích đánh giá thực trạng, trình bày mục tiêu, giới hạn của đề tài, ý nghĩa lý luận và thực tiễn, phương pháp nghiên cứu, hướng tiếp cận giải quyết vấn đề và nội dung thực hiện của đề tài.

Chương 2: Trình bày cơ sở lý thuyết của đề tài liên quan đến vấn đề truy hồi thông tin bao mô tả cấu trúc, các phương pháp truy hồi thông tin và đánh giá hệ thống truy hồi thông tin. Các lý thuyết nền tảng về mô hình không gian vector Ontology cùng với các phương pháp xây dựng mô hình dữ liệu.

Chương 3: Mô hình và giải pháp: Chương này đề xuất các mô hình gồm một mô hình ontology mô tả tri thức về một lĩnh vực đặc biệt trong đó sử dụng keyphrase là thành phần chính để hình thành các khái niệm của ontology; Các kỹ thuật xử lý tài liệu văn bản; Xây dựng mô hình VSM trong tra cứu tài liệu có sử dụng ngữ nghĩa cho câu truy vấn.

Chương 4: Cài đặt thử nghiệm và đánh giá: Thiết kế mô hình dữ liệu ontology hỗ trợ xử lý câu truy vấn; Xây dựng chức năng tra cứu nâng cao cho hệ thống quản lý văn bản Tic-Office. Tiến hành thực nghiệm, so sánh và đánh giá kết quả

Chương 5: Kết luận và hướng phát triển: Tổng kết những kết quả đạt được của luận văn, những hạn chế và hướng phát triển của đề tài trong tương lai.

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 Giới thiệu tổng quan về vấn đề nghiên cứu

1.1.1 Nhu cầu và thực trạng tìm kiếm hiện nay

Cách đây hơn 20 năm, vào năm 1997 lần đầu tiên Việt Nam kết nối Internet. Tuy là nước cho phép mở Internet chậm hơn so với thế giới, nhưng đến nay, Việt Nam lại đang là quốc gia có sự phát triển Internet mạnh mẽ nhất và đây được xem là động lực cho phát triển kinh tế văn hóa xã hội và hội nhập quốc tế, hiện nay Việt Nam đang thực hiện chuyển đổi số toàn diện trong tất cả các lĩnh vực đời sống, kinh tế, chính trị và xã hội. Một trong nhưng điều làm nên sự thành công của việc chuyển đổi số là hạ tầng mạng viễn thông phủ rộng khắp, từ thành thị đến nông thôn được phủ sóng 3G/4G với tốc độ cao. Ngoài ra một yếu tố không thể thiếu cho sự thành công đó là số lượng người dân có sử dụng Internet với số lượng khá lớn so với tổng dân số toàn quốc cụ thể như:

Dân số Việt Nam có 68.72 triệu người có sử dụng Internet trên tổng số 97.8 triệu dân đạt 70.3 % tỷ lệ người dân sử dụng Internet, cao hơn mức trung bình của thế giới là 59.2%, tỷ lệ người dùng truy cập trên thiết bị di động đạt 96.9% và có thời gian sử dụng trung bình trong ngày đạt 6 giờ 47 phút [17].

Internet đã góp phần làm thay đổi cuộc sống có chất lượng tốt hơn, đã trở thành phương tiện giúp việc truyền đạt, trao đổi thông tin, hợp tác, giao lưu… giữa mọi cá nhân, tổ chức trên khắp thế giới, là nơi chia sẻ thông tin, hình ảnh (cá nhân, tập thể hoặc của một nhóm người nào đó) thông qua các mạng xã hội. Ngoài ra, mọi người truy cập Internet để livestream, xem bộ phim hay, nghe nhạc,mua hàng online... và đặc biệt hiện nay nhu cầu học trực tuyến đang được rất nhiều người quan tâm trong thời gian diễn dịch bệnh. vì vậy, Internet đã làm thay đổi tư duy, suy nghĩ của đa số người dân từ việc mạnh dạn mua hàng online, chủ động được trong việc học tập, giải trí vô hình chung góp phần làm cho Internet ngày càng phát triển.

Bên cạnh nhu cầu về học tập, giải trí thì nhu cầu tìm kiếm thông tin là một nhu cầu không thể thiếu khi sử dụng Internet, theo [17] thống kê những trang web được

có lượng truy cập nhiều nhất tại Việt Nam tháng 12 năm 2020, trong đó trang tìm kiếm Google với hơn 1 tỷ lượt truy cập trong tháng, như vậy cho thấy nhu cầu tìm kiếm của người dùng khi có sử dụng Internet là rất lớn, vì vậy nhu cầu tìm kiếm thông tin được xem quan trọng nhất trong nhu cầu sử dụng internet của người dùng.

#	WEBSITE	TOTAL VISITS	UNIQUE VISITS	TIME PER VISIT	PAGES PER VISIT
01	GOOGLE.COM	1.08B	54.2M	19M 53S	6.84
02	VNEXPRESS.NET	475M	32.3M	16M 46S	4.41
03	24H.COM.VN	313M	31.3M	15M 47S	4.82
04	YOUTUBE.COM	266M	37.4M	32M 02S	4.63
05	KENH14.VN	263M	32.8M	11M 09S	4.27
06	FACEBOOK.COM	254M	32.6M	25M 58S	5.50
07		220M	29.3M	18M 03S	2.23
08	GOOGLE.COM.VN	177M	19.9M	14M 25S	7.53
09	DANTRI.COM.VN	164M	18.1M	14M 08S	4.27
10	VIETNAMNET.VN	142M	28.0M	11M 21S	3.38

Hình 1.1: Website có lượng truy cập nhiều nhất trong tháng 12/2020 [17]

1.1.2 Khảo sát hệ thống tìm kiếm văn bản

Hầu hết đối với các hệ thống quản lý dữ liệu hiện nay thì các yêu cầu về quản lý, chia sẻ và tìm kiếm thông tin là chức năng cơ bản cần phải có trong hệ thống quản lý. Trong đó chức năng tra cứu thông tin quản lý thì chỉ dừng ở mức độ tìm kiếm cơ bản theo từ khóa được lưu trữ trong dữ liệu. Với phương pháp tìm kiếm theo từ khóa thì kết quả chỉ tìm được nội dung liên quan tới từ khóa chứ không tìm được các nội dung liên quan tìm ẩn trong nội dung tìm kiếm.

Để hệ thống có thể giao tiếp được với người dùng thì cần phải có công cụ hỗ trợ xây dựng các quy tắc, các đối tượng cụ thể cho từng nhóm tri thức. Một trong các mô hình dữ liệu được sử dụng trong thời gian gần đây là mô hình Ontology.

Ontology biểu diễn các nội dung liên quan về một lĩnh vực cụ thể, mô tả mọi thứ trong lĩnh vực đó như là các thuật ngữ, từ viết tắt, từ đồng nghĩa, các thuộc tính và mối quan hệ giữa chúng. Từ đó mới có thể giúp hệ thống có thể hiểu được các ngữ nghĩa tìm ẩn trong nội dung tìm kiếm.

Ontology là một hướng tiếp cận mới trong việc nghiên cứu và phát triển trong lĩnh vực tìm kiếm thông tin theo ngữ nghĩa. Nghiên cứu ứng dụng của Ontology có thể áp dụng trong nhiều lĩnh vực, một trong lĩnh vực dùng trong việc xây dựng các hệ thống truy xuất nội dung từ các tài liệu liên quan, để đáp ứng nhu cầu tìm kiếm được đầy đủ và chính xác, hỗ trợ rất nhiều cho công tác quản trị hệ thống quản lý tài liệu, đồng thời góp phần tiết kiệm nhiều thời gian trong việc tìm kiếm văn bản.

Từ nhu cầu thực tế của hệ thống quản lý văn bản của Hội nông dân tỉnh Tây Ninh cùng với sự hướng dẫn tận tình của Thầy PGS.TS Đỗ Văn Nhơn, tôi quyết định chọn đề tài: "**Xây Dựng Chức Năng Tra Cứu Thông Tin Văn Bản Dựa Trên Web Ngữ Nghĩa Của Hệ Thống Tic-Office**" làm luận văn tốt nghiệp.

1.2 Mục tiêu đề tài

Qua quá trình khai thác sử dụng hệ thống Tic-Office, hệ thống cũng đã mang lại một số lợi ích cho công tác quản lý điều hành văn bản một cách thuận tiện, nhanh chóng. Văn bản được quản lý trên hệ thống trực tuyến có thể xử lý, điều hành từ xa, từ đó góp phần nâng cao công tác điều hành của nhà quản lý. Bên cạnh đó ngoài các chức năng về quản lý văn bản đến và đi thì hệ thống còn có chức năng tìm kiếm lại văn bản đã được lưu trữ. Hiện tại chức năng của hệ thống chỉ mới đáp ứng nhu cầu tìm kiếm thông tin cơ bản cho người dùng.

Hệ thống Tic-Office cung cấp chức năng tra cứu theo từ khóa dựa trên nội dung trích yếu mà hệ thống lưu trữ trong dữ liệu, hoặc tìm theo số ký hiệu của văn bản được quản lý.

Ví dụ: Người dùng muốn tìm kiếm từ khóa "HĐND" thì hệ thống sẽ trả về trích yếu có bao gồm từ khóa "HĐND" mà không tìm thấy các nội dung có liên quan trong nội dung của tài liệu văn bản được nêu trong Hình 1.2

Hình 1.2: Kết quả chức năng tìm kiếm theo từ khóa

Một hệ thống tra cứu thông tin văn bản là một hệ thống sẽ tìm tất cả nội dung của trích yếu và nội dung của tài liệu mà có liên quan đến thông tin mà người sử dụng hệ thống cần tra cứu. Những thông tin được người dùng đưa vào hệ thống bởi các câu truy vấn (query). Những tài liệu - văn bản "liên quan" (relevant) với câu truy vấn sẽ được hệ thống trả về.

Để đáp ứng yêu cầu tra cứu có thể tìm kiếm đầy đủ thông tin trong tài liệu trong hệ thống thì đề tài cần thực hiện các nội dung như sau:

- Tìm hiểu về web ngữ nghĩa, xây dựng mô hình dữ liệu hỗ trợ biểu diễn câu truy vấn
- Tìm hiểu về kỹ thuật xử lý ngôn ngữ tự nhiên, kỹ thuật rút trích dữ liệu từ hình anh scan của tài liệu.
- Kỹ thuật so khớp giữa tài liệu và câu truy vấn sử dụng mô hình VSM.
- Xây dựng chức năng tra cứu nâng cao cho hệ thống Tic-Office để hỗ trợ người dùng trong tìm kiếm văn bản được đầy đủ.

1.3 Đối tượng và phạm vi nghiên cứu

1.3.1 Đối tượng nghiên cứu

Hệ thống quản lý văn bản của Hội Nông Dân, nhu cầu và hiện trạng tra cứu

Phương pháp tổng hợp nội dung bằng cách rút trích đặc trưng văn bản sử dụng Named Entity Recognition (NER) để rút trích thực thể có nghĩa từ văn bản.

Quy trình thiết kế một ontology với thực thể là một từ hoặc cụm từ đồng nghĩa hoặc tên viết tắt. Một thực thể có thể bao gồm lớp, tên viết tắt, từ đồng nghĩa, định danh, thuộc tính…để biểu diễn thông tin tìm ẩn của các từ để hỏi trong câu truy vấn

Công cụ Protégé mô tả thiết kế mô hình ontology

Các thư viện hỗ trợ phát triển ứng dụng Web ngữ nghĩa (owlready2, owlDotNetApi, SemWeb).

Các mô hình tìm kiếm văn bản và so sánh độ tương đồng của các văn bản.

Phân tích thiết kế và xây dựng chức năng tìm kiếm văn bản theo ngữ nghĩa cho hệ thống Tic-Office

1.3.2 Phạm vi nghiên cứu

Hằng năm, Hệ thống Tic-Office quản lý văn bản đến, văn bản đi với số lượng hơn 2000 văn bản trên hệ thống trong một năm. Với số lượng văn bản lớn thì nội dung lưu trữ cũng rất nhiều thông tin, vì vậy trong đề tài này tôi chỉ giới hạn phạm vi thực hiện tìm kiếm theo: nội dung tìm ẩn có mối liên hệ với nhau thông qua từ đồng nghĩa, từ viết tắt và nội dung của tài liệu giới hạn trong năm 2021 của hệ thống.

1.4 Phương pháp nghiên cứu

1.4.1 Giả thuyết nghiên cứu

Khi hệ thống Tic-Office kết hợp sử dụng mô hình dữ liệu để biểu diễn nội dung câu truy vấn vào chức năng tra cứu thông tin thì kết quả nội dung tìm kiếm sẽ bao gồm những nội dung tìm ẩn trong văn bản mà khi sử dụng chức năng tra cứu theo từ khóa không thực hiện được.

Hệ thống tra cứu theo ngữ nghĩa sẽ góp phần vào công tác tìm kiếm, khai thác và sử dụng tài liệu một cách hiệu quả, tốt hơn so với chức năng tra cứu hiện tại đang sử dụng.

1.4.2 Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết: Tìm hiểu kỹ thuật OCR, kỹ thuật rút trích NE, nghiên cứu các lý thuyết liên quan đến xây dựng hệ thống Web ngữ nghĩa. Thu thập, tổng hợp thông tin về văn bản của hệ thống Tic-Office.

Phương pháp khảo sát: Tìm hiểu quy trình lưu trữ, cấu trúc dữ liệu và công tác quản lý các văn bản của hệ thống Tic-Office. Tìm hiểu hệ thống tra cứu tại của hệ thống Tic-Office hiện có.

Phương pháp thực nghiệm: Xây dựng chức năng tra cứu nâng cao, so sánh với chức năng tra cứu hiện tại, đánh giá kết quả đạt được của hai chức năng tra cứu

1.5 Ý nghĩa khoa học và thực tiễn của đề tài

1.5.1 Ý nghĩa khoa học

Áp dụng công nghệ mới trong tìm kiếm thông tin của tài liệu của Web ngữ nghĩa. Phát triển các ứng dụng để góp phần từng bước phổ biến và làm phát triển công nghệ này.

1.5.2 Ý nghĩa thực tiễn

Ứng dụng các kỹ thuật xử lý ngôn ngữ tự nhiên, kỹ thuật rút trích NE, sử dụng mô hình VSM có sử dụng thực thể vào lĩnh vực tìm kiếm nội dung văn bản theo ngữ nghĩa, góp phần phục vụ tốt công tác nghiên cứu, tìm hiểu, sử dụng và khai thác tài liệu của hệ thống Tic-Office. Hỗ trợ công tác văn thư, lưu trữ và tra cứu tài liệu một cách nhanh chóng.

1.6 Nội dung thực hiện

Nghiên cứu khảo sát hiện trạng của hệ thống quản lý văn bản Tic-Office trong việc quản lý văn bản đến, văn bản đi và tra cứu văn bản. Phân tích hiện trạng nhu cầu tìm kiếm và khả năng mở rộng nhu cầu tìm kiếm của ứng dụng.

Chuẩn bị thu thập tổng hợp nội dung dữ liệu từ các tập tin văn bản của hệ thống

Nghiên cứu phương pháp xây dựng mô hình dữ liệu ontology biểu diễn nội dung của câu truy vấn

Nghiên cứu các phương pháp xử lý bao gồm:

- Tìm hiểu kỹ thuật rút trích nội dung văn bản từ hình ảnh
- Tìm hiểu kỹ thuật rút trích thực thể từ nội dung văn bản
- Nghiên cứu giải pháp so khớp nội dung văn bản sử dụng mô hình không gian vector kết hợp thực thể biểu diễn nội dung của câu truy vấn

Nghiên cứu các thư viện, công cụ hỗ trợ xây dựng chức năng tra cứu nâng cao cho hệ thống

Bổ sung chức năng tra cứu theo ngữ nghĩa cho hệ thống Tic-Office.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Vấn đề truy tìm thông tin

2.1.1 Cấu trúc của một hệ thống truy tìm thông tin

Hệ thống truy tìm thông tin (Information Retrieval, viết tắt IR) là một hệ thống tìm kiếm thông tin các yêu cầu của người dùng đặt ra và thực hiện tìm kiếm trong tất cả nguồn dữ liệu mà hệ thống đang lưu trữ, quản lý để trả về cho người dùng danh sách thông tin đúng với yêu cầu đưa ra.

Hệ thống IR tập trung chủ yếu vào văn bản (document) được quản lý, lưu trữ, truy xuất bằng cách nào để dễ dàng có thể truy vấn (query) nhanh chóng, kịp thời.

Tài liệu là các đối tượng chứa các thông tin bao gồm các đối tượng như các tài liệu văn bản, hình ảnh, âm thanh, video… Tuy nhiên các hệ thống truy hồi thông tin đa phần chỉ đề cập đến loại đối tượng là văn bản vì nội dung thông tin dễ dàng được truy xuất, các loại đối tượng khác thì gặp nhiều khó khăn trong truy xuất các nội dung của tài liệu.

Hệ thống IR có nội dung chính là thiết lập chỉ mục văn bản và tra cứu

Lập chỉ mục là bước đầu khảo sát, phân tích tài liệu để đưa ra các thông tin từ tài liệu và biểu diễn các thông tin theo yêu cầu của hệ thống IR. Nội dung biểu diễn thường là các từ (word) , cụm từ có nghĩa (phrase) hoặc được tổ chức thành danh sách các từ khóa có trọng số thể hiện độ ưu tiên của từ khóa.

Tra cứu là bước tìm kiếm thông tin được lưu trong CSDL có những tài liệu hoặc trích yếu có nội dung đúng với nội dung câu truy vấn. Trong quá trình tìm kiếm thông tin nội dung cần tìm kiếm được biểu diễn bằng ngôn ngữ tự nhiên hoặc một kiểu quy ước của hệ thống quy định.

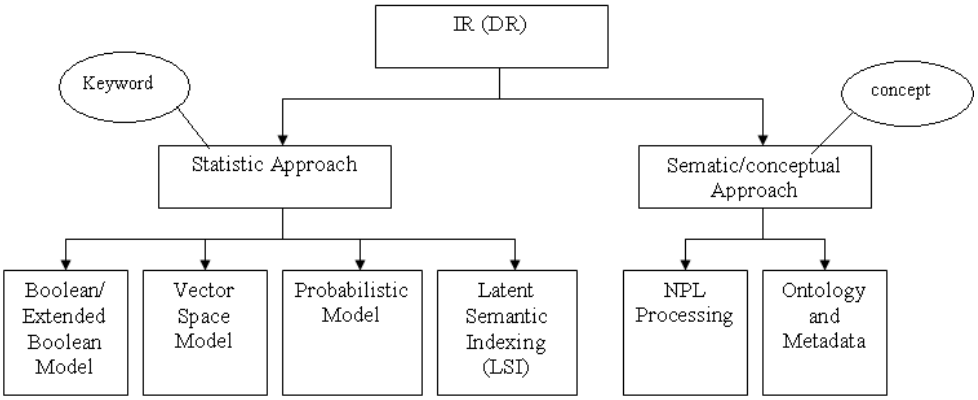
Để đánh giá một tài liệu và câu truy vấn có độ tương đồng thì Hệ thống IR sử dụng một trong hàm Cosine để đánh giá độ tương đồng của nội dung câu truy vấn và nội dung văn bản được lưu trữ trong CSDL sau đó các văn bản có độ tương đồng nằm trong ngưỡng cho phép sẽ được trả về cho người dùng

Các hệ thống IR có thể được phân loại như sau:

- Tìm kiếm theo từ khóa: Người dùng sẽ nhập vào từ khóa có thể là tên viết tắt hoặc từ/cụm từ để mô tả yêu cầu cần tìm kiếm. Hệ thống dựa vào yêu cầu và tìm trong các nội dung tài liệu kiểm tra có tài liệu nào thỏa mãn yêu cầu có thể loại trừ các từ hay sử dụng (stopword) để kết quả trả về được chính xác hơn. Như vậy một hệ thống nếu được xem hoạt động tốt khi số lượng kết quả trả về phải có độ tương đồng tương đối cao khi so sánh văn bản với nội dung truy vấn có nhiều nội dung trùng khớp nhau thì sẽ được trả kết quả cho người dùng. Hiện nay các mô hình truy tìm thông tin được sử dụng cho từ khóa như: mô hình Boolean, mô hình không gian vector, các mô hình xác suất…
- Tìm kiếm dựa trên khái niệm hay ngữ nghĩa: Hệ thống mô tả câu truy vấn hoặc nội dung tài liệu bằng các tập khái niệm hay sử dụng các công nghệ lưu trữ thông tin có cấu trúc về từng lĩnh vực cụ thể. Công cụ sử dụng chủ yếu trong mô hình này tập trung vào xử lý ngôn ngữ tự nhiên và công nghệ mạng ngữ nghĩa ontology.

2.1.2 Các phương pháp truy hồi thông tin

Hiện tại trên thế giới có nhiều hướng nghiên cứu cho hệ thống IR, điển hình thì có hai hướng cơ bản cho việc nghiên cứu IR là thống kê theo từ khóa và mạng ngữ nghĩa. Trong phương pháp sử dụng từ khóa để đánh giá kết quả trả về thì hệ thống dựa vào các tài liệu được xếp hạng cao đến thấp, ưu tiên các tài liệu cao được trả về. Trong khi đó phương pháp nghiên cứu theo mạng ngữ nghĩa lại tập trung phân tích nội dung, cú pháp, ngữ nghĩa của thông tin để truy tìm các nội dung tìm ẩn của câu truy vấn và nội dung của các tài liệu. Mô phỏng cấu trúc để máy và con người có thể hiểu với nhau, có thể tham khảo thêm trong các tài liệu, bài báo [2][7]



Hình 2.1: Các phương pháp truy hồi thông tin

- **Truy tìm thông tin theo hướng tiếp cận thống kê**

Các mô hình truy tìm thông tin theo từ khóa hiện đang được nghiên cứu theo hướng thống kê bao gồm các mô hình: mô hình Boolean, Boolean mở rộng (extended Boolean), mô hình không gian Vector (Vector space model), các mô hình xác xuất (Probabilistic models).

Ý tưởng của phương pháp này là sử dụng một danh sách các thuật ngữ trong tài liệu hay câu truy vấn là một dạng biểu diễn nội dung của câu truy vấn và tài liệu đó. Khi một thuật ngữ của tài liệu được chọn thì phải mã hóa theo mô hình toán học để máy tính có thể xử lý được.

2.1.2.1 Mô Hình Boolean

Boolean là một mô hình đơn giản dễ cài đặt sử dụng trong các mô hình truy hồi thông tin được sử dụng cho các hệ thống không yêu cầu nghiêm ngặt về nội dung dữ liệu.

Mô hình Boolean được tính toán bằng đại số boolean và tập hợp trong toán học nên cài đặt đơn giản, dễ sử dụng và thời gian tìm hiểu nhanh chóng. Với mô hình này, mỗi văn bản được trình bày bởi một vector nhị phân, vector chỉ có hai giá trị {0,1}, nếu từ khóa thứ k được tìm thấy trong văn bản V_l trọng số được xác định là $W_{kl} = 1$, nếu không tồn tại trong văn bản thì $W_{kl} = 0$.

Các phép toán logic "AND, OR, NOT" được sử dụng để biểu diễn nội dung câu truy vấn khi muốn tìm kiếm ngữ nghĩa chính xác. Ví dụ câu truy vấn muốn tìm là

"q_1 or q_2" thì trong văn bản phải thỏa điều kiện tồn tại một trong hai nội dung thỏa q_1 hoặc nội dung thỏa q_2.

Mô hình Boolean sử dụng trạng thái "đúng hoặc sai" để diễn đạt từ khóa có tồn tại trong văn bản hay không. Nếu từ khóa được ghi nhận là "đúng" tương ứng có tồn tại trong văn bản đang xét, "sai" thì từ khóa không nằm trong nội dung văn bản. Do mô hình Boolean chỉ xét từ khóa có tồn tại trong văn bản hay không nên kết quả tìm kiếm không thể tìm được các kết quả có nội dung liên quan

Ví dụ: khi xét câu truy vấn Query = q_1 AND (q_2 OR q_3)

Giả sử khi văn bản chỉ chứa nội dung q_3 nhưng không có chứa nội dung q_1 thì kết quả tìm kiếm sẽ trả về không tìm thấy nội dung, do q_1 không thỏa điều kiện "and" trong câu truy vấn. Nếu nội dung văn bản chứa q_1 và tồn tại một trong hai nội dung q_2 hoặc q_3 thì kết quả sẽ tìm thấy nội dung trả về

Một số cải tiến về việc sử dụng mô hình Boolean vào các hệ thống IR:

- Thứ nhất, mô hình boolean sử dụng cho các văn bản có yếu tố cụ thể như tác giả, tóm tắt, trích yếu thì chính xác hơn là sử dụng cho toàn bộ nội dung văn bản
- Thứ hai, khi sử dụng toán tử AND, OR, NOT thì có sử dụng thêm toán tử proximity để tính độ khoảng cách gần của nội dung trong văn bản. Toán tử này cho biết các từ nằm trong cùng văn bản và có đứng gần nhau không và có giá trị bắt đầu từ 0 là đứng gần nhau.
- Thứ ba, mô hình boolean dùng các toán tử AND, OR, NOT để thực hiện phép toán kết hợp hoặc đồng nghĩa của một từ hoặc cụm từ trong văn bản. Ví dụ về sự kết hợp của hai truy vấn q_1="Trường THPT", q_2="Tây ninh", nếu "q_1 AND q_2" thì câu có nghĩa là "Trường THPT Tây ninh" mô tả một danh từ là Trường học có vị trí ở tỉnh Tây ninh, nhưng đứng một mình thì q_1 mô tả về một trường học ở một nơi không xác định , q_2 thì mô tả tên của một tỉnh. Ví dụ về phép toán đồng nghĩa q_1="kế thừa", q_2="thừa hưởng", khi "q_1 OR q_2" thì hai kết quả đều có nội dung tương tự nhau về mặt ngữ nghĩa nhưng khác nhau về cách thể hiện từ ngữ trong văn bản

- Như vậy tùy vào từng hệ thống phục vụ cho nhu cầu gì mà chúng ta sử dụng phép toán boolean cho đúng với nhu cầu thực tế hệ thống.

2.1.2.2 Mô hình Boolean nâng cao (Advanced Boolean Model)

Khi đánh giá mô hình boolean có sử dụng proximity thì kết quả của mô hình cũng trả về một trong hai giá trị là đúng hoặc sai. Do do khi ứng dụng vào thực tế có trường hợp kết quả trả về có nhiều văn bản liên quan hoặc không có trả về văn bản nào liên quan. Khi toán tử OR được sử dụng nhiều trong câu truy vấn có nhiều thực thể liên kết với nhau dẫn đến tình trạng một văn bản chứa gần như tất cả các thực thể đó cũng có thể không chính xác bằng một văn bản mà chỉ chứa một thực thể chính. Trong trường hợp toán tử AND cũng không phải là trường hợp lý tưởng vì nếu sử dụng toán tử AND cho nhiều thực thể của câu truy vấn thì điều kiện này chỉ đúng khi tất cả thực thể đều tồn tại trong văn bản nếu không thì sẽ không có kết quả trả về. Do đó khi sử dụng toán tử cần chú ý sử dụng vừa phải tránh trường hợp làm dư thừa thông tin hoặc mất mát thông tin không cần thiết.

Từ những vấn đề đã nêu, Đã có nhiều nghiên cứu về việc phát triển các mô hình boolean mở rộng khác nhau để đánh giá được kết quả trả về, có thể sử dụng cùng lúc nhiều toán tử mở rộng để tăng khã năng độ chính xác. Một trong các phương pháp sử dụng của mô hình mở rộng là thay đổi giá trị của hàm boolean thay vì chỉ trả về hai giá trị 0 hoặc 1 thì kết quả trả về sẽ có giá trị từ 0 đến 1 tương ứng với độ tương đồng giữa biểu thức và văn bản. Ví dụ mô hình p-norm là một mô hình sử dụng theo phương pháp biểu diễn giá trị từ 0 đến 1 [22].

Ưu điểm của mô hình Boolean:

- Đơn giản, dễ hiểu, dễ cài đặt và sử dụng.
- Mô hình lý thuyết chặt chẽ, rõ ràng.
- Khã năng mở rộng mô hình cao, dễ dàng thực hiện cải tiến mở rộng.

Nhược điểm:

- Mô hình chỉ có hai trạng thái "Đúng/Sai" nên kết quả trả về một là dữ liệu trả về quá nhiều so với nội dung cần tìm, hai là không có dữ liệu nào liên quan đến câu truy vấn được trả về. Do đó, hiệu quả truy tìm không cao.

- Mối liên hệ giữa các từ hoặc cụm từ hay vị trí giữa chúng không kiểm tra.
- Không đánh giá được thứ tự ưu tiên, độ tương đồng giữa nội dung truy vấn và văn tài liệu
- Việc xây dựng các mô hình boolean để biểu diễn các nội dung cho người dùng cũng là một vấn đề phức tạp. Đòi hỏi người dùng phải hiểu biết về kỹ thuật phân tích và suy luận logic.

Sau khi có nhiều mô hình cải tiến mô hình boolean thành các mô hình mở rộng mới cũng đã giải quyết được một vài điểm yếu của mô hình, nhưng nhìn chung mô hình Boolean vẫn còn nhiều bất cập do đó cần có một mô hình khác để xử lý tối ưu mô hình hiện tại với yêu cầu sẽ xem xét độ tương đồng của câu truy vấn và tài liệu để đạt kết quả tốt hơn.

2.1.2.3 Mô Hình Không Gian Vector (VSM)

Mô hình VSM khắc phục những hạn chế của mô hình boolean bằng cách đánh trọng số cho đối tượng đặc trưng. Trọng số đối tượng đặc trưng không giới hạn bởi hai trị 0 hoặc 1, các trọng số này được sử dụng để tính toán độ đo tương đồng của mỗi văn bản với câu truy vấn.

Mô hình VSM sẽ mô tả mỗi tài liệu văn bản thành một tập hợp vector các đối tượng có tồn tại trong tất cả tài liệu văn bản, mỗi đối tượng được quy định là một chiều của không gian đó, tổng hợp tất cả các chiều của đối tượng trong không gian được gọi là không gian tài liệu[1].

Mỗi đối tượng sẽ được gán thêm trọng số để quản lý đối tượng đó trong không gian, và chỉ có giá trị trong tập các tài liệu đang xử lý. Mỗi đối tượng được đánh trọng số riêng biệt trong mỗi tài liệu khác nhau. Giá trị trọng số thể hiện tầm quan trọng của đối tượng đó trong tài liệu hay trích yếu, nó phản ánh nội dung mà tài liệu đang xem xét.

Nội dung của một đối tượng trong tài liệu nó có thể là nội dung chính trong tài liệu này nhưng chỉ là nội dung không cần thiết của tài liệu khác và sẽ được đánh giá trị 0 để cho biết đối tượng này không tồn tại trong tài liệu đang xét.

Mỗi đối tượng có trọng số được gán trong không gian tài liệu được xem như là

một tọa độ của tài liệu đó, một vector tài liệu được biểu diễn là vector đi từ gốc tọa độ đến một tọa độ đối tượng trong không gian tài liệu.

Khi tài liệu được mô tả bằng không gian tài liệu thì câu truy vấn cũng được mô tả thành các đối tượng được gán các trọng số tương ứng, các đối tượng này được xem như là một vector truy vấn trong không gian tài liệu đang xét.

Ưu điểm của mô hình không gian vector:

- Đơn giản, dễ hiểu, dễ cài đặt.
- Thực hiện xếp hạng theo độ đo cosin các văn bản theo mức độ liên quan.
- Giải quyết được hạn chế không tìm theo ngữ nghĩa liên quan của mô hình boolean

Nhược điểm:

- Các từ khóa biểu diễn được xem là độc lập với nhau.
- Vector không gian tài liệu sẽ tăng lên khi số lượng tài liệu lớn và làm chậm thời gian xử lý của không gian vector

2.1.2.4 Mô Hình Xác Suất (Probability Model)

Mô hình xác suất là một biểu diễn toán học của một hiện tượng ngẫu nhiên. Nó được xác định bởi không gian mẫu, các sự kiện trong không gian mẫu và xác suất liên quan đến mỗi sự kiện.

Giả sử cho tài liệu văn bản d trong tập văn bản D và câu truy vấn là q thì mô hình xác suất để tính độ tương đồng giữa văn bản d và câu truy vấn q sử dụng công thức tính xác suất để tính giả thuyết liên quan giữa câu truy vấn và văn bản. Danh sách văn bản kết quả liên quan là các văn bản có tổng xác suất liên quan với câu truy vấn lớn nhất.

Ưu điểm của mô hình xác suất:

- Dễ dàng đánh giá tài liệu dựa vào xác suất khi đã được xếp hạng
- Khã năng tìm kiếm của mô hình xác suất có tốc độ tìm kiếm nhanh hơn khi không dùng xác xuất.

Nhược điểm:

- Không thể biểu diễn thông tin ngữ nghĩa về một tài liệu theo công thức xác

suất.

- Không đánh giá trọng số xuất hiện của đặc trưng trong tài liệu.
- Giả định các từ khóa biểu diễn độc lập nhau.
- Phải chia tập tài liệu được chia thành 2 loại: thích hợp hay không thích hợp.

2.1.3 Đánh giá một hệ thống tìm kiếm thông tin

Một hệ thống IR được đánh giá hiệu quả khi thỏa mãn hai độ đo cơ bản là độ chính xác (Precision) và độ bao phủ (Recall).

So với các độ đo khác, hai độ đo này đáp ứng các yêu cầu cơ bản của hệ thống truy xuất thông tin. Các chỉ số này đo lường mức độ hài lòng của người dùng đối với các tài liệu do hệ thống tìm thấy.

Gọi S là tập các tài liệu tìm được (tài liệu liên quan đến hệ thống).

Gọi U là tập hợp các tài liệu liên quan theo đánh giá của người dùng. Sau đó, độ chính xác và độ bao phủ sẽ được xác định như sau [23]:

Độ chính xác: là sự tương ứng giữa số tài liệu mà hệ thống tìm thấy có liên quan đến câu truy vấn theo người dùng trên tổng số các tài liệu tìm thấy của hệ thống.

$$\text{Độ chính xác } = \frac{|S \cap U|}{|S|} \tag{2.1}$$

Độ chính xác 100% toàn bộ tài liệu được lưu trữ trong kho được tìm thấy của hệ thống đều đáp ứng nhu cầu tìm kiếm của người dùng.

Độ bao phủ: là với tất cả số lượng tài liệu được lưu trữ trong kho hệ thống có liên quan đến người có được tìm thấy hết không khi người dùng thực hiện tìm kiếm. Nếu số lượng trả về tìm kiếm không bao gồm toàn bộ tài liệu liên quan trong hệ thống thì độ bao phủ chưa đạt 100%.Ngược lại nếu trả về toàn bộ tài liệu liên quan thì độ bao phủ đạt 100%.

$$\text{Độ bao phủ } = \frac{|S \cap U|}{|U|} \tag{2.2}$$

Đối với bất kỳ hệ thống nào thì độ chính xác và độ bao phủ luôn tỷ lệ nghịch với nhau vì khi tăng độ chính xác lên thì độ bao phủ giảm xuống và ngược lại. Do đó khi thiết kế hệ thống tùy vào chức năng mà ta lựa chọn tiêu chí đánh giá cho phù hợp

2.2 Ontology

2.2.1 Định nghĩa

2.2.1.1 Trong triết học

Ontology (Bản thể học) là sự tra vấn triết học về bản tính nền tảng của hiện hữu, thực tại, tồn tại. Các triết gia khác nhau tán thành những bản thể học khác nhau vì họ có những quan điểm khác nhau về cái đang tồn tại ở cấp độ nền tảng hay phổ biến nhất. Bản thể học của Descartes, chẳng hạn, bàn về các tinh thần, vật chất và Thượng đế, trong khi đó bản thể học của Sartre lại bàn về tồn tại và sự phủ định của nó, không tồn tại hay hư vô. Bản thể học đôi khi được mô tả là một nhánh của siêu hình học, nhưng trên thực tế nó là thuật ngữ rộng hơn siêu hình học ở chỗ có hữu thể học siêu hình học và hữu thể học phi siêu hình học. Các hữu thể học siêu hình học, lý thuyết của Plato về các mô thức chẳng hạn, cho rằng về cơ bản thực tại chính là các ý niệm hay các bản chất, trong khi đó các hữu thể học phi siêu hình học, thuyết hiện sinh của Sartre chẳng hạn, cho rằng sự hiện hữu của tồn tại là nền tảng. Xem thêm hiện hữu đi trước bản chất, thuộc bản thể học và cấp độ bản thể học[18].

2.2.1.2 Trong lĩnh vực Trí tuệ nhân tạo

Trong Trí tuệ nhân tạo ontology cũng đã có rất nhiều định nghĩa khác nhau từ nhiều nhà nghiên cứu trên thế giới, một số khái niệm được xem là kinh điển và được công nhận rộng rãi như:

- Gruber (1993) định nghĩa "một ontology giống như một khai báo tường minh của khái niệm hóa trong một miền tri thức".
- Borst (1997) về định nghĩa có thay đổi so với Gruber cho rằng "ontology là mô tả hình thức của khái niệm hóa được chia sẽ trong các miền tri thức"
- Studer (1998) tổng hợp hai định nghĩa của Gruber và Borst và giải thích chi tiết về hai định nghĩa cụ thể như sau "Sự khái niệm hóa được hiểu có nghĩa là mô tả các mô hình dữ liệu trừu tượng của các sự vật, hiện tượng trên thế giới được xác định qua các khái niệm liên quan của sự vật, hiện tượng đó. Sử dụng tường minh có nghĩa là các kiểu khái niệm và các mối quan hệ giữa chúng là được xác định rõ ràng. Hình thức có nghĩa là ontology phải

được xây dựng trên ngôn ngữ mà có thể giao tiếp được với máy tính. Chia sẻ có nghĩa là tri thức trong mô hình ontology có thể được kết hợp xây dựng với các ontology trong miền tri thức khác và được công nhận bởi một nhóm hoặc một cộng đồng chứ không theo tri thức chủ quan của cá nhân".

- Motta (1999) định nghĩa "ontology là sự mô tả một phần của một tập các khái niệm được khai báo hình thức hóa các thông tin của một miền tri thức cần quan tâm. Vai trò chính của một ontology là có thể chia sẽ và sử dụng qua lại của các miền tri thức khác nhau".
- Uschold và Jasper (1999) thì cho rằng "ontology bao gồm các định nghĩa và các mối quan hệ giữa các định nghĩa, hình thành nên một cấu trúc của miền tri thức và giới hạn ngữ nghĩa của thuật ngữ trong mô hình dữ liệu".
- Weiss (1999) định nghĩa "ontology là một mô tả của các khái niệm và các mối quan hệ trong miền tri thức quan tâm. Ontology không chỉ tổ chức mô hình phân cấp các lớp mà còn mô tả các quan hệ giữa các lớp trong mô hình phân cấp".
- Hendler (2001) phát biểu "ontology là một tập hợp các thuật ngữ tri thức (knowledge term), bao gồm các lớp phân cấp, thực thể có quan hệ ngữ nghĩa với nhau, có thể áp dụng các luật suy diễn vào các lớp để tạo ra mối quan hệ giữa các thực thể và có tính logic trong một miền tri thức cụ thể".

Nhìn chung, định nghĩa về ontology thì qua mỗi thời điểm có các khái niệm, các định nghĩa thể hiện một cách nhìn khác nhau về mô hình dữ liệu và đi cùng với khái niệm là một phương pháp luận và kỹ thuật xây dựng mô hình dữ liệu ontology. Một định nghĩa mang tính tổng hợp và đúng theo định hướng xây dựng ontology của như sau: Một ontology mô tả một tập từ vựng chung và có thể chia sẽ thông tin qua lại trong một miền tri thức, bao gồm định nghĩa của các khái niệm cơ bản mà có thể giao tiếp được máy tính để có thể hiểu được trong một lĩnh vực nào đó và có mối liên hệ giữa chúng.

2.2.2 Các thành phần của ontology

Ontology thông thường được thiết kế từ các thành phần như :

- Class là các lớp trong mô hình dữ liệu thể hiện các khái niệm trừ tượng là dữ liệu cốt lõi trong các mô hình ontology trong miền tri thức. Các lớp thường được thiết kế theo mô hình phân cấp lớp cha – con có sử dụng nguyên tắc kế thừa trong thiết kế. Những lớp con có thể biểu diễn nội dung chi tiết cụ thể hơn so với các lớp cha.
- Properties là thuộc tính mô tả các tính chất đặc trưng khác nhau của khái niệm và đều được gán giá trị cho nó. Thuộc tính so với quan hệ thì có sự khác nhau dựa trên giá là kiểu dữ liệu. Một thuộc tính thì cũng bao gồm các thuộc tính con và mối quan hệ ràng buộc giữa chúng.
- Relation là quan hệ mô tả các mối quan hệ giữa các khái niệm trong mô hình dữ liệu ví dụ quan hệ về từ đồng nghĩa, quan hệ dẫn xuất…Các quan hệ nhị phân dùng cho mô tả các thuộc tính của các lớp. Tuy nhiên, quan hệ có giá trị là các khái niệm còn thuộc tính có giá trị là các kiểu dữ liệu do đó hai giá trị của quan hệ và thuộc tính khác nhau hoàn toàn.
- Instance là thực thể dùng để mô tả các phần tử riêng biệt của khái niệm, là các thể hiện của lớp. Mỗi thể hiện của lớp biểu diễn một sự cụ thể hóa của khái niệm đó.
- Function là một đối tượng đặc biệt trong mô hình dữ liệu, nó có tính duy nhất trong mô hình và được sử dụng như là một thuộc tính hoặc một quan hệ nào đó
- Axioms là khẳng định (bao gồm cả các quy tắc) ở dạng logic cũng bao gồm lý thuyết tổng quát mà ontology mô tả trong miền tri thức của ứng dụng[3]. Định nghĩa này khác với định nghĩa của "tiên đề" trong ngữ pháp tổng quát và hình thức logic. Trong những lĩnh vực này, tiên đề chỉ bao gồm những phát biểu được khẳng định là tri thức luôn đúng. Như được sử dụng ở đây, "tiên đề" cũng bao gồm lý thuyết bắt nguồn từ các phát biểu tiên đề.

2.2.3 Phân loại ontology

Về cơ bản có các loại ontology sau:

- Ontology biểu diễn tri thức (Knowledge representation Ontology) sử dụng

các nguyên tắc xử lý nguyên thủy được dùng để chuẩn hóa dữ liệu trong mô hình ontology, Frame Ontology của Gruber là loại mô hình ontology áp dụng theo phương thức này như frame, slot và mối quan hệ giữa các slot cho phép các tri thức frame cơ sở hoặc hướng theo đối tượng của tri thức.

- Ontology tổng quát (Generic Ontology) là tập gồm các nội dung liên quan với nhau về các sự vật, hiện tượng,…có nội dung tổng quát bao hàm nhiều lĩnh vực, các mô hình dữ liệu khác có thể kế thừa dữ liệu từ mô hình tổng Knowledge representation Ontology quát này sử dụng. Ví dụ mô hình tổng quát WordNet tổng hợp tất cả các từ vựng về tất cả các lĩnh vực khác nhau.
- Metadata ontology theo một định nghĩa nổi tiếng từ năm 1992, là "đặc tả của một khái niệm hóa" - một cái nhìn trừu tượng, đơn giản hóa về thế giới được biểu thị cho các mục đích từ chia sẻ kiến thức đến ra quyết định. Mặc dù các ontology thường được cho là phức tạp về mặt ngữ nghĩa, với các tập hợp phong phú các mối quan hệ giữa các khái niệm và tiên đề của nó được thiết kế để hỗ trợ việc tham khảo. ví dụ về mô hình dữ liệu về dịch vụ từ điển đồng nghĩa của Phần Lan, các lược đồ khái niệm Simple Knowledge Organization System (SKOS), và mô hình dữ liệu về từ vựng trực tuyến của Dublin Core [17].
- Ontology miền (Domain Ontology) là những ontology được xây dựng với mục đích có thể sử dụng lại trong một miền tri thức nào đó, nó cung cấp từ vựng về các khái niệm và các mối quan hệ của các thực thể trong một miền tri thức. Ví dụ: ontology về y khoa MeSH, GALEN hay ontology về sinh học Gene Ontology, OBO.
- Ontology tác vụ (Task Ontology) là mô hình sử dụng các từ thuật ngữ để thực thi một tác vụ nào đó.
- Ontology lĩnh vực - tác vụ (Domain – Tast Ontology) là mô hình dữ liệu sử dụng tác vụ phục vụ cho miền tri thức nào đó, có thể sử dụng lại nhiều lần cho tác vụ đó.
- Ontology ứng dụng (Application Ontology) dùng cho các ứng dụng cụ thể.

- Ontology chỉ mục (Index Ontology) là mô hình gán chỉ số vào danh sách nội dung của miền tri thức
- Ontology hỏi và trả lời (Tell and Ask Ontology) sử dụng cho các hệ thông chatbot dùng trong chăm sóc khách hàng…

Mô hình dữ liệu ontology được chia thành 2 nhóm chính: một nhóm mô tả các dữ liệu tri thức, biểu diễn nội dung để thể hiện miền tri thức cần xử lý như Metadata ontology, Domain ontology, Application ontology, nhóm còn lại thì tập trung vào tri thức dùng để giải quyết vấn đề như Task ontology, Domain-Task ontology. Cả hai nhóm ontology có thể kết hợp để xây dựng nên một ontology hoàn chỉnh hơn.

Ngoài ra, các ontology còn được phân loại dựa vào tính phức tạp của mô hình biểu diễn dữ liệu

- Lightweight ontology: ontology bao gồm các lớp, phân cấp lớp, mối quan hệ giữa lớp và thuộc tính mô tả của các lớp
- Heavyweight ontology: cải tiến mô hình Lightweight bằng các bổ sung vào mô hình các phát biểu tiên đề, các function và các tập luật suy diễn.

2.2.4 Vai trò của Ontology

Ontology mục đích ban đầu là tạo ra các miền tri thức gồm nhiều lĩnh vực khác nhau để có được thông tin đa dạng, phục vụ cho nhu cầu xử lý thông tin của con người cũng như máy tính có thể xử lý và thao tác được. Bên cạnh đó các mô hình dữ liệu còn có thể dùng để chia sẽ thông tin giữa các hệ thống xử lý dữ liệu với nhau.

Để các hệ thống khác nhau có thể chia sẽ dữ liệu dùng chung một mô hình thì cần có một bộ quy tắc giao tiếp chuẩn để có thể nhận biết với nhau. Do mỗi hệ thống sử dụng các khái niệm, thuật ngữ, cấu trúc riêng biệt hoặc nếu có dùng chung khái niệm thuật ngữ nhưng cách biểu diễn của mỗi hệ thống lại theo một hướng xử lý khác nhau, hoàn cảnh áp dụng các khái niệm khác nhau thì khi đó các hệ thống sẽ không giao tiếp được với nhau, việc xác định các yêu cầu, khã năng liên kết của hệ thống kém, không tái sử dụng được mô hình dữ liệu và tính chia sẽ dữ liệu giữa các hệ thống hầu như không dùng được. Do đó sẽ phát sinh thêm nhiều chi phí để kết nối các hệ thống lại với nhau, ngoài việc kết nối các hệ thống để có thể chia sẽ dữ liệu thì vấn đề

về xử lý dữ liệu cũng là một yêu cầu không thể thiếu đối với các hệ thống thông minh.

Các hệ thống này dựa trên các miền tri thức rõ ràng, phân loại được các đối tượng trong miền tri thức, tránh nhầm lẫn về ngữ nghĩa hoặc xây dựng các luật suy diễn không đúng với yêu cầu đặt ra dẫn đến hệ thống không sử dụng được. Do đó các tri thức cần phải có cơ chế tự học, phân lớp các đối tượng đúng vào miền tri thức sử dụng giảm thiểu sự nhầm lẫn, xung đột giữa cac lớp, thực thể, cung cấp tập từ vựng ngữ nghĩa chính xác hỗ trợ chia sẽ giữa các hệ thống.

Ontology là mô hình đáp ứng các yêu cầu cần thiết của hệ thống với các chức năng như sau :

- Chia sẽ thông tin trong mô hình dữ liệu giữa hệ thống khác nhau, giữa hệ thống và con người để có thể giao tiếp được với nhau.
- Cho phép sử dụng lại tri thức. Ví dụ, có thể kế thừa một ontology về tập từ vựng để xây dựng một ontology kiểm tra lỗi chính tả của từ vựng.
- Khi xây dựng mô hình ontology chú ý phân tích đầy đủ nội dung cần xây dựng về miền tri thức cụ thể. Các mối quan hệ giữa các thuộc tính và các lớp phải đặc tả rõ ràng giúp cho những người sử dụng lại hệ thống có thể dễ dàng nắm bắt, tìm hiểu các ngữ nghĩa, cấu trúc liên quan được sử dụng mô hình.
- Mô hình dữ liệu nên thiết kế theo từng loại mô hình chuyên biệt về xử lý tri thức và miền tri thức cho các lĩnh vực. Ví dụ mô hình ontology mô tả về miền tri thức của lĩnh vực về mạng internet so với mô hình xử lý sự cố về mạng.
- Một mô hình dữ liệu muốn được kế thừa sử dụng lại thì yếu tố đầu tiên là các khái niệm phải rõ ràng, chính xác, không trùng lặp các khái niệm và các mối quan hệ giữa thuộc tính và khái niệm phải chính xác. Muốn kế thừa hay sử dụng một ontology ta phải phân tích và tìm hiểu các khái niệm và quan hệ giữa chúng trong ontology đó.

Theo Aldea, các ontology có khả năng:

- Cung cấp một cấu trúc để chú giải nội dung của một tài liệu với thông tin

ngữ nghĩa, điều này cho phép trích chọn thông tin thích hợp từ những tài liệu đó.

- Tích hợp thông tin từ nhiều nguồn khác nhau nhờ cung cấp một cấu trúc cho tổ chức của nó và tạo thuận lợi cho trao đổi dữ liệu, tri thức và các mô hình.
- Đảm bảo sự đồng nhất và chính xác nhờ công thức hóa các ràng buộc nội dung của thông tin.
- Tạo các thư viện của các mô hình có khả năng trao đổi và tái sử dụng.
- Cho phép lập luận, nghĩa là cho phép tiến triển từ xử lý cú pháp đến xử lý ngữ nghĩa và cho phép các hệ thống suy diễn về các đối tượng dựa trên các luật sinh tổng quát.

2.2.5 Các ứng dụng dựa trên Ontology

Ngày nay ontology không chỉ dừng lại trong việc chia sẽ thông tin dữ liệu mà nó còn không ngừng phát triển và được áp dụng vào hầu hết các lĩnh vực khác nhau trong môi trường có liên quan đến dữ liệu điển hình như hệ thống xử lý ngôn ngữ tự nhiên, truy hồi thông tin, mua bán trên sàn thương mại điện tử, quản trị cơ sở dữ liệu, công nghệ phần mềm, mạng và an toàn bảo mật… Ontology cung cấp các miền tri thức đáp ứng đầy đủ trong các hệ thống xử lý dữ liệu của từng lĩnh vực cụ thể với kết quả tốt hơn khi không sử dụng mô hình ontology.

W3C sử dụng ontology biểu diễn miền tri thức bao gồm nhiều lĩnh vực để làm nền tảng xây dựng hệ thống web ngữ nghĩa. Web ngữ nghĩa là một sự cải tiến của web 2.0 sử dụng cấu trúc html để hiển thị nội dung, web ngữ nghĩa phát triển từ web 2.0 có mô tả thêm chi tiết về nội dung ngữ nghĩa của thông tin mà hệ thống có thể hiểu được các thông tin đó thông qua mô hình ontology giúp hệ thống làm việc tốt hơn với các yêu cầu ngày càng cao của người dùng.

Việc phát triển ontology với mục đích đầu tiên là nâng cao khã năng tìm kiếm thông tin trên hệ thống web hiện tại, với khã năng chỉ mới dừng ở việc tìm kiếm theo từ khóa, chưa đáp ứng đủ thông tin mà người dùng đặt ra. Do đó ontology được dùng để gán nhãn lại nội dung trên các trang web, nguồn thông tin được chia sẽ trên internet,

các nội dung liên quan nhằm nâng cao hiệu quả trong việc tìm kiếm nội dung trên môi trường internet.

Trong tiến trình khai phá dữ liệu hay tích hợp dữ liệu, việc ứng dụng ontology mang lại nhiều lợi thế, chẳng hạn như đối với các hệ thống bao gồm nhiều nguồn cơ sở dữ liệu khác nhau (khác nhau về cách thức lưu trữ và nội dung thông tin), mỗi nguồn dữ liệu sẽ có một ontology mô tả về nó. Các ontology đó sẽ được hợp nhất vào một ontology chung và khi người dùng đưa ra yêu cầu thì hệ thống sẽ chuyển truy vấn đến nguồn cơ sở dữ liệu tương ứng.

Hiện nay có nhiều đơn vị kinh doanh trên môi trường Internet, việc áp dụng mô hình ontology vào việc quản lý thông tin các sản phẩm được áp dụng rộng rãi. Ontology mô tả thông tin chi tiết về sản phẩm như nguồn gốc, thông tin hướng dẫn sử dụng, chức năng của sản phẩm, và những thông tin chi tiết của các sản phẩm có nội dung giống nhau thì được chuẩn hóa thành các nhóm để hỗ trợ khi người dùng tìm kiếm một sản phẩm thì có thể gợi ý các sản phẩm tương tự để người dùng lựa chọn. Có thể dựa vào hành vi tìm kiếm của người dùng có thể đưa ra các sản phẩm kèm theo sử dụng kết hợp với sản phẩm đang tìm của người dùng.

Trong lĩnh vực giáo dục đào tạo cũng xây dựng nhiều hệ thống có sử dụng mô hình dữ liệu ontology để mô tả và chia sẽ các thông tin về các tài liệu học tập, nghiên cứu. Ví dụ như hệ thống chia sẽ tài nguyên trực tuyến thegateway.org, hệ thống chia sẽ ngang hàng www.edutella.org, hệ thống learning iis.fon.bg.ac.yu, các hệ thống giáo dục này chủ yếu sử dụng mô hình ontology vào biểu diễn và lưu trữ các nội dung của tri thức cụ thể, xây dựng mô hình biểu diễn tài liệu, lập chỉ mục cho tài liệu và cuối cùng xây dựng phương pháp tìm kiếm theo ngữ nghĩa liên quan đến nội dung tài liệu.

2.2.6 Các hướng tiếp cận xây dựng ontology

Trong những ngày đầu mô hình ontology mới phát triển thì việc xây dựng mô hình dữ liệu đa phần thực hiện bằng thủ công, việc thực hiện này tốn rất nhiều thời gian vào thu thập dữ liệu, phân tích nội dung dữ liệu tạo ra tập dữ liệu và xây dựng mối liên hệ giữa chúng. Sau đó việc xây dựng mô hình có sử dụng các công cụ tự động hoặc bán tự động với sự giám sát của con người giúp giảm thiểu thời gian thực

hiện nhưng vấn đề đặt ra là phụ thuộc vào thuật toán mà công cụ sử dụng, và nguồn dữ liệu mà công cụ sử dụng cho từng lĩnh vực cụ thể. Do đó chất lượng của mô hình áp dụng tự động hoặc bán tự động phụ thuộc hoàn toàn vào công cụ xử lý.

Một trong nhưng phương pháp xây dựng ontology thông dụng hiện nay là rút trích thông tin nội dung từ các nguồn dữ liệu khác nhau như từ internet. Kỹ thuật xử lý được áp dụng để rút trích thông tin nội dung bằng vào việc áp dụng phương pháp học máy, xử lý ngôn ngữ tự nhiên và phương pháp đơn giản nhất là thống kê theo từ khóa.

Phương pháp xử lý ngôn ngữ tự nhiên áp dụng trong một lĩnh vực tri thức cụ thể từ đó rút trích ra các từ vựng, tìm hiểu ngữ pháp của nội dung, từ đó đưa ra các khái niệm và dựa vào mối liên hệ của các ngữ pháp và từ vựng để hình thành nên mối quan hệ giữa các khái niệm trong tập nội dung xử lý

Phương pháp thống kê là phương pháp đơn giản nhất trong các phương pháp xây dựng ontology, có nhược điểm là tốn nhiều thời gian và dữ liệu khó xây dựng mối quan hệ giữa các khái niệm.

Phương pháp sử dụng học máy áp dụng các mô hình phân lớp, xử lý dữ liệu nhằm đưa ra các đặc trưng của dữ liệu, các tập luật được phát hiện trong dữ liệu từ đó tiến hành xây dựng các mô hình ontology dựa trên các dữ liệu đặc trưng và các tập luật được tìm thấy.

Nguồn dữ liệu để xây dựng mô hình ontology thì có rất nhiều nguồn đa dạng từ internet như nội dung từ website, nội dung văn bản, hình ảnh, video… Để xây dựng mô hình có nội dung chính xác thì chúng ta phải biết chọn lọc nguồn dữ liệu chính xác, nguồn gốc rõ ràng cho phù hợp với nội dung cần xây dựng.

Các hệ thống xây dựng ontology có thể sử dụng dữ liệu từ nhiều nguồn khác nhau để xây dựng nên ontology, có thể được phân chia thành các loại sau đây:

- Dữ liệu có cấu trúc: Hệ thống xây dựng lên các ontology dựa vào các dữ liệu có cấu trúc như từ database schema, từ những ontology đã có sẵn, từ những cơ sở tri thức và từ các mạng từ vựng như WordNet.
- Dữ liệu bán cấu trúc: đây cũng là một nguồn khác mà các hệ thống thường

sử dụng, bao gồm các từ điển, các văn bản HTML và XML.

- Dữ liệu không có cấu trúc: đây là nguồn dữ liệu khó rút trích tri thức nhất.

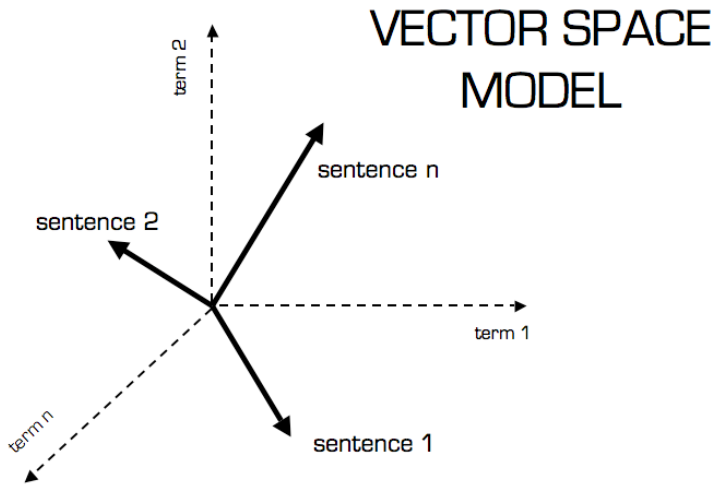
2.3 Mô hình Không gian Vector (VSM)

2.3.1 Giới thiệu

Vector space model (Mô hình không gian vector) là một mô hình đại số (algebraic model) thể hiện thông tin văn bản như một vector, các phần tử của vector này thể hiện mức độ quan trọng của một từ và cả sự xuất hiện hay không xuất hiện (Bag of words) của nó trong một tài liệu [16].

Với mỗi truy vấn, hệ thống tìm kiếm sẽ sử dụng một độ đo Rel(q, d) [15] để tính độ tương đồng giữa truy vấn (query) đó với các tài liệu (docs), từ đó xếp hạng được kết quả trả về.

2.3.2 Mô hình không gian Vector



Hình 2.2: Mô hình VSM [16]

Ý tưởng của Vector Space Model là biểu diễn văn bản và các câu truy vấn dưới dạng Vector, Rep(d) của docs và Rep(q) của query sẽ cho kết quả là các vector.

Sau đó tính độ tương đồng của query với từng văn bản theo công thức *Sim(Rep(q), Rep(d))* để tìm ra docs nào phù hợp nhất với query [15].

Biến đổi các queries và docs thành dạng vector như sau:

$$dj = (w1,j, w2,j, \ldots, wt,j)$$

$$q = (w1,q, w2,q, \ldots, wn,q)$$

Từ đó sử dụng một độ đo khoảng cách trên vector q và dj để xếp hạng các văn bản.

Các truy vấn và văn bản được biểu diễn cùng một định dạng với nhau (cùng kích thước, cách biểu diễn trọng số).

Relevance(d,q) ⇔ Similarity(d,q)

Similarity(d,q) có thể được tính bằng bất cứ độ đo trên vector nào (Euclidean, Consine Similarity, …)

R(q)={d∈C | rel(d,q)>θ, rel(d,q)=Δ(Rep(q),Rep(d))}

Kết quả của câu truy vấn được tính bằng cách tìm ra độ tương đồng của 2 hàm biểu diễn câu truy vấn và tài liệu [15].

CHƯƠNG 3: MÔ HÌNH VÀ GIẢI PHÁP

3.1 Giới thiệu hệ thống Tic-Office

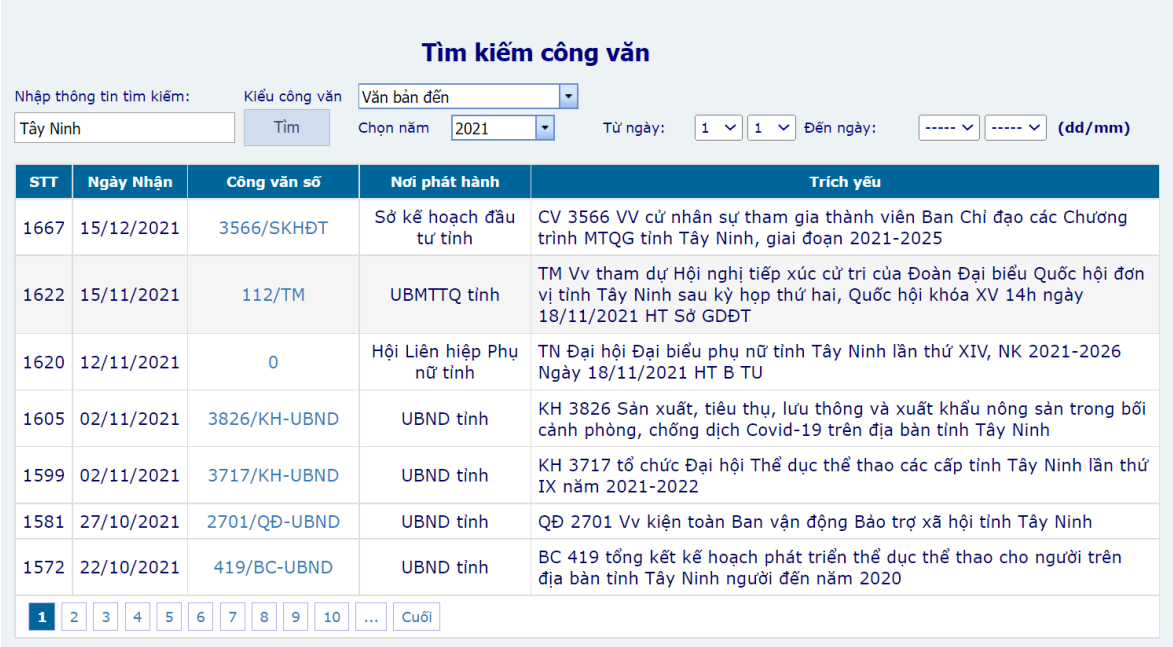
Các chức năng của hệ thống

Chức năng quản lý văn bản: Hệ thống sẽ quản lý được văn bản gửi đến và văn bản chuyển đi của Hội Nông Dân từ các Sở ban ngành, huyện, thành phố trong tỉnh. So với các hệ thống quản lý văn bản khác thì hệ thống Tic-Office chỉ có một số chức năng cơ bản liên quan đến xử lý, điều hành văn bản. Hệ thống tập trung chủ yếu vào ba chức năng chính như:

- Quản lý văn bản đến từ các cơ quan hành chính khác được giới thiệu trong Hình 3.1
- Quản lý văn bản đi từ Hội nông dân đến các đơn vị huyện… được giới thiệu trong Hình 3.2
- Tra cứu tìm kiếm văn bản, chức năng chỉ mới dừng lại ở mức độ tìm kiếm theo từ khóa trên trích yếu hoặc số văn bản, được giới thiệu trong Hình 3.3

Hình 3.1: Chức năng quản lý văn bản đến

Hình 3.2: Chức năng quản lý văn bản đi



Hình 3.3: Chức năng tra cứu văn bản theo từ khóa

3.2 Mô hình ontology cho ngữ nghĩa của câu truy vấn

Trong đề tài nghiên của của tác giả [14] đã trình bày mô hình CK_ONTO (Classed Keyphrase based Ontology) để xây dựng một hệ thống tra cứu theo ngữ nghĩa của tài liệu và tính toán độ tương đồng của tài liệu và câu truy vấn. Tác giả sử dụng mô

hình CK_ONTO đầy đủ để biểu diễn nội dung của các tài liệu cần truy vấn. Trong đề tài này tôi sử dụng mô hình CK_ONTO rút gọn để biểu diễn nội dung của câu truy vấn, mô hình gồm 3 thành phần:

$$\mathbf{(C, K, R_{KK})}$$

Trong đó:

- **K: Một tập hợp các keyphrase**

Keyphrase cụm từ khóa là một tập hợp các từ riêng biệt tạo thành một cụm từ, là một trong các yếu tố để tạo thành các khái niệm, ngoài ra keyphrase còn có nghĩa là đơn vị ngôn ngữ biểu diễn như một từ, cụm từ, một ngữ. Nói cách khác keyphrase được sử dụng như là các từ, cụm từ, các thuật ngữ chuyên ngành dùng để diễn đạt các khái niệm khoa học dùng trong các tài liệu hướng dẫn nghiên cứu khoa học.

Keyphrase là bộ phận cơ bản và chủ yếu của mô hình ontology. Về mặt cấu tạo thì keyphrase được chia thành nhóm: keyphare đơn và keypharse kết hợp. Keyphrase đơn là những keyphrase chỉ mô tả cho một khái niệm, được tạo thành từ một từ vựng đơn là từ, hoặc một từ tương đương như cụm từ cố định. Ví dụ: Cước, thuế, phí, quyền sở hữu... Keyphrase kết hợp bao gồm nhiều từ đơn kết hợp thành, được xây dựng theo cách kết hợp các keyphrase đơn lại, mà giữa các keyphrase có quan hệ ngữ nghĩa với nhau. Dựa vào nội dung ý nghĩa của mối quan hệ giữa các keyphrase kết hợp ta có thể chia thành hai loại như sau:

- Đẳng lập: Keyphrase kết hợp lại với nhau nhưng có ý nghĩa quan hệ ngang hàng nhau và có cú pháp đơn giản là dùng các liên từ kết hợp lại như “và”, “với”, “bởi”…

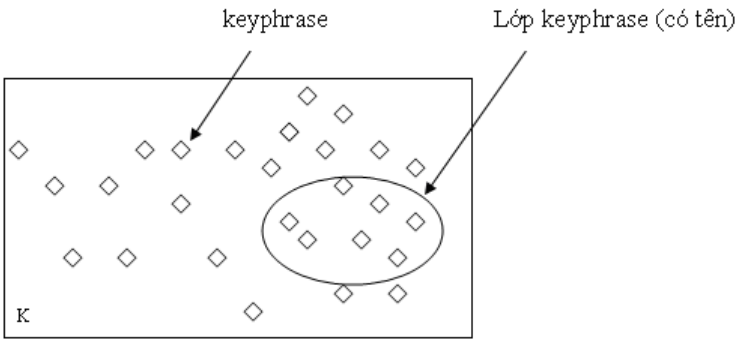
  Ví dụ: du lịch và dịch vụ, công nghệ và truyền thông…
- Chính phụ: Những keyphrase có yếu tố hình thành này phụ thuộc vào yếu tố hình thành kia, yếu tố phụ có nhiệm vụ phân loại, tách biệt cho yếu tố chính, biểu hiện các thuộc tính, đặc tính, đặc trưng.
- Ví dụ: không gian tài liệu, truy hồi thông tin, lên kế hoạch…

Như vậy, gọi **K** = {k| k là keyphrase thuộc về phạm vi đang xét}, $K = K_1 \cup K_2$, trong đó K_1 là tập các keyphrase đơn và K_2 là tập các keyphrase kết hợp.

- **Một tập hợp C các lớp keyphrase**

Mỗi lớp keyphrase c ∈ C là một nhóm các keyphrase có mối liên hệ với nhau theo một yếu tố hay ngữ nghĩa nào đó. Chúng có thể chứa các keyphrase, các lớp khác, hay là sự kết hợp của cả hai.

Trong mô hình dữ liệu Ontology nói chung thì các lớp trong mô hình được thiết kế mở có thể tạo được các mối quan hệ phân cấp cha-con tương ứng với nhiều lớp khác nhau ví dụ lớp A đang là con của lớp B, và lớp C con lớp B nhưng có thể gán lớp C làm lớp con của A. Do vậy các lớp có thể được gán vào bất kỳ lớp cha nào nếu phù hợp. Một keyphrase có thể thuộc nhiều lớp khác nhau. Sự phân lớp trong K được phân thành nhiều cấp theo mức độ cụ thể của khái niệm tăng dần. Xây dựng được một tập hợp lớp tốt sẽ tạo nên một hệ thống tốt, tuy nhiên việc phân lớp các keyphrase khi phân tích và mô tả một miền tri thức không phải là việc đơn giản, không có một phương pháp hoàn chỉnh để tìm lớp. Trong phạm vi nghiên cứu, dựa trên ngữ nghĩa của keyphrase, của các lớp chủ đề, việc gán keyphrase vào một (hay một số) lớp chủ đề thích hợp được thực hiện thủ công với các kỹ thuật điều khiển bằng tay dưới sự giám sát và ý kiến của một số chuyên gia tri thức về lĩnh vực khảo sát.



Hình 3.4: Không gian các keyphrase

Như vậy, ta gọi **C** = {c ∈ P(K) | c là lớp keyphrase mô tả các lĩnh vực hay chủ đề con thuộc về lĩnh vực đang xét}. Đối với lĩnh vực từ đồng nghĩa, viết tắt ta có

$C = \{C_i \in \mathrm{P}(K)\}, i = \overline{1,...,5}\}$, có 5 lớp tương ứng với 5 nhóm từ. $K = \bigcup_{i=1}^{5} C_i$.

Ví dụ: Lớp NhomTu_DongNghia chứa các keyphrase liên quan cấu trúc dữ liệu như sau: NhomTu_DongNghia = phí, khoản điều chỉnh, phần, tiền trợ cấp, tiền thưởng, tài sản thế chấp, tài sản kí quỹ, số dư, kim loại quí, phổ biến, giông tố, điên, thận trọng, thiện chí,…}, trong đó bao gồm các lớp con khác như: DanhTu_DongNghia, TrangTu_DongNghia….

Đa phần ontology có thế mạnh ở khả năng diễn đạt mô tả quan hệ giữa các lớp và thuộc tính. Có 2 đối tượng chính trong ontology là lớp và keyphrase, lớp mang nội dung tính chất trừu tượng, còn keyphrase là thực thể mô tả chi tiết cho các lớp, giữa 2 đối tượng có các mối quan hệ lẫn nhau như mối quan hệ giữa các lớp, giữa các keyphrase hoặc giữa lớp và keyphrase.

- **Một tập hợp R_{KK} các quan hệ giữa các keyphrase**

Các keyphrase được quản lý trong cùng tập K không tồn tại một cách biệt lập, tách rời nhau mà luôn có những mối quan hệ nhất định. Phân loại quan hệ ngữ nghĩa giữa các keyphrase là rất phong phú và phức tạp, phụ thuộc vào những nội dung đặc trưng ngữ nghĩa cũng như những yêu cầu, mục tiêu của miền tri thức tiếp cận.

Ta có tập $\mathbf{K} \neq \varnothing$, một quan hệ hai ngôi trên **K** là một tập con của $K \times K$, nghĩa là một tập hợp các cặp keyphrase thuộc K và R_{KK} = {r | $r \subseteq K \times K$ }. Tùy thuộc vào miền tri thức, ta có nhiều quan hệ về ngữ nghĩa khác nhau trên keyphrase. Nhìn chung, các quan hệ này có thể được chia thành ba nhóm chính: nhóm quan hệ tương đương, nhóm quan hệ phân cấp, nhóm quan hệ không phân cấp. Trong [14] công trình ontology đã xây dựng được $R_{KK} = \{r_i\}_{i=1}^{25}$ tương ứng với 25 quan hệ chính được nghiên cứu trong công trình. Cho hai phần tử x và y thuộc **K**, ta nói x có quan hệ r_i với y khi và chỉ khi (x,y) $\in r_i$ và viết là x r_i y, ngược lại y có quan hệ r_i^{-1} so với x.

Các keyphrase được gọi là tương đồng nhau khi xét về mặt ngữ nghĩa thì chúng co nội dung giống nhau, khi đó chúng có thể thay thế cho nhau trong một số trường hợp nào đó.

Keyphrase r_1 đồng nghĩa với từ viết tắt r_2 hoặc keyphrase r_1 đồng nghĩa với

keyphrase r_2 nếu trong trường hợp nào đó mà r_1 có thể thay thế cho r_2 thì ta gọi đó là quan hệ tương đương của hai keyphrase

Ví dụ về các từ tương đương nhau

Bảng 3.1: Bảng ví dụ mối quan hệ tương đương

Equivalent keyphrase	**Selected keyphrase**	**Relationship**
UBND	Ủy ban nhân dân	"is a acronym of"
TP	Thành phố	"is a acronym of"
Giấy ủy quyền	Giấy chuyển quyền	"is a synonym of"
Hiếm xảy ra	Không thường xuyên	"is a synonym of"
Năng động	Hoạt bát	"is a synonym of"
Tôn kính	Kính trọng	"is a synonym of"

Những keyphrase đồng nghĩa với nhau được gôm lại thành một nhóm gọi là nhóm keyphrase đồng nghĩa. Trong các nhóm này chúng ta cần xác định được một keyphrase có tính chất nổi bật, thường được sử dụng, có thể dễ dàng suy luận ra các keyphrase khác. Việc xác định được keyphare như vậy cũng tương đối khó vì các phrase tùy vào hoàn cảnh mà có thể thay đổi keyphrase chính. Một số tiêu chí có thể dựa vào như số lần xuất hiện của keyphrase đó, khã năng ghép được với các keyphrase khác được cụm từ có nghĩa…

Như vậy, ngoài các quan hệ đồng nghĩa, viết tắt sử dụng trong đề tài thì còn có quan của các keyphrase có thể được liên kết với nhau thông qua 23 quan hệ khác từ r_3 đến r_{25} (được mô tả trong bảng sau):

Bảng 3.2: Quan hệ giữa các keyphrase trong CK_ONTO

	Quan hệ ngữ nghĩa	**Relation Symbol**	**Mô tả**
r_1	Synonym	syn	A đồng nghĩa với B
r_2	Acronym	acr	A là dạng viết tắt của B
r_3	Near synonym	nsyn	A gần nghĩa với B

r_4	A part of	partOf	A là một phần/công đoạn của B
r_5	A kind of	kindOf	A là một (một dạng của) B
r_6	Beneficiary	benef	A hưởng lợi ích từ B
r_7	Same class	Sacl	A cùng lớp với B
r_8	Agent	agent	A là tác nhân của B, quan hệ chủ thể - hành động
r_9	Cause	cause	A là nguyên nhân gây ra B
r_{10}	Influence	inf	A ảnh hưởng đến B
r_{11}	Instrument	inst	A được sử dụng như là một phương tiên công cụ cho B
r_{12}	Make	make	A tạo ra B
r_{13}	Possession	poss	A sở hữu B
r_{14}	Person	pers	Liên quan đến con người/tổ chức
r_{15}	Aim	aim	Thực hiện A để mà/với mục đích B
r_{16}	Location	loc	Quan hệ vị trí/ không gian
r_{17}	Temporal	temp	Quan hệ thời gian
r_{18}	Manner	manner	A là cách thức mà B xảy ra
r_{19}	Support	support	A xây dựng trên nền tảng B
r_{20}	Extension	ex	A là mở rộng của B
r_{21}	Property	pro	A là một thuộc tính của B
r_{22}	Relation	re	A có liên quan với B
r_{23}	Circumstance	circ	A là một trường hợp/tình huống của B
r_{24}	Source	source	A có xuất xứ từ B
r_{25}	Application	app	A được ứng dụng trong B

Vấn đề xây dựng các keyphrase có quan hệ với nhau cũng yêu cầu chuyên gia trong lĩnh vực cần xây dựng để có thể phân tích chính xác. Ngoài ra thời gian cũng là một vấn đề quan trọng, thời gian để phân tích các mối quan hệ tương đối lâu dài. Nhưng đổi lại chúng ta có được một mô hình dữ liệu đầy đủ, đồng thời, khi tìm kiếm

một thông tin nào đó, ta có thể nhận được những thông tin về các vấn đề khác liên quan tới nó. Vì vậy, để tìm kiếm được những thông tin chính xác, chúng ta cần biết các loại quan hệ và tìm hiểu các phương pháp để xác định được các quan hệ đó.

3.3 Công cụ hỗ trợ xử lý tài liệu văn bản

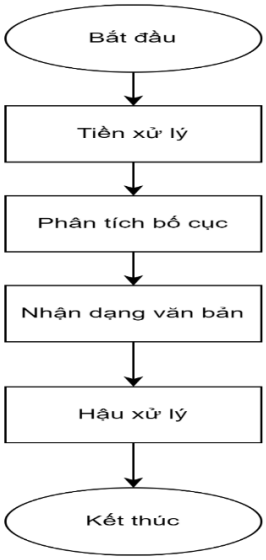
3.3.1 Phương pháp nhận dạng văn bản

3.3.1.1 Giới thiệu

Hiện nay, nhu cầu trích xuất từ hình ảnh ngày càng tăng, bên cạnh sự gia tăng nhu cầu là sự phát triển của công nghệ nhận dạng ký tự quang học (Optical Character Recognition), còn được gọi là nhận dạng ký tự quang học viết tắt là OCR. Đây là một công nghệ chuyển đổi hình ảnh chữ viết tay hoặc đánh máy thành các ký tự được mã hóa bằng máy tính. Giả sử chúng ta cần chỉnh sửa một số tài liệu giấy như: Bài báo trên tạp chí, tờ rơi hoặc tệp PDF hình ảnh. Rõ ràng, chúng ta không thể sử dụng máy quét để chuyển đổi các tài liệu này thành các tệp văn bản có thể chỉnh sửa được. Tất cả những gì máy quét có thể làm là tạo ra một hình ảnh hoặc ảnh chụp nhanh của tài liệu. Để trích xuất và tái sử dụng dữ liệu từ các tài liệu được quét từ hình ảnh của máy ảnh hoặc hình ảnh của các tệp PDF, chúng ta cần một phần mềm OCR. Nó sẽ xuất ra các ký tự có trên hình ảnh, kết hợp chúng thành các từ và sau đó kết hợp các từ thành câu. Nhờ đó, chúng ta có thể truy cập và chỉnh sửa nội dung của tài liệu gốc. Tương tự như vậy, các tài liệu cổ bị hư hỏng theo thời gian sử dụng chữ viết tay hoặc đánh máy lại các tài liệu này sẽ tốn rất nhiều thời gian, và không đảm bảo tính chính xác và an toàn của tài liệu gốc[12].

3.3.1.2 Phương pháp nhận dạng văn bản

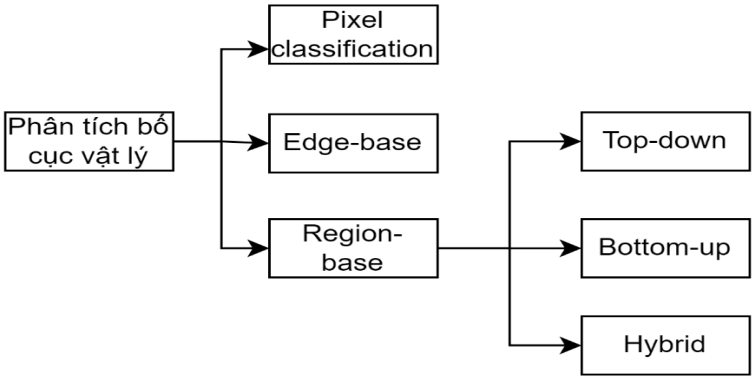
Trong công trình [20] tác giả nghiên cứu một hệ thống nhận dạng văn bản được tổ chức bao gồm bốn thành phần [10]: Tiền xử lý, phân tích bố cục, nhận dạng văn bản và hậu xử lý như được thể hiện trong hình:



Hình 3.5: Tổ chức xử lý nhận dạng văn bản

Trong đó, Tiền xử lý nhằm mục đích cải thiện chất lượng của hình ảnh. Phân tích bố cục cho phép xác định các đối tượng cơ bản trong tài liệu. Các đối tượng được xử lý ở bước này bao gồm chữ ký, từ, ký tự, … Nhận dạng văn bản phân loại các đối tượng được nhận dạng trong bước trước. Cuối cùng, quá trình hậu xử lý kiểm tra kết quả của bước phân loại trên cơ sở thông tin ngữ cảnh. Các mục tiếp theo sẽ lần lượt phân tích cụ thể từng thành phần của hệ thống nhận dạng văn bản.

Phân tích bố cục thực hiện việc phân đoạn hình ảnh tài liệu thành các vùng có nội dung đồng nhất (phân tích bố cục vật lý) và gán ý nghĩa và sắp xếp cho các vùng đó (phân tích bố cục logic). Các kĩ thuật phân tích bố cục vật lý có thể được biểu diễn trong hình [5].



Hình 3.6: Phân loại các thuật toán phân tích bố cục vật lý

Phương pháp từ dưới lên (bottom-up) thực hiện tìm, gán nhãn và phân tích các thành phần nhỏ lân cận nhau, sau đó tiến hành nhóm chúng lại và tìm đường bao quanh chúng tạo thành các vùng lớn hơn. Trong phương pháp từ trên xuống (top-down), một trang được phân đoạn từ các thành phần lớn hơn thành các thành phần con nhỏ hơn. Tesseract sử dụng phương pháp kết hợp của hai phương pháp này (hybrid)[5].

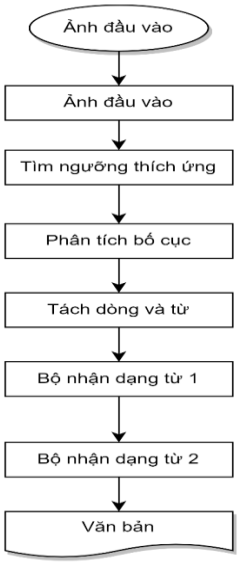
3.3.1.3 Quy trình xử lý Tesseract OCR của tài liệu của hệ thống Tic-Office

Tesseract OCR hoạt động theo từng bước theo sơ đồ khối được hiển thị trong Hình 3.7

Bước đầu tiên là Ngưỡng thích ứng, có chức năng chuyển đổi hình ảnh đầu vào thành hình ảnh nhị phân.

Bước tiếp theo là phân tích thành phần liên thông, được sử dụng để trích xuất phác thảo ký tự. Phương pháp này rất hữu ích vì nó thực hiện OCR (nhận dạng văn bản) với văn bản màu trắng và nền đen.

Sau đó, các phác thảo được chuyển đổi thành blob. Các blob này được tổ chức thành các dòng văn bản, các dòng và vùng được phân tích cho một số vùng cố định hoặc kích thước văn bản tương đương sử dụng phương pháp dựa trên phát hiện tabstop [4][8][9].



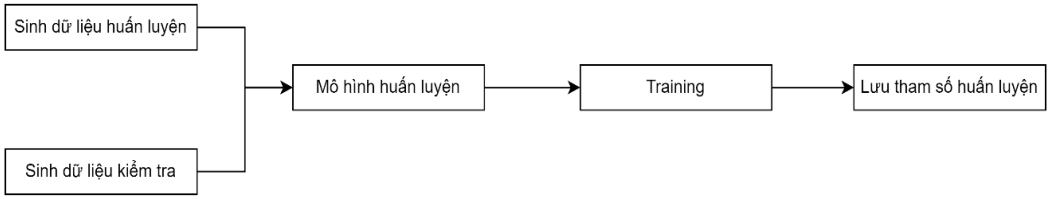
Hình 3.7: Kiến trúc của Tesseract OCR

Quá trình nhận dạng ký tự - là sử dụng các đặc trưng kết hợp với các phương pháp nhận dạng theo ngữ cảnh như: N-gram và nhận dạng từ - là phương pháp sử dụng xác xuất dự đoán các ký tự để đưa ra các từ hoàn chỉnh.

Sau quá trình tách thành các phân đoạn kí tự, Tesseract tìm kiếm các tương đồng giữa các phân đoạn này với danh sách các từ, các số để đưa ra các từ. Tiếp theo, các từ được tách ra sẽ được đưa vào quá trình nhận dạng từ, toàn bộ từ sẽ được phân đoạn và được tách ra thành các kí tự đơn để nhận dạng các kí tự đơn qua hai bộ phân loại là: phân loại tĩnh (static character classifier) và phân loại thích ứng (adaptive character classifier) dựa trên các đặc trưng.

Sau khi kết thúc quá trình nhận tách và nhận dạng các kí tự đơn, một danh sách các kí tự lựa chọn được đưa ra, vấn đề tiếp theo là giải quyết sự mơ hồ (ambiguity) này để chọn ra lựa chọn tốt nhất cho từ [20].

Quá trình đào tạo một mô hình nhận dạng văn bản được thực hiện mô hình hóa thành các bước như Hình 3.8



Hình 3.8: Sơ đồ huấn luyện dữ liệu nhận dạng

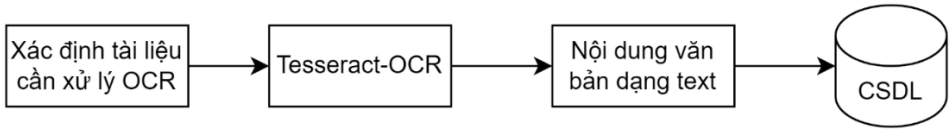
Quá trình sinh dữ liệu đào tạo và sinh dữ liệu kiểm tra nhận đầu vào là văn bản dạng ký tự số và đầu ra là ảnh của văn bản đó, sau đó tất cả các ảnh và văn bản nhãn này được lưu trong tệp nén có đuôi .lstmf;

Tùy thuộc vào các phương thức đào tạo mà mô hình đào tạo được sửa, thay đổi, hoặc xây dựng lại từ đầu;

Trong quá trình đào tạo, phải kiểm soát tỉ lệ lỗi, tỉ lệ bỏ sót, điều chỉnh các tham số đào tạo. Bên cạnh đó, các tệp có đuôi checkpoint được sinh ra để lưu lại trọng số của mô hình tại các bước nhất định;

Cuối cùng, lưu trữ tham số bằng cách nén tệp có đuôi checkpoint tại bước cuối hoặc một bước nào đó, với các tệp liên quan thành tệp có đuôi .traineddata [20].

Quy trình xử lý trích xuất nội dung tài liệu



Hình 3.9: mô tả quy trình xử lý tài liệu văn bản

Các bước thực hiện:

- Bước 1: Xác định tập tài liệu cần rút trích nội dung
- Bước 2: Sử dụng công cụ Tesserract-OCR để xử lý hình ảnh văn bản
- Bước 3: Lưu nội dung đã được rút trích vào CSDL

3.3.2 Phương pháp rút trích nội dung thực thể

3.3.2.1 Định nghĩa:

Thực thể là các đối tượng của thế giới thực bao gồm cả đối tượng có thể nhìn thấy hoặc không nhìn thấy được.

Thực thể trong văn bản thì được thể hiện trong các dạng : Tên riêng, Danh từ hoặc cụm danh từ, Đại từ.

Các loại thực thể được nhận dạng trong nội dung văn bản

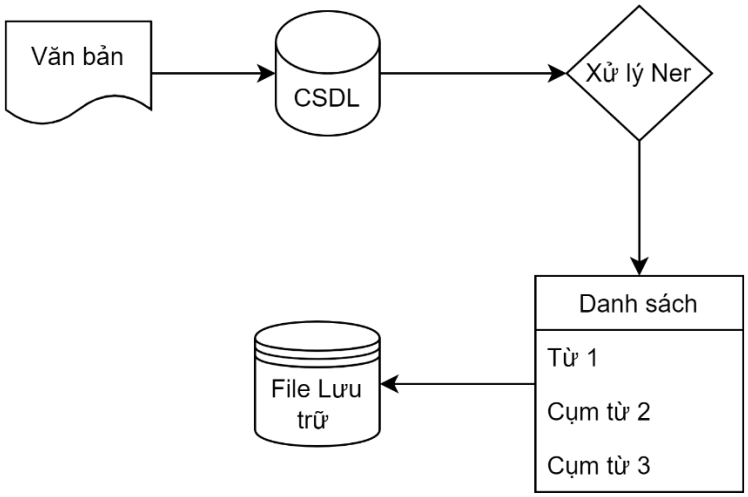
- **Người**: Thực thể chỉ đối tượng là con người
- **Tổ chức**: Thực thể chỉ một tổ chức, một nhóm người được thành lập theo một mô hình quản lý nào đó
- **Cơ sở vật chất**: Thực thể do con người làm ra thường là các đối tượng xây dựng và kiến trúc, như sân vận động, bảo tàng, nhà ga v.v…
- **Vị trí**: Thực thể liên quan đến vị trí địa lý như địa giới hành chính, kênh rạch, địa danh…
- **Quốc tịch**: Thực thể mô tả con người thuộc một quốc gia nào đó.
- **Tôn giáo**: Thực thể chỉ các tổ chức tôn giáo.

Nguyên tắc chung nhận dạng thực thể: Trong một tài liệu được nhận dạng thì không xuất hiện một thực thể có tên đan xen nhau, không hai tên có phần chung với nhau. Trong trường hợp có sự đan xen nhau giữa các tên thì chỉ có tên dài nhất được chọn.

Ví dụ trong câu: "Phòng Giáo dục huyện Châu thành đang họp giao ban." ta chỉ nhận "Phòng Giáo dục huyện Châu thành" là thực thể chỉ các tổ chức (Organization) và bỏ qua thực thể chỉ địa điểm "Châu thành".

Nhận dạng thực thể có tên (Named Entity Recognition – NER) nhằm rút trích các từ, cụm từ trong văn bản là tên của một đối tượng nào đó, điển hình như tên người, tên tổ chức, tên địa danh, thời gian v.v.

3.3.2.2 Quy trình xử lý rút trích thực thể



Hình 3.10: Mô hình xử lý văn bản thành thực thể

Mô tả các bước thực hiện

- Bước 1: Sử dụng công cụ OCR xử lý văn bản lưu vào CSDL
- Bước 2: Sử dụng Underthesea để phân tách nội dung thành các thực thể
- Bước 3 Lưu nội dung đã phân tách thành các tập tin nội dung với tên tập tin theo cấu trúc

3.3.3 Mô hình Conditional Random Field (CRFs)

CRFs dự đoán các nhãn không chỉ dựa trên các từ mà còn tính đến vùng lân cận. Điều này có ý nghĩa, một chuỗi các thực thể hoặc có nhãn thường tồn tại một mẫu nhất định. CRF được sử dụng rộng rãi để mô hình hóa thông tin theo thứ tự. Một số ví dụ có thể kể đến là gán nhãn theo trình tự, giải mã trình tự gen hoặc thậm chí phát hiện đối tượng và phân đoạn hình ảnh trong thị giác máy tính [21].

Mô hình CRFs cho phép các quan sát trên toàn bộ *X*, nhờ đó chúng ta có thể sử dụng nhiều thuộc tính hơn phương pháp Hidden Markov Model. Một cách hình thức chúng ta có thể xác định được quan hệ giữa một dãy các nhãn ***y*** và một câu đầu vào ***x*** qua công thức sau [13].

$$P(y \mid x) = \frac{1}{Z(x)} \exp\left(\sum_t \sum_k \lambda_k t_k (y_{i-1}, y_i, x) + \sum_t \sum_k \mu_k s_k (y_i, x) \right) \quad (3.1)$$

Ở đây ***x,y*** là chuỗi dữ liệu quan sát và chuỗi trạng thái tương ứng;

$t_k(y_{i-1},y_i,x,i)$: là thuộc tính của toàn bộ chuỗi quan sát và các trạng thái tại vị trí i-1, i trong chuỗi trạng thái;

$s_k(y_i,x,i)$: là thuộc tính của toàn bộ chuỗi quan sát và trạng thái tại vị trí i trong chuỗi trạng thái;

λ_j, μ_k: là các tham số được thiết lập từ dữ liệu huấn luyện.

Khi định nghĩa các thuộc tính , chúng ta xây dựng 1 chuỗi các thuộc tính $b(x,i)$ của chuỗi dữ liệu quan sát để diễn tả vài đặc trưng nào đó của phân phối thực nghiệm của dữ liệu huấn luyện.

Ví dụ cho chuỗi quan sát X là " Mỹ Đình- Hà Nội"

$$b(\text{x,i}) = \begin{cases} 1 \text{ nếu quan sát ở vị trí i và từ là = "Đình"} \\ 0 \text{ nếu khác} \end{cases}$$

Mỗi một hàm mô tả sẽ nhận một giá trị của một trong số các giá trị thực $b(x,\text{i})$ là trạng thái hiện tại(nếu trong trường hợp hàm trạng thái) hoặc là trạng thái trước và trạng thái hiện tại (trong trường hợp là hàm dịch chuyển) nhận giá trị riêng. Do đó toàn bộ hàm mô tả có giá trị thực.

Hàm trạng thái $s_k(y_i,x,\text{i})$ dùng để xác định định danh của trạng thái

Ví dụ một hàm trạng thái như sau:

$$s_i = \begin{cases} 1 \text{ nếu } x_i = \text{“chuỗi các số” và } y_i = \text{”Nhãn của chuỗi số”} \\ 0 \text{ nếu ngược lại} \end{cases}$$

Hàm dịch chuyển giúp thêm vào mối quan hệ giữa một nhãn và các nhãn liền kệ với nó.

$$t_i = \begin{cases} 1 \text{ nếu } x_{i-1} = \text{“Mỹ”},\ x_i = \text{“ Đình” và } y_{i-1} = \text{”N1”},\ y_i = \text{”N2”} \\ 0 \text{ nếu ngược lại.} \end{cases}$$

N1: Nhãn bắt đầu địa chỉ

N2: Nhãn từ tiếp theo địa chỉ

Ở đó Z(x) là thừa số chuẩn hóa. Và được tính theo công thức sau:

$$Z(x) = \sum_y \exp\left(\sum_t \sum_k \lambda_k t_k(y_{i-1}, yi, x) + \sum_i \sum_k \mu_k s_k(y_i, x) \right) \quad (3.2)$$

Trong đó: $\theta(\lambda_1, \lambda_2 \ldots\ldots, \mu_1, \mu_2)$ là các véctơ tham số của mô hình.

Chú ý rằng đối với các công thức (2.1) và (2.2) ta có thể viết một cách đơn giản như sau:

$$s_k(y_i, x, i) = s_k(y_{i-1}, y_i, x, i) \text{ và } F_j(y, x) = \sum_{i=1}^{n} fi(y_{i-1}, y_i, x, i) \quad (3.3)$$

Ở đó $f_j(y_{i-1}, y_i, x, i)$ là hàm trạng thái $s_k(y_{i-1}, y_i, x, i)$ hoặc hàm dich chuyển $t_k(y_{i-1}, y_i, x, i)$. Điều này cho ta tính được xác suất của nhãn y khi biết chuỗi quan sát x:

$$P(y|x, \lambda) = \frac{1}{Z(x)} \exp\left(\sum_j \lambda_j F_j(y, x) \right) \quad (3.4)$$

3.4 Xây dựng mô hình VSM trong tra cứu tài liệu có sử dụng ngữ nghĩa cho câu truy vấn

3.4.1 Số hóa văn bản theo mô hình không gian vector

Cách tổ chức dữ liệu

Để biểu diễn lưu trữ tài liệu thành không gian vector thì mỗi tài liệu được xây dựng thành một tập các từ được gán chỉ mục là trọng số của từ đó trong tài liệu, tập từ chỉ mục này dùng để xác định trong không gian vector với mỗi từ là một chiều trong không gian đó.

Ví dụ Giả sử tập D = $\{d_1,d_2,...,d_n\}$ có n văn bản và tập C = $\{c_1,c_2,...,c_m\}$ có m từ chỉ mục biểu diễn cho tập văn bản. Vậy không gian vector biểu diễn tập chỉ mục C có m tập chỉ mục và tập văn bản D có n tập văn bản là một vector $m \times n$ chiều

Với m là số tập chỉ mục thể hiện trên m dòng

n là tập văn bản thể hiện tên n cột

$$D = \begin{pmatrix} d_{11} & d_{21} & \bullet & \bullet & \bullet & d_{n1} \\ d_{12} & d22 & \bullet & \bullet & \bullet & d_{n2} \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ d_{m1} & d_{m2} & \bullet & \bullet & \bullet & d_{mn} \end{pmatrix}$$

Hàm tính trọng số của từ chỉ mục

Hàm tính trọng số của từ chỉ mục trong tập văn bản:

$$\mathbf{w_{ij} = t_{ij} \times T_i \times n_j}$$

Trong đó:

- t_{ij}: tổng số lần xuất hiện của từ chỉ mục trong một văn bản

-T_i : tổng số lần xuất hiện của từ chỉ mục trong toàn bộ văn bản

- n_j: là hệ số điều chỉnh chiều dài của các văn bản trong tập văn bản..

Bảng 3.3: Bảng các hàm tính trọng số cục bộ

STT	Hàm	Tên hàm	Viết tắt
1	1 if $f_{ij} > 0$ 0 if $f_{ij} = 0$	*Binary*	BNRY

2	f_{ij}	*Within_document frequency*	FREQ
3	$1+\log f_{ij}$ if $f_{ij} > 0$ 0 if $f_{ij} = 0$	*Logarithm*	LOGA
4	$(1+\log f_{ij})/(1+\log a_j)$ if $f_{ij} > 0$ 0 if $f_{ij} = 0$	*Normalized log*	LOGN
5	$0.5 + 0.5(f_{ij}/x_j)$ if $f_{ij} > 0$ 0 if $f_{ij} = 0$	*Augumented normalized term frequency*	ATF1

Để tính trọng số cục bộ t_{ij} của chỉ mục trong một văn bản thì trong hình 4.1 có các hàm cơ bản dùng để tính trọng số cục bộ của từ chỉ mục trong văn bản

Hàm nhị phân Binary (BNRY) là hàm đơn giản và dễ sử dụng nhất trong tất cả các hàm đang xét. Hàm BNRY chỉ xét theo trạng thái 0,1 nếu có tồn tại từ chỉ mục đó trong văn bản thì được gán bằng 1 ngược lại gán =0

$$t_{ij} = \begin{cases} 1, & f_{ij} > 0 \\ 0, & f_{ij} = 0 \end{cases} \quad \text{(BNRY)}$$

Trong đo f_{ij} là số lần tìm thấy chỉ mục i trong văn bản j. Nhược điểm của hàm là không phân biệt số từ xuất hiện bao nhiêu lần, chỉ tính có xuất hiện hay không.

Hàm Within_document frequency (FREQ): là hàm tính tổng số lần xuất hiện của từ chỉ mục trong văn bản

$$t_{ij} = f_{ij} \quad \text{(FREQ)}$$

Trong đo f_{ij} là số lần tìm thấy chỉ mục i trong văn bản j. Nhược điểm của hàm FREQ có trọng số quá lớn khi một từ chỉ mục xuất hiện quá nhiều lần

Hàm *Logarithm* được sử dụng để cập nhật số lần xuất hiện của một từ chỉ mục khi từ đó xuất hiện nhiều lần trong văn bản

$$t_{ij} = \begin{cases} 1+\log f_{ij} & if \quad f_{ij} > 0 \\ 0 & if \quad f_{ij} = 0 \end{cases} \quad \text{(LOGA)}$$

$$\text{Và } t_{ij} = \begin{cases} \dfrac{1+\log f_{ij}}{1+\log a_j} & if \quad f_{ij} > 0 \\ 0 & if \quad f_{ij} = 0 \end{cases} \qquad \text{(LOGN)}$$

Trong đó a_j là số lần xuất hiện trung bình của từ chỉ mục trong một văn bản. Hàm LOGN sẽ có trọng số nhỏ hơn hàm LOGA do được đã được chuẩn hóa thêm trọng số trung bình a_j trong văn bản.

Ngoài ra còn có hàm tính trọng số cục bộ là sự kết hợp BNRY và FREQ thành hàm ATF1

$$t_{ij} = \begin{cases} 0.5 + 0.5\left(\dfrac{f_{ij}}{x_j}\right) & if \quad f_{ij} > 0 \\ 0 & if \quad f_{ij} = 0 \end{cases} \qquad \text{(ATF1)}$$

trong đó x_j là tần suất xuất hiện nhiều nhất của các từ chỉ mục trong văn bản j.
khi sử dụng ATF1 thì t_{ij} có giá trị từ 0.5 đên 1.0 cho các từ chỉ mục xuất hiện trong văn bản.

Trọng số toàn cục là trọng số để phân biệt các từ chỉ mục trong toàn bộ tập văn bản với nhau. Các hàm dùng để tính trọng số toàn cục dựa trên ý nghĩa: từ chỉ mục số lần tần suất xuất hiện ít trong toàn bộ văn bản thì có giá trị phân biệt cao hơn. Một hàm tính trọng số toàn cục phổ biến là IDF (*inverted document frequency)* [6].

Bảng 3.4: Bảng các hàm trọng số toàn cục

Hàm	Tên hàm	Viết tắt
$\log\left(\dfrac{N}{n_i}\right)$	*Inverse document frequency*	*IDFB*
$\log\left(\dfrac{N-n_i}{n_i}\right)$	*Probabilistics inverse*	*IDFP*
$1+\sum_{j=1}^{N}\dfrac{\frac{f_{ij}}{F_i}\log\frac{f_{ij}}{F_i}}{\log N}$	*Entropy*	*ENPY*

$\frac{F_i}{n_i}$	*Global frequency IDF*	*IGFF*
1	*No global weight*	*NONE*

Ý nghĩa của các tham số trong các hàm:

- *N* là số văn bản trong tập toàn bộ văn bản
- n_i là số văn bản mà từ chỉ mục *i* xuất hiện
- *Fi* là tần suất xuất hiện của từ chỉ mục *i* trong toàn bộ văn bản đang xét

Công thức chuẩn cosines COSN để tính hệ số chuẩn hóa trong mô hình không gian vector:

$$N_j = \frac{1}{\sqrt{\sum_{i=0}^{m}\left(G_i T_{ij}\right)^2}} \qquad \text{(COSN)}$$

Trong đó:

G_i là trọng số toàn cục của chỉ mục trong toàn bộ văn bản

T_{ij} là trọng số cục bộ của chỉ mục i trong văn bản j

Với hàm COSN, văn bản có xuất hiện từ chỉ mục nhiều thì sẽ có hệ số thấp hơn văn bản có ít từ chỉ mục hơn bởi vì mỗi văn bản có chiều dài khác nhau nên hệ số COSN dùng để làm cân bằng các từ chỉ mục trong tập văn bản.

Trong hàm tính trọng số cục bộ, trọng số toàn cục, hệ số chuẩn hoá thì mỗi hàm đều có ưu điểm và nhược điểm riêng nên việc sử dụng riêng biệt hoặc kết hợp 3 hàm lại với nhau tùy thuộc vào từng hệ thống mà sử dụng các hàm cho hợp lý.

3.4.2 Ma trận biểu diễn tập văn bản

Để biểu diễn tập văn bản D có n văn bản và có m từ chỉ mục được vector hóa thành mô hình vector A, Vector A được gọi là vector của chỉ mục văn bản. Trong đó số tập văn bản n được biểu diễn thành n cột, còn số chỉ mục m được biểu diễn thành m dòng, do số chỉ trong toàn bộ văn bản lúc nào cũng lớn hơn nhiều so với tập văn bản đang xét.

Ví dụ 4.1: Giả sử ta tập văn văn với *n = 4*, trong đó mỗi văn bản chỉ có một câu là tiêu đề của một cuốn sách:

D1: How to **Bake** **Bread** without **Recipes**

D2: The Classic Art of Viennese **Pastry**

D3: Numerical **Recipes**: The Art of Scientific Computing

D4: **Pastry**: A Book of Best French **Recipes**

Giả sử có *m = 5* từ chỉ mục cho các văn bản trên – các từ gạch chân

T1: bak(e, ing)

T2: recipes

T3: bread

T4: cake

T5: pastr(y, ies)

Với 4 văn bản và 5 từ chỉ mục ta biểu diễn ma trân term document A_{5x4} như sau:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Truy vấn văn bản

Trong mô hình không gian vector, truy vấn tập dữ liệu văn bản để tìm tài liệu liên quan đến truy vấn dựa trên các kỹ thuật tính toán trên mô hình không gian vector. Truy vấn được coi là một tập hợp các từ chỉ mục và được thể hiện dưới dạng các tài liệu trong văn bản. Bởi vì câu truy vấn rất ngắn, có nhiều từ chỉ mục của văn bản không xuất hiện trong truy vấn. Điều đó có nghĩa là hầu hết các thành phần của vector truy vấn là bằng không. Thủ tục truy vấn chính là tìm các tài liệu trong văn bản liên quan đến truy vấn, còn được gọi là các tài liệu có độ đo tương tự "cao" với truy vấn. Theo biểu diễn hình học, các văn bản được chọn là các văn bản gần với truy vấn bằng một biện pháp đo lường nhất định.

Cosines là hàm thông dụng dùng để tính góc của hai vector trong không gian nếu góc =0 thì hai vector đó trùng nhau, nếu góc=90 thì hai vector vuông góc, nếu góc =108 thì 2 vector ngược hướng nhau.

Nếu ma trận term – document *A* có các cột được ký hiệu là d_j , $j = 1, ..., n$ thì *n* độ đo *cosines* của vector truy vấn *q* với *n* văn bản trong tập văn bản được tính theo công thức:

$$\cos\theta_j = \frac{d_j^T q}{\|d_j\|_2 \|q\|_2} = \frac{\sum_{i=1}^{m} d_{ij} q_i}{\sqrt{\sum_{i=1}^{m} d_{ij}^2} \sqrt{\sum_{i=1}^{m} q_i^2}} \quad (3.5)$$

Sử dụng tập văn bản trong ví dụ 4.1 ở trên để ví dụ cho thủ tục truy vấn, dựa trên công thức (3.1) tính góc của các vector trong không gian vector 5 chiều (R^5). Giả sử người sử dụng cần những thông tin về nấu ăn và muốn tìm kiếm các cuốn sách về *baking bread*. Với câu truy vấn trên tương ứng với vector truy vấn là:

$q^{(1)} = (1 \quad 0 \quad 1 \quad 0 \quad 0)^T$

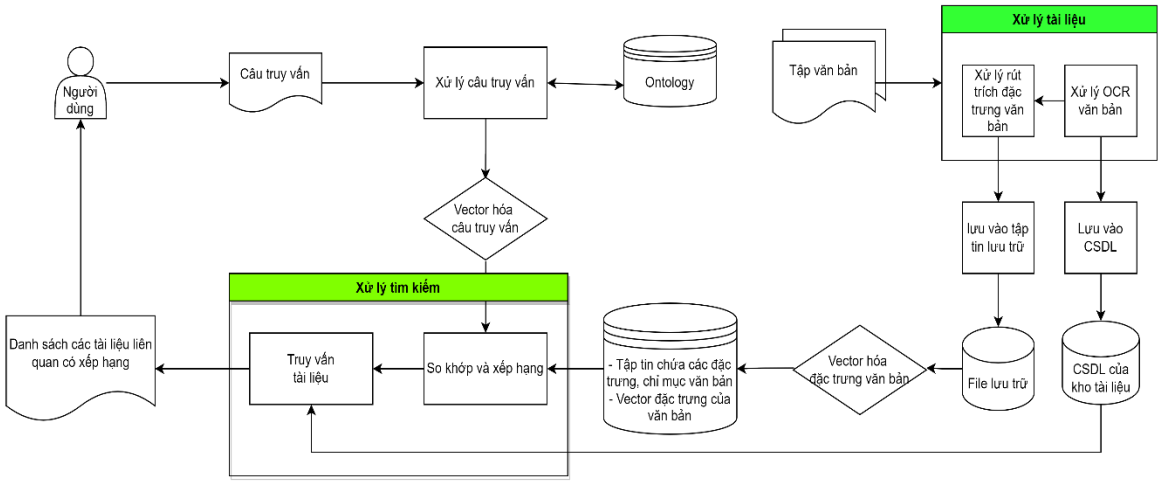
với các phần tử khác không cho hai từ *baking* và *bread*. Việc tìm kiếm các văn bản liên quan được thực hiện bằng cách tính *cosines* của các góc θ_j giữa vector truy vấn $q^{(1)}$ với các vector văn bản d_j bằng công thức (3.1). Một văn bản được xem như liên quan (*relevant*) và được trả về nếu *cosines* của góc được tạo bởi vector truy vấn và vector văn bản đó lớn hơn một ngưỡng (*threshold*) cho trước. Trong cài đặt thực tế ngưỡng được kiểm nghiệm và quyết định bởi người xây dựng hệ thống. Nhưng đối với ví dụ nhỏ này chỉ sử dụng ngưỡng là 0.5.

Với vector truy vấn $q^{(1)}$, chỉ có giá trị *cosines* của các góc khác *zero*: $\cos\theta_1 = 0.8165$. Vậy các văn bản liên quan đến *baking* và *bread* chỉ có *D1* được trả về, các văn bản *D2*, *D3* và *D4* không liên quan và được bỏ qua.

Nếu người sử dụng chỉ muốn tìm các cuốn sách về *baking*, thì kết quả sẽ khác, trong trường hợp này vector truy vấn là:

$q^{(2)} = (1 \quad 0 \quad 0 \quad 0 \quad 0)^T$, và *cosines* của các góc giữa vector truy vấn và 5 vector văn bản theo thứ tự là: **0.5774**, **0**, **0**, **0**, **0**. Vì vậy chỉ văn bản *D1*, là cuốn sách về *baking bread* thoả ngưỡng cho trước 0.5 và được trả về. các văn bản khác thì không có sự liên quan đến baking nên không được trả về.

3.4.3 Kiến trúc mô hình tìm kiếm tài liệu VSM



Hình 3.11: Quy trình xử lý câu truy vấn của hệ thống VSM

Mô tả các bước thực hiện

- Bước 1: người dùng nhập vào nội dung câu truy vấn
- Bước 2: Xử lý câu truy vấn dựa vào mô hình dữ liệu ontology
- Bước 3: Xử lý rút trích đặc trưng, chỉ mục của của tập văn bản
- Bước 4: Tạo tập tin đặc trưng và chỉ mục của văn bản
- Bước 5: Tạo tập tin ma trận đặc trưng của văn bản
- Bước 6: Lưu các tập tin đặc trưng, chỉ mục, ma trận đặc trưng vào kho chờ yêu cầu xử lý
- Bước 7: Xử lý ma trận nội dung truy vấn và ma trận đặc trưng văn bản
- Bước 8: Trả về kết quả các tài liệu có xếp hạng cho người dùng

CHƯƠNG 4: CÀI ĐẶT, THỬ NGHIỆM, ĐÁNH GIÁ

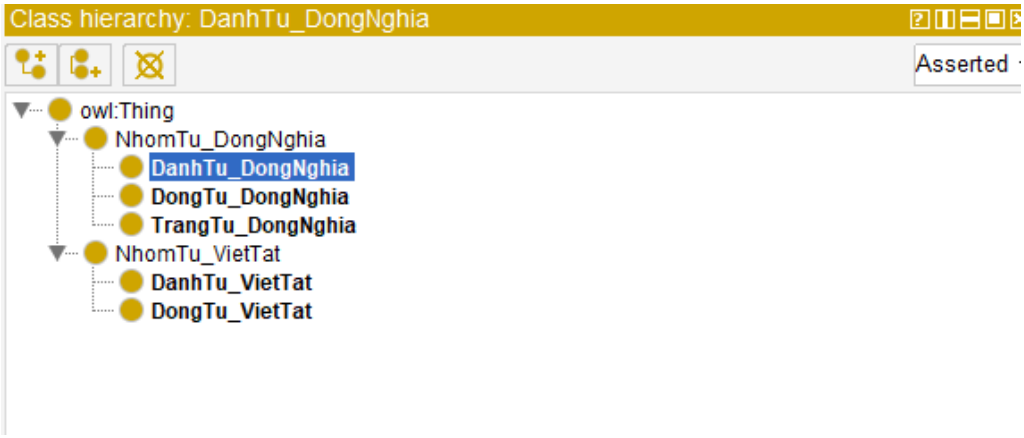
4.1 Cài đặt

4.1.1 Xây dựng mô hình dữ liệu ontology

Mô hình ontology áp dụng cho đề tài sử dụng mô hình CK_ONTO đơn giản gồm 3 thành phần {C, K, R_{KK}} đã được giới thiệu ở Chương 3.

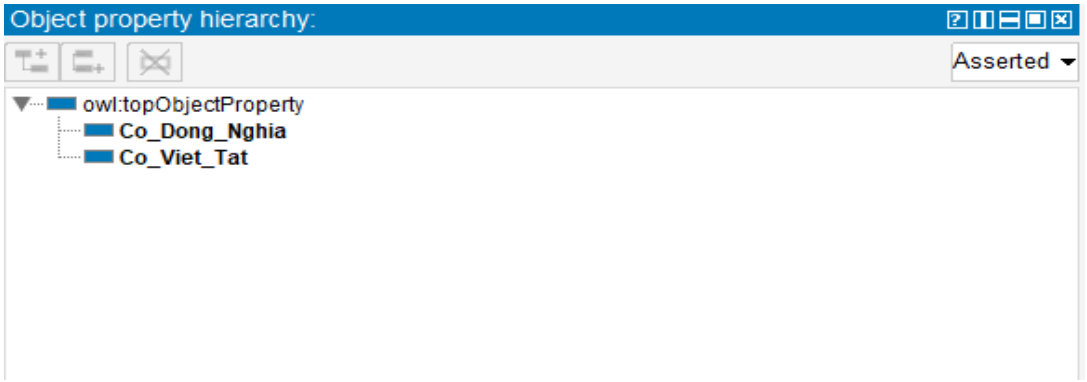
Các bước thực hiện thiết kế mô hình ontology sử dụng công cụ Protégé

Bước 1: Xây dựng các Class (Lớp đối tượng). Đối tượng chính của mô hình gồm 2 thành phần "NhomTu_DongNghia" và "NhomTu_VietTat", trong 2 lớp cha này có các lớp con mô tả chi tiết từng thể loại đồng nghĩa hoặc viết tắt được nêu ở Hình 4.1



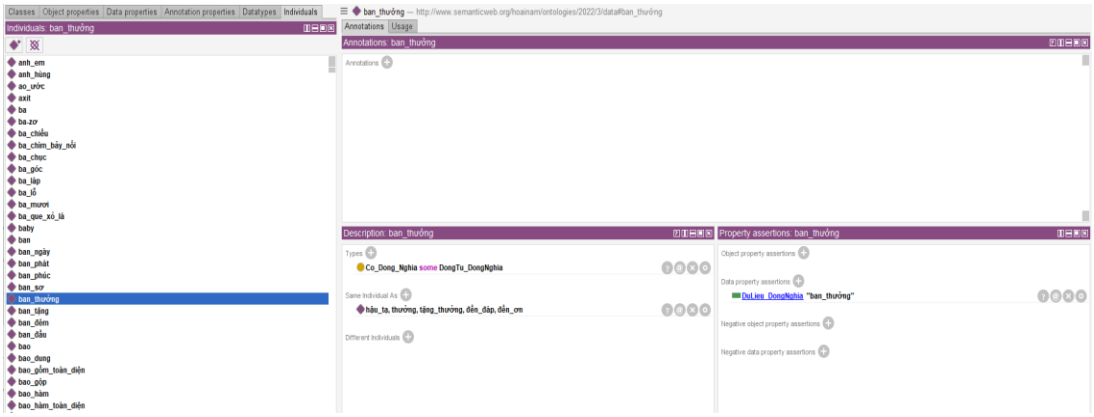
Hình 4.1: mô tả các lớp trong ontology

Bước 2: Xây dựng các thuộc tính cho các lớp, Thuộc tính mô tả cho lớp có 2 thuộc tính của đối tượng là " Co_Dong_Nghia" thuộc lớp "NhomTu_DongNghia" và "Co_Viet_Tat" thuộc lớp "NhomTu_VietTat"



Hình 4.2: mô tả thuộc tính của đối tượng

Bước 3: Xây dựng mối quan hệ giữa các thực thể của các lớp, các thực thể có mối quan hệ đồng nghĩa với nhau thông qua mô tả "Same individual as"



Hình 4.3: mô tả các thực thể có mối quan hệ với nhau

4.1.2 Module trích xuất nội dung của tài liệu sử dụng Tesseract OCR

Để xử lý nội dung tài liệu trong hệ thống, ta sử dụng công cụ mã nguồn mở Tesseract OCR để rút trích nội dung trong tài liệu

Module thực hiện trích xuất nội dung

Input:

- Tài liệu d ⊆ tập tài liệu D cần trích xuất nội dung
- Chỉ mục nhận biết tài liệu d

Thuật toán:

Bước 1: Chuyển các trang tài liệu thành một tập tin hình ảnh

Bước 2: Sử dụng Tesseract OCR trích xuất nội dung của từng tập tin hình ảnh thành các nội dung văn bản text.

Bước 3: Lưu nội dung văn bản đã được rút trích vào CSDL

Output:

Nội dung văn bản của tập tài liệu d

4.1.3 Module rút trích đặc trưng của tài liệu

Input:

- Nội dung của tài liệu d
- Chỉ số phân biệt văn bản (docID)
- Ngày xử lý văn bản

Thuật toán:

Bước 1: Đọc nội dung văn bản từ CSDL

Bước 2: Sử dụng Underthesea để phân tích nội dung văn bản thành thực thể có nghĩa

Bước 3: Lưu nội dung đã phân tích thành các tập tin có tên định dạng "yyyymmdd_ docID.txt"

Output:

Tập hợp các tập tin nội dung text chứa các nội dung đã phân tích

4.1.4 Module API kết nối đến hệ thống Tic-Office

Input:

- Tháng bắt đầu tìm kiếm
- Tháng kết thúc tìm kiếm
- Nội dung tìm kiếm

Thuật toán:

Bước 1: Phân tích xử lý câu truy vấn sử dụng mô hình dữ liệu ontology

Bước 2: Sử dụng module tìm kiếm văn bản thực hiện tìm theo nội dung truy vấn

Bước 3: Trả về chỉ mục văn bản và xếp hạng kết quả thỏa mãn điều kiện tìm kếm

Output:

Tập hợp các chỉ mục của văn bản có xếp hạng tìm kiếm

4.1.5 Cài đặt phân hệ tìm kiếm văn bản

Phân hệ tìm kiếm văn bản gồm 2 module chính:

- Module xây dựng vector đặc trưng của tài liệu
- Module xử lý tìm kiếm thông tin gồm có các nội dung:
  - Tính các độ đo Cosin
  - Xếp hạng kết quả tìm kiếm tài liệu
  - Giao diện thực hiện tìm kiếm và hiển thị kết quả tài liệu có độ tương đồng so với nội dung truy vấn

Module tạo ma trận vector đặc trưng văn bản

Dữ liệu sau khi phân tích thành các thực thể được lưu trữ dưới dạng tập tin text sẽ cung cấp dữ liệu đầu vào cho phân hệ tìm kiếm văn bản: từ tập tin chứa nội dung đã trích xuất nội dung của mỗi văn bản sẽ được vector hoá thành một vector và

toàn bộ văn bản được biểu diễn thành một ma trận ứng với mỗi văn bản là một cột trong ma trân vector, các đặc trưng được thể hiện trong các dòng của vector đó.

Module xử lý tìm kiếm

Các bước thực hiện cơ bản tính độ đo cosin:

Thực hiện lọc ra tất cả các từ đặc trưng trong câu truy vấn bằng cách sử dụng mô hình dữ liệu ontology để hỗ trợ biểu diễn các nội dung liên quan, thực hiện vector hóa câu truy vấn thành vector biểu diễn đặc trưng nội dung truy vấn

Thực hiện tính toán các độ đo Cosine giữa vector câu truy vấn và ma trận vector đặc trưng của tập văn bản, sau đó thực hiện so sánh tất cả các đô đo cosin đã được tính toán với ngưỡng để trả về các văn bản liên quan với câu truy vấn.

- Tính các độ đo Cosin:

Module này thực hiện tìm kiếm các văn bản trong tập văn bản liên quan với câu truy vấn (các văn bản có độ đo Cosine "cao" với câu truy vấn) bằng cách tính độ đo Cosine của từng vector cột (của ma trận từ đặc trưng-văn bản) với vector truy vấn. Một văn bản được xem như liên quan và được trả về nếu độ đo Cosine của vector truy vấn với vector văn bản đó lớn hơn một ngưỡng (threshold). Trong cài đặt của module này, ngưỡng được chọn là 0.03.

- Chức năng xếp hạng kết quả truy tìm

Các văn bản trả về sẽ được hiển thị theo thứ tự độ liên quan với câu truy vấn từ cao đến thấp. Việc xếp hạng kết quả trả về được thực hiện theo thứ tự giảm dần của các độ đo Cosine đã tính toán được.

- Chức năng giao diện thực hiện truy vấn và hiển thị kết quả trả về

Để mang tính ứng dụng thực tiễn cao, giao diện thực hiện truy vấn văn bản được thiết kế theo dạng ứng dụng web.

Cài đặt phân hệ tìm kiếm văn bản VSM

Phân hệ tìm kiếm tài liệu văn bản được cài đặt dựa trên quy trình xử lý tìm kiếm được nêu ở Hình 3.11

- Dữ liệu đầu vào:

Hệ truy tìm văn bản được cài đặt thử nghiệm trên tập 2000 văn bản thuộc nhiều lĩnh vực (kinh tế, chính trị, xã hội…) đã được trích xuất nội dung và lưu dưới dạng tập tin văn bản, ta có các tập tin dữ liệu đầu ra được dùng làm dữ liệu đầu vào cho phân hệ tìm kiếm văn bản có cấu trúc tên tập tin như sau: *"Document\ 20210104_1.txt", "Document\ 20210104_2.txt", "Document\ 20210104_3.txt"...*

Các bước thực hiện:

- Chạy module tạo ma trận đặc trưng văn bản

Tạo các tập tin chứa ma trận vector đặc trưng của tập tài liệu văn bản. Ta có các tập tin: *"202101_docindex.bin", "202101_idfDict.bin", "202101_tfidf.bin", "202101_tokens.bin"...*

- Chạy module xử lý tìm kiếm văn bản

Thực hiện nhập câu truy vấn, kết quả tìm kiếm sẽ trả về các văn bản liên quan với nội dung truy vấn và xếp hạng giảm dần theo độ đo Cosin điều kiện thõa mãn ngưỡng cho trước.

- Giao diện chức năng tra cứu nâng cao được giới thiệu ở Hình 4.4

Ngày Nhận	Ngày gửi	Công văn số	Nơi phát hành	Nơi nhận	Trích yếu	Rank
19/03/2021	19/03/2021	1144-CV/HNDT	Văn thư	Sở Nông nghiệp và Phát triển nông thôn tỉnh	CV 1144 VV góp ý kế hoạch cơ cấu lại ngành nông nghiệp tỉnh Tây Ninh giai đoạn 2021-2025	0.10213853050906062
01/02/2021	01/02/2021	210-QĐ/HNDT	Văn thư	Chánh văn phòng, Trưởng Ban xây dựng Hội, các đồng chí có tên tại điều 1	QĐ 210 công nhận ủy viên Ban Thường vụ HND huyện Tân Biên khóa XI, NK 2018-2023	0.07987793827830175
04/01/2021	04/01/2021	1043-CV/HNDT	Văn thư	HND Thành phố Hồ Chí Minh	CV 1043 Hỗ trợ Hội viên nông dân nghèo nhân dịp Tết Nguyên đán năm 2021	0.07221596470477779
15/03/2021	15/03/2021	1135-Cv/HNDT	Văn thư	Sở Nông nghiệp và Phát triển nông thôn tỉnh	CV 1135 Vv thống nhất hình thức, thẩm quyền ban hành văn bản thực hiện Quyết định số 01/2012/QĐ-TTg ngày 09/01/2012	0.07146228307423051
01/02/2021	01/02/2021	212-QĐ/HNDT	Văn thư	Chánh văn phòng, Trưởng Ban xây dựng Hội, các đồng chí có tên tại điều 1	QĐ 212 công nhận kết quả bầu bổ sung ủy viên Ban Thường vụ HND huyện Bến Cầu khóa XI, NK 2018-2023	0.06771015795279606
18/01/2021	18/01/2021	326/VP	UBND tỉnh		CV 326 Vv triển khai thực hiện Kế hoạch số 45/KH-BCĐTƯ ngày 07/01/2021 của Ban Chỉ đạo tháng hành động về an toàn vệ sinh lao động Trung ương	0.06703538302921673

Hình 4.4: Chức năng tra cứu nâng cao theo ngữ nghĩa

4.2 Kết quả thử nghiệm

Để đánh giá hiệu quả truy tìm tài liệu của hệ thống hiện tại so với hệ thống cũ, luận văn dựa trên kết quả tìm kiếm của toàn bộ hệ thống thử nghiệm thể hiện qua hai độ đo là độ chính xác (precision) và độ bao phủ (recall) để đo sự thỏa mãn của người dùng với các tài liệu mà hệ thống tìm thấy.

Hiện tại, chúng tôi đã xây dựng được một bộ dữ liệu chuẩn trên tập dữ liệu từ đồng nghĩa là một CK_ONTO đơn giản chỉ sử dụng 3 thành phần {C,K,R_{KK}} và một tập nội dung tài liệu đã được rút trích từ hơn 2000 tập tin văn bản trong hệ thống. Tuy nhiên, công tác thực nghiệm cũng gặp nhiều khó khăn vì tốn nhiều chi phí xây dựng và gia công dữ liệu vốn phải có sự can thiệp của con người, đòi hỏi kiến thức của chuyên gia về lĩnh vực và phụ thuộc nhiều vào ngôn ngữ. Hơn nữa, việc đánh giá hiệu quả tìm kiếm của hệ thống cũng đòi hỏi nhiều công sức thủ công cho việc xác định tập tài liệu có liên quan đến từng mẫu truy vấn trên tổng số các tài liệu có trong kho để so sánh với kết quả trả về của các hệ thống.

Với những hạn chế trên, chúng tôi chỉ tiến hành thử nghiệm trong các tài liệu trong khoảng thời gian 3 từ tháng 01/2021 đến tháng 03/2021 bao gồm 672 tập tin tài liệu văn bản. Ứng với tập tài liệu thực hiện khảo sát trên 20 câu truy vấn có chọn lọc và tính toán các độ đo recall, precision tương ứng, với ngưỡng chặn là 0.03. Hệ thống hiện tại tìm được hầu hết các tài liệu có liên quan đến nội dung cần tìm và được sắp xếp theo thứ tự độ liên quan giảm dần chính xác hơn so với thứ tự độ liên quan trong hệ thống cũ. Kết quả thực nghiệm với độ đo precision trung bình của hệ thống hiện tại là 85.9% và độ đo recall trung bình là 80.5% trên tập cơ dữ liệu thử nghiệm so với hệ thống cũ được thể hiện ở Bảng 4.1 và Bảng 4.2

Kết quả thử nghiệm của câu truy vấn trên bộ dữ liệu bao gồm 672 tài liệunhư sau:

Gọi S: số lượng tài liệu mà hệ thống tìm thấy được đánh giá là có liên quan theo người dùng.

T: Tổng số các tài liệu tìm thấy của hệ thống

U: Tổng số tài liệu liên quan theo đánh giá của người dùng có trong hệ thống

Bảng 4.1: Thống kê kết quả tìm kiếm trên chức năng tra cứu mới

STT	Query	S	T	U	P (S/T)	R (S/U)
1	công nghệ thông tin	43	55	59	78%	73%
2	nông sản	42	61	61	69%	69%
3	quy hoạch giao thông	60	78	82	77%	73%
4	dự toán ngân sách	73	80	81	91%	90%
5	hội nhập	44	50	55	88%	80%
6	chi phí	55	69	73	80%	75%
7	không khí ô nhiễm	35	41	51	85%	69%
8	giới thiệu du lịch chuyên nghiệp	47	59	60	80%	78%
9	giải quyết khiếu nại	50	55	59	91%	85%
10	chuyên canh hè thu	12	13	13	92%	92%
11	hàm lượng khoa học công nghệ cao	38	44	46	86%	83%
12	đài phát thanh truyền hình	28	32	36	88%	78%
13	trung tâm thương mại	50	59	61	85%	82%
14	mật độ chăn nuôi	40	45	53	89%	75%
15	tăng trưởng kinh tế	41	48	61	85%	67%
16	sở tài chính	78	88	91	89%	86%
17	ngân sách	53	58	62	91%	85%
18	phát triển nông thôn	61	66	66	92%	92%
19	an toàn giao thông	69	77	77	90%	90%
20	kinh nghiệm sản xuất	60	68	72	88%	83%
21	khai thác khoáng sản	33	38	41	87%	80%
22	tình hình kinh tế xã hội	42	47	49	89%	86%

Bảng 4.2: Thống kê kết quả tìm kiếm trên chức năng tra cứu cũ

STT	Query	S	T	U	P S/T)	R (S/U)
1	công nghệ thông tin	0	55	59	0%	0%
2	nông sản	4	61	61	7%	7%
3	quy hoạch giao thông	0	78	82	0%	0%
4	dự toán ngân sách	2	80	81	3%	2%
5	hội nhập	10	50	55	20%	18%
6	chi phí	2	69	73	3%	3%
7	không khí ô nhiễm	0	41	51	0%	0%
8	giới thiệu du lịch chuyên nghiệp	0	59	60	0%	0%
9	giải quyết khiếu nại	2	55	59	4%	3%
10	chuyên canh hè thu	0	13	13	0%	0%
11	hàm lượng khoa học công nghệ cao	0	44	46	0%	0%
12	đài phát thanh truyền hình	1	32	36	3%	3%
13	trung tâm thương mại	0	59	61	0%	0%
14	mật độ chăn nuôi	2	45	53	4%	4%
15	tăng trưởng kinh tế	0	48	61	0%	0%
16	sở tài chính	21	88	91	24%	23%
17	ngân sách	5	58	62	9%	8%
18	phát triển nông thôn	6	66	66	9%	9%
19	an toàn giao thông	17	77	77	22%	22%
20	kinh nghiệm sản xuất	0	68	72	0%	0%
21	khai thác khoáng sản	0	38	41	0%	0%
22	tình hình kinh tế xã hội	0	47	49	0%	0%

Từ bảng thống kê Bảng 4.1 và Bảng 4.2 ta nhận thấy mô hình tra cứu áp dụng mô hình dữ liệu ontology và VSM trong việc so khớp tài liệu với câu try vấn thể hiện được đầy đủ nội dung hơn so với chức năng tra cứu hiện tại. Có nhiều nội dung tra

cứu có thể tìm kiếm được trên chức năng mới mà không tìm được trên chức năng hiện tại.

4.3 Đánh giá

Với mô hình tìm kiếm chuyên biệt có tính tập trung cao vào nội dung đặc trưng của tài liệu, kết quả trả về có độ chính xác khá cao trên tập dữ liệu thử nghiệm, được đánh giá là thoả mãn tốt nhu cầu khai thác thông tin của người sử dụng. Với việc tích hợp khả năng phân tích ngữ nghĩa, ngoài kết quả trả về, chúng ta có thể tìm thấy những dữ liệu liên quan khác từ đó mở rộng vấn đề ngoài kết quả tìm kiếm. Từ kết quả thực tế cho thấy, việc sử dụng mô hình ontology cùng với các kỹ thuật xử lý liên quan đã giúp cho hệ thống tìm kiếm với độ chính xác và độ phủ trung bình cao hơn so với hệ thống cũ trên cùng một bộ dữ liệu thử nghiệm.

Vẫn có số một trường hợp hệ thống cũ cho ra kết quả tốt hơn, nhưng nhìn chung hệ thống mới đã đem lại kết quả khả quan hơn rất nhiều. Ngoài kết quả thực nghiệm đã được trình bày ở trên. Việc triển khai và thử nghiệm mô hình giải pháp mới đã đem lại thành công rất đáng khích lệ. Hệ thống cho thấy tính khả thi và thực nghiệm của giải pháp kết hợp mô hình ontology, VSM và các kỹ thuật xử lý khác.

Bên cạnh đó, thành phần tập dữ liệu đồng nghĩa trong ontology giúp cho hệ thống có khả năng xác định mối quan hệ ngữ nghĩa giữa các đối tượng của câu truy vấn so với giải pháp của hệ thống cũ.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được của đề tài

Với những mục tiêu đề ra ban đầu, luận văn đã thực hiện hoàn chỉnh nội dung nghiên cứu. Nhóm nghiên cứu đã đề xuất được một mô hình ontology biểu diễn nội dung câu truy vấn cùng với các kỹ thuật xử lý ngữ nghĩa để cải thiện giải pháp thiết kế hệ hỗ trợ tìm kiếm theo ngữ nghĩa cho hệ thống Tic-Office. Mô hình ontology đã giúp hệ thống có khả năng mở rộng việc xử lý câu truy vấn và xác định độ tương đồng ngữ nghĩa giữa các keyphrase một cách tự động. Từ đó, cải thiện độ chính xác và độ bao phủ của tập kết quả trả về trong quá trình tìm kiếm tài liệu theo ngữ nghĩa.

Luận văn đã phân tích và đánh giá các phương pháp tiếp cận trong việc tổ chức lưu trữ và xử lý ngữ nghĩa của tài liệu đã biết, đặc biệt là giải pháp "Kết hợp sử dụng mô hình VSM với mô hình ontology biểu diễn nội dung truy vấn". Từ đó, về mặt lý thuyết, luận văn đã đóng góp trong việc phát triển mô hình biểu diễn tri thức của lĩnh vực và các kỹ thuật xử lý liên quan tới ngữ nghĩa.

Mở rộng kỹ thuật xử lý câu truy vấn, tận dụng các thông tin mô tả cấu trúc của một khái niệm được định nghĩa trong ontology, tiến hành phân tích và nhận dạng mẫu câu truy vấn để đưa câu truy vấn về dạng cụ thể hơn. Nếu không có kết quả nào được trả về thì ta tiến hành xử lý bằng cách rút trích tự động các đặc trưng diễn đạt nội dung chính muốn tìm kiếm, biểu diễn câu truy vấn thành mô hình vector để thuận tiện trong quá trình so khớp nội dung.

Bên cạnh việc nêu lên các ưu thế và lợi ích của việc nghiên cứu, phát triển mô hình cùng với các đặc trưng dựa trên ngữ nghĩa, chúng tôi đã cài đặt và xây dựng được một ứng dụng thử nghiệm từ những cải tiến này. Đây là một hệ thống quản lý văn bản Tic-Office của Hội nông dân tỉnh Tây Ninh, với yêu cầu sử dụng bao gồm các tác vụ chính là tổ chức lưu trữ, quản lý và tìm kiếm, đặc biệt là chức năng tìm kiếm theo ngữ nghĩa liên quan đến nội dung của tài liệu. Từ đó là cơ sở để đánh giá tính hiệu quả của việc cải tiến so với chức năng tra cứu đã có của hệ thống. Kết quả đạt được là hệ thống đã cho ra kết quả tìm kiếm có độ chính xác trung bình là 85.9%

và độ phủ trung bình là 80.5% cao hơn với hệ thống cũ trên tập dữ liệu là 672 tài liệu được phân bố trong thời gian thử nghiệm là 3 tháng.

Luận văn đã đạt được mục tiêu là xây dựng chức năng tra cứu theo ngữ nghĩa của hệ thông văn bản Tic-Office để hỗ trợ người dùng tìm kiếm các văn bản liên quan với nội dung truy vấn và cũng đã xây dựng được mô hình dữ liệu ontology từ đồng nghĩa, từ viết tắt hỗ trợ cho câu truy vấn của người dùng. Giúp tìm được các nội dung liên quan về ngữ nghĩa trong tài liệu

Luận văn đã áp dụng các kỹ thuật rút trích dữ liệu đặc trưng từ hình ảnh góp phần tăng thêm nội dung của văn bản được quản lý, ngoài ra đề tài còn áp dụng mô hình so khớp văn bản sử dụng Vector space model kết hợp mô hình dữ liệu hỗ trợ cho câu truy vấn làm tăng kết quả chính xác của câu truy vấn.

5.2. Những hạn chế của đề tài

Các kỹ thuật đề xuất trong luận văn như OCR, rút trích đặc trưng vẫn còn nhiều hạn chế như văn bản quét không chính xác bị lệch, mờ, thiếu chữ dẫn đến tình trạng rút trích còn nhiều khó khăn trong việc xử lý. Công cụ rút trích đặc trưng văn bản vẫn chưa phân loại được một số đối tượng danh từ riêng, từ viết tắt vào nhóm đối tượng dẫn đến bị thiếu sót trong xử lý dữ liệu.

Quá trình xử lý câu truy vấn đã được mở rộng, tuy nhiên vẫn còn khá đơn giản, chưa tận dụng đầy đủ thông tin trong cấu trúc của lớp. Nếu thông tin của lớp được khai thác đầy đủ thì hệ thống sẽ có khả năng xử lý các cấu truy vấn phức tạp hơn như ở dạng câu hỏi hay câu diễn đạt mệnh đề. Mặc dù còn đơn giản nhưng đây là cơ sở cho việc mở rộng xử lý câu truy vấn so với việc chỉ tìm theo từ khóa trong hệ thống cũ. Đề tài mới chỉ dừng ở mức xây dựng mô hình dữ liệu hỗ trợ trong câu truy vấn, chưa hỗ trợ biểu diễn cho nội dung văn bản vì vậy thông tin so khớp giữa văn bản và câu truy vấn chưa đầy đủ so với nội dung thực tế của văn bản.

5.3. Hướng phát triển

Tiếp tục phát triển, hoàn thiện các mô hình biểu diễn tri thức, biểu diễn ngữ nghĩa cho tài liệu văn bản, mô hình xử lý ngôn ngữ tự nhiên để rút trích nội dung từ văn bản được chính xác. Nghiên cứu các thuật toán nhằm hỗ trợ tìm kiếm nhanh

chóng hơn, dễ dàng hơn, cho kết quả chính xác hơn với nhu cầu tìm kiếm của người dùng. Nghiên cứu các công cụ hỗ trợ tự động, tự động hóa càng cao càng tốt trong từng khâu xử lý chẳng hạn như các mô hình và giải pháp rút trích các đặc trưng từ tài liệu kết hợp với mô hình dữ liệu của tài liệu, các kỹ thuật trong xác suất thống kê, máy học.... Nghiên cứu các giải pháp mới trong lĩnh vực tìm kiếm ngữ nghĩa để tìm ra khả năng tìm kiếm trong nhiều lĩnh vực tri thức khác. Đưa đến một giải giải pháp xây dựng một hệ thống tra cứu xử lý tổng hợp toàn bộ quy trình quản lý văn bản.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Bilal Ahmad Abu-Salih, "*Applying Vector Space Model (VSM) Techniques in information Retrieval for Arabic Language"*

[2] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, "*An Introduction to Information Retrieval*", Cambridge University Press Cambridge, England, 2009.

[3] Carola Eschenbach, Michael Gruninger (FOIS 2008), "*Formal Ontology in Information Systems"*

[4] Faisal Shafait, Ray Smith (2010), *"Table detection in heterogeneous documents"*

[5] Mindy Bokser,(1992*) "Omnidocument Technologies"*

[6] Nordianah Ab Samat, Masrah Azrifah Azmi Murad, Muhamad Taufik Abdullah, Rodziah Atan (2008), "*Term Weighting Schemes Experiment Based on SVD for Malay Text Retrieval"*, Faculty of Computer Science and Information Technology University Putra Malaysia, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.10, October.

[7] Rajendra Prasath and Sudeshna Sarkar, *"Cross-Language Information Retrieval with Incorrect Query Translations"*

[8] Ray Smith, Daria Antonova, Dar-Shyang Lee, (2009), *"Adapting the Tesseract Open Source OCR Engine for Multilingual OCR"*

[9] Ray Smith, (2009) ,*"Hybrid Page Layout Analysis via Tab-Stop Detection"*

[10] Simone Marinai, (2008),*"Introduction to Document Analysis and Recognition"*

[11] Phạm Tuấn Dạt, Nguyễn Văn Thủy (2016), *Ứng dụng thư viện lập trình mã nguồn mở xây dựng chương trình nhận dạng văn bản chữ việt, anh từ ảnh số*

[12] Nguyễn Đình Ngọc (2015), "*Xây dựng phần mềm nhận dạng ký tự quang học sử dụng mã nguồn mở Tesseract ocr"*

[13] Nguyễn Thị Loan (2009), "*Tìm hiểu mô hình CRF và ứng dụng trong trích chọn thông tin trong tiếng việt* ", Trường đại học Công nghệ đại học quốc gia Hà Nội.

[14] Huỳnh Thị Thanh Thương (2012), *"Nghiên cứu mô hình tổ chức và kỹ thuật tìm kiếm có ngữ nghĩa trên kho tài nguyên học tập lĩnh vực CNTT"*, Trường đại học Khoa Học Tự Nhiên TP.HCM.

[15] https://blog.duyet.net/2019/08/ir-vector-space-model.html

[16] https://butchiso.com/2013/10/tim-hieu-ve-mo-hinh-khong-gian-vector.html

[17] Chuyển đổi số tại Việt Nam và những thống kê ấn tượng đầu năm 2021 | Visual Story - Báo Lao Động (laodong.vn)

[18] HỮU THỂ HỌC / BẢN THỂ HỌC (Ontology) - Triết học (triethoc.edu.vn)

[19] DCMI: Ontology (dublincore.org)

[20] Deep learning ứng dụng trong nghiệp vụ nhận dạng văn bản - An Toàn Thông Tin (antoanthongtin.vn)

[21] Conditional Random Fields - Trí tuệ nhân tạo (trituenhantao.io)

[22] Lp space - Wikipedia

[23] Precision and recall - Wikipedia

PHỤ LỤC

1. Source code xử lý trích xuất tài liệu

```
def ocr_pdf(path):
    try:
        print('Convert image to string...')
        pytesseract.pytesseract.tesseract_cmd = r"C:\\Program
Files\\Tesseract-OCR\\tesseract.exe"

       # Grayscale, Gaussian blur, Otsu's threshold
        image = cv2.imread(path)
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
        blur = cv2.GaussianBlur(gray, (3,3), 0)
        thresh = cv2.threshold(blur, 0, 255, cv2.THRESH_BINARY_INV +
cv2.THRESH_OTSU)[1]

        # Morph open to remove noise and invert image
        kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3,3))
        opening = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel,
iterations=1)
        invert = 255 - opening

        # Perform text extraction
        data = pytesseract.image_to_string(invert, lang='vie', config='--psm
6')

    except:
        wrire_file_text("Error: "+path )

    return data
```

2. Souce code rút trích nội dung tài liệu

```
from underthesea import sent_tokenize
import regex as re
from underthesea import word_tokenize
import pymssql

server = '127.0.0.1\\home'
user = 'namhh.tnh'
password = 'vnpt@123'
#CONNECT SQL LAY NOI DUNG
index =0
with pymssql.connect(server, user, password, "data_hnd") as conn:
```

```
    with conn.cursor(as_dict=True) as cursor:
        cursor.callproc('sp_sel_write_file_txt')#sp_sel_noidung_sent_token
        for row in cursor:
            index = index+1
            _doc_id = row['Doc_Id']
            _mscv =row['Mscv']
            _noidung = accent_vietnamese(row['NoiDung'])
            _noidungkodau = no_accent_vietnamese(row['NoiDung_NoAccent'])
            _ngay_xl =  row['Ngay_XuLy']

            if check_duplicate_string(_noidung) == 0:
                with open("D:\\Accent\\"+ _ngay_xl+"_"+str(_doc_id) +".txt",
'a', encoding="utf-8", newline='') as f_writer:
                    f_writer.write(_noidung+"\n")


conn.close()
```

3. Source code hàm xử lý chính kiểm tra tương đồng câu truy vấn và nội dung tài liệu

```
def _main_
(self,local_path,fromDate,toDate,query,data_onto,nguong_giatri=0):
        self.local_path = local_path
        self.fromDate = fromDate
        self.toDate = toDate
        thang =fromDate[0:6]
        qt = word_tokenize(query)
        qt =qt + data_onto
        print('--------')
        print(qt)
        self.word=qt #luu lai de tra ve postman
        qt = [self.accent_vietnamese(item) for item in qt if item != '']
        qt=list(set(qt))
        lst={}
        lstIndex = {}
        lst_res ={}
        m_thang = self.compute_months(fromDate[0:6],toDate[0:6])
        for k_thang in m_thang:
            with open(local_path+"\\"+k_thang+"_docindex.bin", 'rb') as fin:
                _docIndex =  pickle.load(fin)
                _listkey= _docIndex.keys()
            with open(local_path+"\\"+k_thang+"_tokens.bin", 'rb') as fin:
                _tokens= pickle.load(fin)
            with open(local_path+"\\"+k_thang+"_idfDict.bin", 'rb') as fin:
```

```
                _idfDict= pickle.load(fin)
            with open(local_path+"\\"+k_thang+"_tfidf.bin", 'rb') as fin:
                _tfidf = pickle.load(fin)
            qtV=dict.fromkeys(_tokens,0)
            for word in qt:
                if word in _tokens:
                    qtV[word]+=1
            for words in qtV:
                if word in _tokens:
                    qtV[words]=qtV[words]*_idfDict[word]
            res={}
            temp=0
            vec1=np.array([list(qtV.values())])
            for x in _listkey:
                vec2=np.array([list(_tfidf[x].values())])
                if cosine_similarity(vec1,vec2)>0:
                    temp=cosine_similarity(vec1,vec2)[0][0]
                    res[x]=temp
            res=sorted(res.items(), key=operator.itemgetter(1),
reverse=True)
            for items in res:
                if items[1]> nguong_giatri: #gia tri min hien thi
                    lst[items[0]]=items[1]
                    postion = list(_docIndex.keys()).index(items[0])
                    _val = list(_docIndex.values())[postion]
                    lstIndex[items[0]]=_val
                    lst_res[_val[9:]]=items[1]

        self.ketqua = lst
        self.Doc_Index = lstIndex
        self.ListID = lst_res
```

4. Source code tạo ma trận vector

```
 def accent_vietnamese(self,s):
        k = s.lower()

        #k = re.sub(r"['!@#$%^&*()_=-\|:';:,<.>/?`~]", ' ', k)
        k = re.sub(r'[0-9]', ' ', k)
        removetable=str.maketrans("", "", "'!@#$%^&*()_=-\|][:';:,<.>/?`~")
        k=[x.translate(removetable) for x in k]
        #s = re.sub("[^A-Za-z]"," ",s)
        k =" ".join("".join(k).split())
        return k
```

```
    def create_tokens(self):
        x = 0
        Doc_Index={}
        docToken ={}
        tokens =[]

        for dirname, _, filenames in os.walk(self.local_path):
            for filename in filenames:
                _ngay = filename[0:8]
                _doc_id =filename[0:filename.index('.')]
                if self.fromDate <= _ngay <=self.toDate :
                    print(_ngay)
                    print(self.fromDate)
                    print(self.toDate)
                    x=x+1
                    if _doc_id in Doc_Index.values():
            postion = list(Doc_Index.values()).index(_doc_id)
                        x = list(Doc_Index.keys())[postion]
                    else:
                        Doc_Index[x]=_doc_id
                    with open(os.path.join(dirname, filename), 'r',
encoding='utf-8') as f:
                        data = f.read().split('\n')
                        if len(docToken)==x-1:
                            docToken[x] = data
                        else:
                            docToken[x] += data
                        tokens += data
        tong_dong = x+1
        tokens = [item for item in tokens if item != '']
        tokens=list(set(tokens
#--------------------Document wise Tokenization----------------------#

        for x in docToken:
            docToken[x] = [item for item in docToken[x] if item !=
'']           docToken[x]=list(set(docToken[x]))
            docToken[x]=sorted(docToken[x])

##--------------------Document wise Tokenization----------------------#
                self.tokens = tokens
        self.Doc_Index = Doc_Index
        self.docToken = docToken
###=======Word Frequency=========================================#

    def word_frequency(self):
```

```
        docV={}
        _ListKeys =self.getListKeys()

        for x in _ListKeys:
            docV[x]=dict.fromkeys(self.tokens,0)

        for x in _ListKeys:
            for word in self.docToken[x]:
                docV[x][word]+=1
        tfDocV={}
        for x in _ListKeys:
            tfDocV[x]={}
            for word,count in docV[x].items():
                tfDocV[x][word]=count

        wordDcount=dict.fromkeys(self.tokens,0)
        for word in self.tokens:
            for x in _ListKeys:
                if word in self.docToken[x]:
                    wordDcount[word]+=1
        idfDict = {}
        for word in self.tokens:
            if wordDcount[word]>0:
                count=wordDcount[word]
                if count> max(_ListKeys):
                    count=max(_ListKeys)
            idfDict[word]=math.log(max(_ListKeys)/count
        tfidf={}
        for x in _ListKeys:
            tfidf[x]={}
            for word in docV[x]:
                tfidf[x][word]=tfDocV[x][word]*idfDict[word]

               self.docV=docV
        self.tfDocV=tfDocV
        self.idfDict=idfDict
        self.tfidf=tfidf
```

5. Source code API kết nối

```
@app.post("/query")
async def query(item: Item ):
    doc_xuly = main_tic.class_tokens()
    query = item.query
    tuthang = item.tuthang
```

```
    denthang = item.denthang
    data_onto=set(onto.get_onto(query))
    doc_xuly._main_file_(path_pkl,tuthang,denthang,query,data_onto,0.03)
    data=[]
    data.append(list(doc_xuly.get_word()))
    data.append(doc_xuly.get_ListKQ())
    return data
```

6. Source code xử lý tìm kiếm giao diện người dùng

```
clsApi cls = new clsApi();
        string session= cls.geSession();
        string json_data =
cls.getData(txtTimKiem.Text,DateTime.Parse(d_tungay.Value.ToString()).ToString("yyyy
MM"), DateTime.Parse(d_denngay.Value.ToString()).ToString("yyyyMM"));
        string[] arr_json = json_data.Split('{');
        arr_json[0] = arr_json[0].Replace("[","").Replace("],","").Replace("\"","");
        ASPxLabel10.Text = arr_json[0];
        arr_json[1] = arr_json[1].Replace("{", "").Replace("}",
"").Replace("\"","");
        string doc_id = "";
        Dictionary<string, string> kq = new Dictionary<string, string>();
        string []arr_doc = arr_json[1].Split(',');
        foreach (string x in arr_doc)
        {
            if (x != "")
            {
                kq[x.Split(':')[0]] = x.Split(':')[1];
                doc_id += "," + x.Split(':')[0];
            }

        }
        doc_id = doc_id.Substring(1);
        DataProvider con = new DataProvider();
        con.Connect();

        string sql = "select distinct sothutu,convert(varchar(10),ngaynhan,103)
ngaynhan,convert(varchar(10),ngaygui,103) ngaygui, vanban_id,
noi_phat_hanh,noi_nhan, " +
            "trich_yeu , AUTO_ID , lastpost,b.Doc_Id ,cast('' as varchar(100))Rank
"+
            " from hoinongdan..Vanban_bk a ,data_hnd..tbl_pos_tag_map b where
a.AUTO_ID=b.mscv and b.doc_id in(" + doc_id+")";


        DataTable dt = con.executeDatatableNonparam(sql);
        con.Disconnect();

        for (int i=0; i<dt.Rows.Count;i++)
        {
            dt.Rows[i]["Rank"] = kq[dt.Rows[i]["Doc_Id"].ToString()];
        }
        DataView dv = dt.DefaultView;
        dv.Sort = "Rank DESC";
```

```
for (int i = 0; i < dt.Rows.Count; i++)
{

    dt.Rows[i]["sothutu"] = (i + 1).ToString();

}
Grid_nangcao.DataSource = dv;
Grid_nangcao.DataBind();
```