

## BỘ GIÁO DỤC VÀ ĐÀO TẠO

**---------------**

LUẬN VĂN THẠC SỸ

# Chuyên ngành: CÔNG NGHỆ THÔNG TIN

CẢI TIẾN TRUY VẤN THÔNG TIN TRONG VĂN BẢN PHÁP LUẬT TIẾNG VIỆT

## GVHD : TS. LÊ THỊ NGỌC THƠ HVTH : VÕ PHẠM THÀNH LUÂN MSHV : 2141860011

**Lớp : 20SCT21**

TP. HCM, tháng 09/2022

### NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

*Tp. HCM, ngày …… tháng …… năm 2022*

### Giáo viên hướng dẫn

**NHẬN XÉT CỦA HỘI ĐỒNG XÉT DUYỆT**

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

............................................................................................................................................

*Tp. HCM, ngày …… tháng …… năm 2022*

### Hội đồng xét duyệt

**MỤC LỤC**

1. [GIỚI THIỆU 7](#_bookmark0)
   1. [Đặt vấn đề 7](#_bookmark1)
   2. [Tính cấp thiết của đề tài 9](#_bookmark2)
2. [MỤC TIÊU, NỘI DUNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU 10](#_bookmark4)
   1. [Mục tiêu của đề tài 10](#_bookmark5)
   2. [Đối tượng nghiên cứu 11](#_bookmark6)
   3. [Nội dung nghiên cứu 11](#_bookmark7)
   4. [Phương pháp luận và phương pháp nghiên cứu 12](#_bookmark8)
3. [CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN 12](#_bookmark9)
4. [TIẾN ĐỘ THỰC HIỆN ĐỀ TÀI 16](#_bookmark12)
5. [BỐ CỤC DỰ KIẾN CỦA LUẬN VĂN 17](#_bookmark13)

[TÀI LIỆU THAM KHẢO 20](#_TOC_250000)

**DANH MỤC CÁC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **TỪ VIẾT TẮT** | **NỘI DUNG** |
| NLP | Natural Language Processing (Xử lý ngôn ngữ tự nhiên) |
| XLNNTN | Xử lý ngôn ngữ tự nhiên |
| NNTN | Ngôn ngữ tự nhiên |
| IR | Information Retrieval (Truy vấn thông tin) |
| LIR | Legal Information Retrieval (Truy xuất thông tin pháp lý) |
| R2NC | Ranking Related N-gram Collections  (Xếp hạng bộ sưu tập N-gram thu thập) |
| TS | Text Summarization (Tóm tắt văn bản) |

**DANH MỤC CÁC HÌNH**

[Hình 1: Truy vấn thông tin văn bản pháp luật về thuế 7](#_bookmark3)

[Hình 2: Kết quả tìm kiếm trên cơ sở dữ liệu thư viện pháp luật 8](#_bookmark3)

Hình 3: Kết quả tìm kiếm mong muốn khi biết cụ thể từ khóa 8

[.............................................................................................................................................. 13](#_bookmark10)

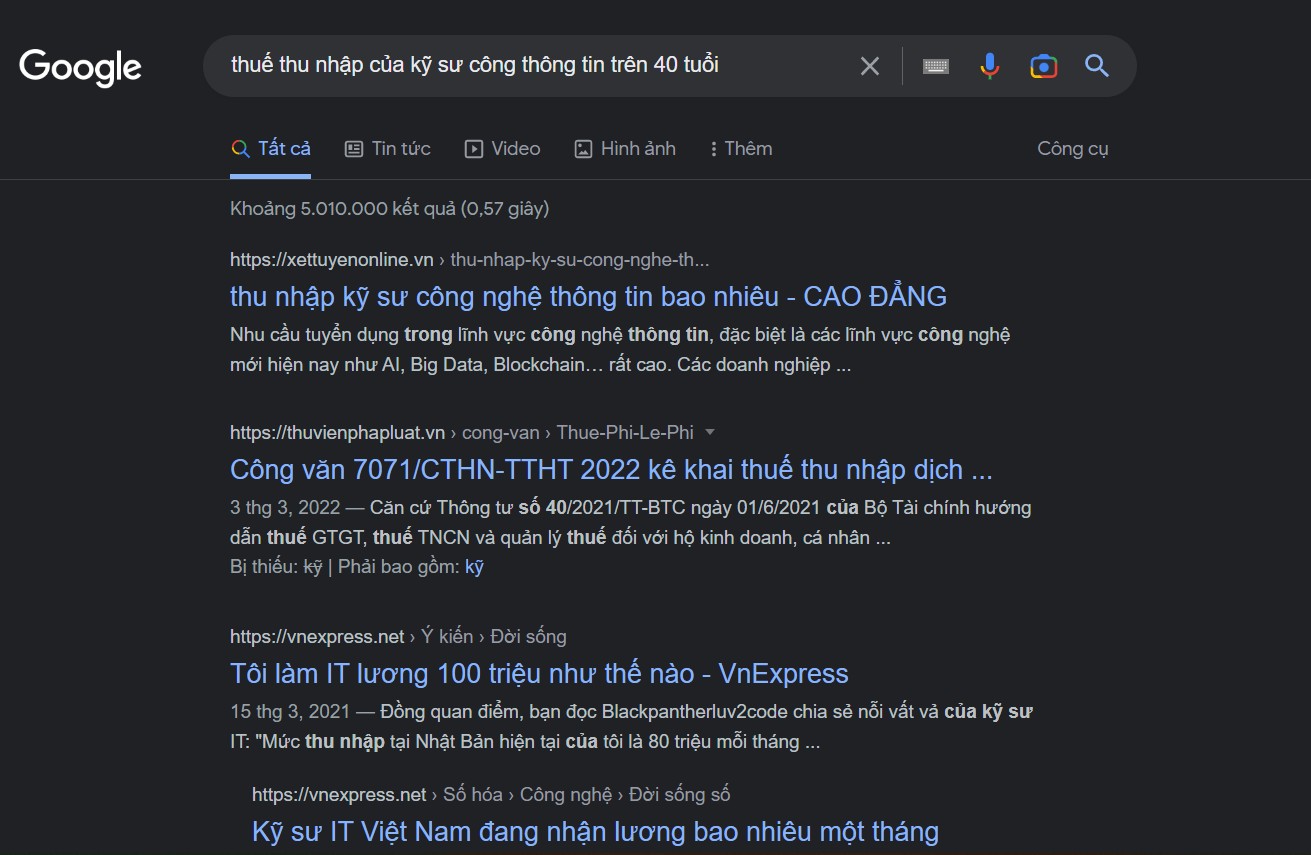
[Hình 4 Quy trình R2NC [5] 14](#_bookmark11)

Hình 5 Tổng quan kiến trúc của hệ thống đề xuất bởi [7] 15

# GIỚI THIỆU

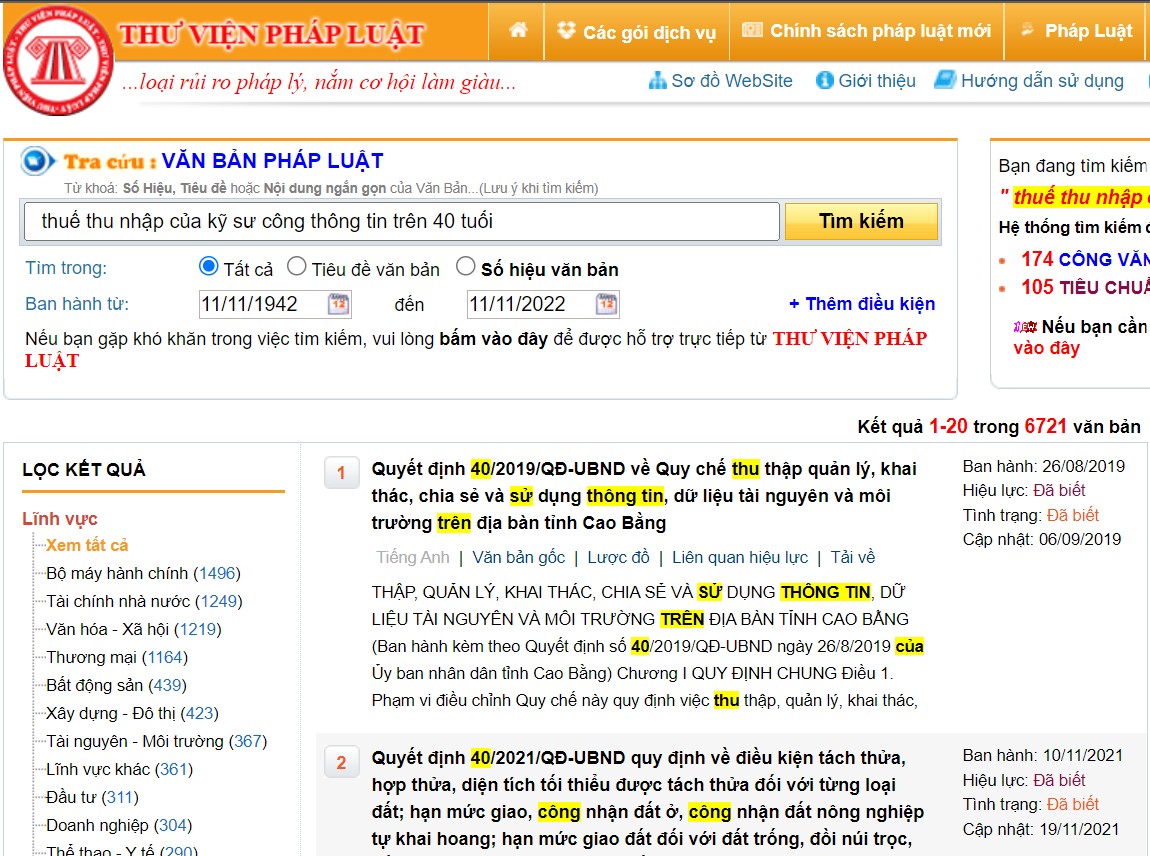
### Giới thiệu tổng quan

Truy vấn thông tin (Information Retrieval - IR) vẫn luôn được ứng dụng trong nhiều lĩnh vực. Điển hình là, các cỗ máy tìm kiếm lớn, như Google Search, Yahoo! Search, Bing Search, Baidu, vẫn đang được ứng dụng để hỗ trợ người sử dụng Internet tìm kiếm thông tin từ các trang web. Các ứng dụng này giúp ích người dùng rất nhiều trong quá trình tìm kiếm thông tin trên phạm vi toàn cầu. Các ứng dụng tìm kiếm có thể nhanh chóng tìm được thông tin theo mong muốn của người dùng vì họ áp dụng kỹ thuật đánh chỉ mục (indexing) các nội dung của các trang web.

Như vậy, các nội dung không thể đánh chỉ mục như các văn bản nội bộ thì ta không thể dễ dàng tìm kiếm trên các cỗ máy tìm kiếm công khai nêu trên. Chẳng hạn như ví dụ trong Hình 1 bên dưới.

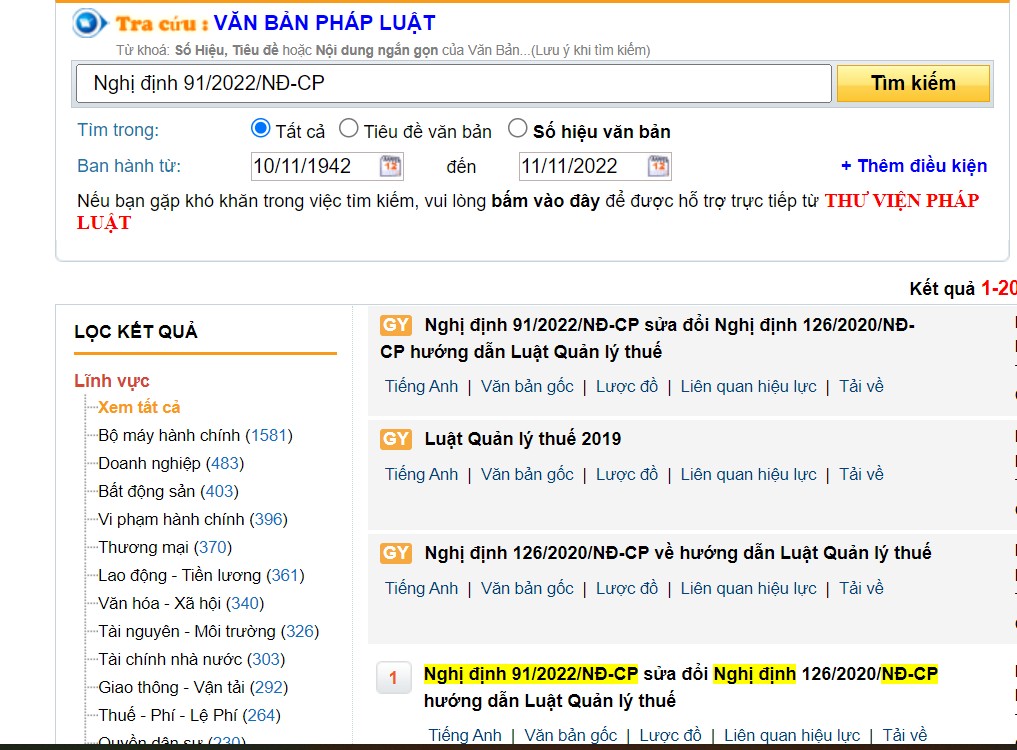
*Hình 1: Truy vấn thông tin văn bản pháp luật về thuế*

Trong ví dụ này, người dùng mong muốn tìm kiếm thông tin về "thuế thu nhập của kỹ sư công thông tin trên 40 tuổi". Khi đó, nếu ta tìm kiếm trên Google, khả năng tìm kiếm thông tin của các cỗ máy tìm kiếm là dựa vào trang thông tin công khai có đăng tin tức liên quan đến các từ khóa có trong câu truy vấn. Tuy nhiên, ta thấy trong trường hợp này kết quả trả về không như mong đợi.

Tương tự vậy, tác giả tìm kiếm thông tin có liên quan đến pháp luật bằng Tiếng Việt tại trang THƯ VIỆN PHÁP LUẬT (https://thuvienphapluat.vn/), kết quả hiển thị trong Hình 2 như bên dưới. Tác giả có thể thấy rằng, kết quả này vẫn chưa đáp ứng đúng nhu cầu thông tin của người dùng.

*Hình 2: Kết quả tìm kiếm trên cơ sở dữ liệu thư viện pháp luật*

Từ quan sát trên, đề tài luận văn này định hướng hỗ trợ người dùng trong quá trình tìm hiểu thông tin về pháp luật bằng cách cải thiện kết quả xếp hạng văn bản trả về.



*Hình 3: Kết quả tìm kiếm mong muốn khi biết cụ thể từ khóa*

Việc tìm kiếm này cần người có chuyên môn tìm kiếm bằng các từ khóa chuyên ví dụ như Nghị định 91/2022/NĐ-CP về quản lý thuế có hiệu lực từ ngày 30/10/2022. Từ đó đặt ra yêu cầu có một phương pháp tìm kiếm phân tích xếp hạng văn bản dự trên câu truy vấn của người dùng trong trường hợp văn bản pháp luật Tiếng Việt.

Do đó, tác giả chọn đề tài **“Cải tiến truy vấn thông tin trong văn bản pháp luật tiếng việt ”** nhằm cải thiện kết quả tìm kiếm thông tin trong miền dữ liệu pháp luật dựa vào các phương pháp XLNNTN.

### Tính cấp thiết của luận văn

Các công cụ tìm kiếm thông tin cần có dữ liệu được cung cấp công khai trên các trang web cho phép tìm kiếm. Tuy nhiên, dữ liệu văn bản pháp luật pháp luật cũng có nhiều văn bản lưu hành nội bộ và không thể trông cậy hoàn toàn vào các cỗ máy tìm kiếm trong tình huống này. Ngoài ra, nếu tác giả áp dụng phương pháp truy vấn thông tin nhị phân vào thì sẽ nhận về các văn bản có chứa đúng các từ khóa xuất hiện trong câu truy vấn. Vậy, những tình huống như đã nêu trong mục 1.1 sẽ gặp khó khăn khi tìm kiếm thông tin. Do đó, tác giả nghiên cứu phương pháp cải thiện kết quả tìm kiếm khi truy vấn thông tin văn bản pháp luật.

Lượng thông tin văn bản đang ngày một gia tăng nhưng tính chính xác trong các nội dung đôi lúc không tương đồng nên cùng một đoạn văn bản cùng lĩnh vực và cùng nội dung đôi lúc nội dung sẽ có những kết quả truy vấn tương tự nhau, vì vậy nhu cầu sử dụng bài toán truy vấn thông tin đang ngày càng trở nên cần thiết, vì nếu có một phương pháp cải tiến truy vấn thông tin hiệu quả thì việc có được kết quả xếp hạng tốt cũng là tiền đề tìm đến sự chính xác theo từng lĩnh vực, từng nội dung một cách tốt hơn.

Trong quá trình nghiên cứu bài toán truy vấn thông tin trong dữ liệu văn bản pháp luật pháp luật tác giả đã đặt ra những câu hỏi cần phải trả lời: Nên đánh chỉ mục đầu vào theo từ, theo cụm từ hoặc kết hợp cả 2 phương pháp nâng cao chất lượng xử lý dữ liệu đầu vào? Làm sao xác định nội dung chính xác vấn đề nào là cần sử lý? Giới hạn bài toán sẽ nằm ở đâu?

Để trả lời được các bài toán này chúng tôi đã đưa ra các giả thuyết phải phân tích và tìm ra được sự liên quan giữa các từ, cụm từ được truy vấn trong các văn bản để tiếp tục xử lý, nếu các từ khóa trên có sự liên quan với các từ trong các văn bản trong miền dữ liệu thì chúng tôi sẽ tiến hành tìm ra các đặc trưng để phân tích ( lấy xử lý dữ liệu) và huấn luyện máy học để phân lớp( đánh index), mở rộng câu truy vấn và xếp hạng các văn bản đó và đưa ra kết quả với hàng loạt văn bản đã xếp hạng.

### Mục tiêu và phương pháp nghiên cứu

Mục tiêu tổng quát của luận văn là cải tiến truy vấn thông tin với dữ liệu đầu vào là các văn bản pháp luật đã được tiền xử lý, phân tích cú pháp và ngữ nghĩa của các từ khóa mở rộng câu truy vấn trong văn bản từ đó tìm ra các đặc trưng để cải tiến xử lý.

Ngày càng có nhiều văn bản pháp luật được cung cấp trong định dạng điện tử. Các thuật toán tìm kiếm và truy vấn thông tin tự động đóng vai trò quan trọng trong việc thực hiện truy vấn hiệu quả vào các tài liệu số hóa như vậy. Mặc dù tìm ki dựa trên từ khóa là phương pháp truyền thống được sử dụng để truy vấn văn bản, chúng thực hiện kém khi đối sánh thuật ngữ theo nghĩa đen được thực hiện để xử lý truy vấn, do từ đồng nghĩa và sự mâu thuẫn của các từ. Để khắc phục những nhược điểm này, một khuôn khổ bản thể học để nâng cao truy vấn của người dùng về việc truy vấn các văn bản pháp lý thực sự có liên quan đã được đề xuất trong nghiên cứu này [1].

Nghiên cứu dự kiến được thực nghiệm trên tập dữ liệu là các văn bản pháp luật được ban hành từ trang thư viện pháp luật nhóm văn bản pháp luật. Chúng tôi thu thập các văn bản này thư viện pháp luật quản lý bởi công ty THƯ VIỆN PHÁP LUẬT, bao gồm khoảng [299,290](https://thuvienphapluat.vn/page/tim-van-ban.aspx?keyword&match=True&area=0) văn bản, quy định, quyết định của pháp luật, trong đó có 16,417 Văn bản tiêu chuẩn Việt Nam.

Nghiên cứu các phương pháp xếp hạng kết quả từ câu truy vấn. Áp dụng mô hình Vector Space Model trong quá trình truy vấn dựa vào tập các phép toán.

Elasticsearch sử dụng "index" để lưu trữ và xử lý dữ liệu, nó tương tự một cơ sở dữ liệu thông thường. Dữ liệu sẽ được lưu trữ thành các "tài liệu" dưới dạng JSON.

Tập trung nghiên cứu mô hình truy xuất thông tin Boolean, liên quan đến việc sử dụng các hàm logic trong truy vấn để lấy dữ liệu cần thiết. Đây là cách tiếp cận sớm cho dữ liệu truy xuất và được sử dụng làm mô hình Boolean trong việc tìm kiếm thông tin. Mô hình này dựa trên lý thuyết tập hợp và đại số Boolean, chúng cùng nhau tạo thành một mô hình để xác định dữ liệu. Tài liệu đang được tìm kiếm trong cơ sở dữ liệu thu thập là tập hợp các phép toán trong khi truy xuất thông tin [2].

Tập trung nghiên cứu thuật toán xếp hạng, vì Boolean không có cơ chế xếp hạng, nó có thể bỏ qua dữ liệu quan trọng, vì vậy cần phải xếp hạng. Kết quả được xếp hạng dựa trên sự xuất hiện của các phép toán trong các truy vấn. Phương pháp này loại bỏ Boolean thường sai cú pháp được sử dụng bởi người dùng cuối, và cung cấp một số kết quả ngay cả khi một phép toán của truy vấn không được chính xác [2].

Tập trung nghiên cứu mô hình không gian vector. Mô hình không gian vectơ là một tài liệu đại số sử dụng vectơ để biểu diễn. Tài liệu và truy vấn đều là vectơ. Tiềnđề cơ bản của việc áp dụng mô hình không gian vectơ là các đối tượng truy xuất thông tin khác nhau được mô hình hóa như các phần tử của không gian vectơ [3].

Tập trung nghiên cứu kĩ về gói phần mềm elasticsearch có thể thử nghiệm với các mô hình vector không gian và có thể triển khai các dịch vụ tìm kiếm dựa trên chúng.

Phương pháp nghiên cứu là tìm hiểu cách hoạt động của Information Retrieval (Truy vấn thông tin), là một hoạt động chỉ có một số ít người tham gia: các thủ thư, các trợ lý luật sư hay các ngành nghề tương tự đòi hỏi nghiệp vụ tra cứu. IR xoay quanh việc tìm kiếm trên dữ liệu phi cấu trúc, tức những dữ liệu có ngữ nghĩa không được xác định (undefined semantics) nhằm thỏa mãn một nhu cầu thông tin nào đó (information need) [4].

Nghiên cứu các thức xếp hạng kết quả từ câu truy vấn của người truy xuất thông tin. Phương pháp này cũng hoạt động tốt đối với các truy vấn phức tạp mà người dùng có thể khó diễn đạt bằng cách sử dụng toán tử Boolean [2].

Nghiên cứu cách áp dụng của Vector Space Model trong quá trình truy vấn dựa vào tập các phép toán [2]. Và đây là một mô hình đơn giản dựa trên đại số tuyến tính và không có hệ nhị phân nào được sử dụng. Nó cho phép tiếp tục đo khoảng cách giữa tài liệu và truy vấn.

Nghiêm cứu về cách thức hoạt động của công cụ Elasticsearch, nó là một công cụ tìm kiếm dựa trên thư viện Lucene. Nó cung cấp một công cụ tìm kiếm toàn văn phân tán, đa đối tượng có khả năng cung cấp với giao diện web HTTP và các tài liệu JSON không có giản đồ [2]. Elasticsearch sử dụng "index" để lưu trữ và xử lý dữ liệu, nó tương tự một cơ sở dữ liệu thông thường. Dữ liệu sẽ được lưu trữ thành các "tài liệu" dưới dạng JSON. Mỗi một index có nhiều "type" (tương tự như bảng trong cơ sở dữ liệu), điều này sẽ giúp tác giả phân tách các loại dữ liệu khác nhau khi lưu vào cùng một index.

### Đóng góp của đề tài

Trong luận văn này, tác giả đã đóng góp những nội dung như sau:

1. Giới thiệu cách tiếp cận **Lucene** cho việc phân tích, đánh chỉ mục và tìm kiếm thông tin với hiệu suất cao trong văn bản pháp luật Tiếng Việt. Trong đó cách tiếp cận này có xem xét véc tơ chủ đề đối với trọng số của mỗi đỉnh

2. Thu thập bộ dữ liệu từ khóa từ bài báo tin tức Tiếng Việt.

3. Đánh giá hiệu suất hoạt động của phương pháp đã đề xuất so với cách tiếp cận gốc trên dữ liệu tin tức Tiếng Việt.

### Bố cục luận văn

Nội dung mà luận văn nghiên cứu bao gồm: Tìm hiểu tổng quan về các hệ thống tìm kiếm thông tin; Tìm hiểu tổng quan về công nghệ tìm kiếm mã nguồn mở Lucene; Phân tích, đề xuất phương án cải tiến, đồng thời đánh giá kết quả thông qua tập dữ liệu liên quan, được tổng hợp một cách khách quan giữa nhiều cá nhân với nhau.

Bố cục của luận văn như sau:

Chương 1: Giới thiệu tổng quan về đề tài nghiên cứu

Chương 2: Cơ sở lý thuyết

Chương 3: Phương pháp thực hiện của mã nguồn mở Lucene, đề xuất cải tiến gán nhãn dữ liệu mở mở rộng câu truy vấn

Chương 4: Thực nghiệm và đánh giá

Chương 5: Kết luận và hướng phát triển

# CƠ SỞ LÝ THUYẾT



## 2.1. Các công trình nghiên cứu liên quan

Trước tiên chúng tôi giới thiệu tổng quan về Lucene Core thư viện được dùng trong bài, Lucene Core là một thư viện Java cung cấp các tính năng lập chỉ mục và tìm kiếm mạnh mẽ, cũng như kiểm tra chính tả và analysis/tokenization.

**Lucene** là phần mềm mã nguồn mở, dùng để phân tích, đánh chỉ mục và tìm kiếm thông tin với hiệu suất cao bằng Java. Lucene được phát triển đầu tiên bởi Doug Cutting được giới thiệu đầu tiên vào tháng 8 năm 2000. Tháng 9 năm 2001 Lucene gia nhập vào tổ chức Apache và hiện tại được Apache phát triển và quản lý. Lucene không phải là một ứng dụng mà chỉ là một công cụ đặc tả API cần thiết cho việc một search engine.

Lucene hỗ trợ cho việc đánh index- chỉ mục và tìm kiếm. Dữ liệu tìm kiếm thô có thể là tập tin dạng PDF, Word hay HTML, Json,... hoặc dữ liệu trong các hệ quản trị dữ liệu như MS SQL, SQL server hay MySQL.

Lucene bắt đầu được phát triển trên ngôn ngữ java, đến nay thì được phát triển bằng nhiều ngôn ngũ như Perl, C++/C#, Python, Ruby, PHP . Được xây dựng và thiết kế theo hướng hướng đối tượng nên các API cũng được cung cấp theo dạng hướng đối tượng. Lucene cho phép người sử dụng kế thừa và phát triển để phù hợp với hiện trạng bùng nổ dữ liệu hiện nay. Chúng tôi đề xuất nghiên cứu mã nguồn mở Lucene để phát triển hệ thống tìm kiếm và tìm phương án cải thiện với ngôn ngữ Tiếng Việt.

Và đây là các phiên bản ngôn ngữ khác nhau của Lucene:

- Beagle dùng một nhánh của Lucene phát triển trong C#, có tên gọi Lucene.Net làm chỉ mục.

- Docco dùng Lucene trong việc tìm kiếm trong máy tính cá nhân.

- CNET dùng Lucene để tìm kiếm danh sách thể loại sản phẩm. - LjFind dùng Lucene để tìm kiếm hơn 110.000.000 bài post ở LiveJournal. - Nutch là một máy tìm kiếm dùng Lucene.

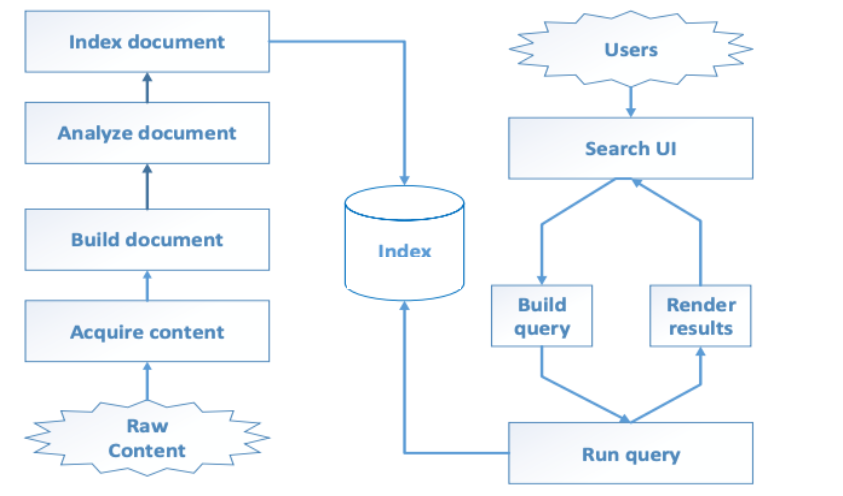
- Red-Piranha cũng là một máy tìm kiếm khác dựa trên Lucene - Wikipedia dùng Lucene để tìm kiếm nội dung toàn bộ văn bản.

- Trình duyệt web Flock dùng Clucene, một phiên bản trong C++, để tìm kiếm toàn văn hoặc tìm kiếm lịch sử của trình duyệt.

- Ants P2P dùng Lucene trong lựa chọn tìm kiếm trong chương trình chia sẻ file khuyết danh của nó.

- Solr một máy chủ tìm kiếm nguồn mở dựa trên Lucene với XML/HTTP APIs, lưu trữ (cache), sao chép, và một giao diện web quản trị.

- LIRE – Lucene Image Retrieval Thư viện CBIR, dùng máy tìm kiếm Lucene.



Hình 2.1 Mô hình Lucene trong hệ thống tìm kiếm thông tin [7].

Các công trình nghiên cứu liên quan dựa trên sự cải tiến Lucene nổi bật có thể nhắc đến là : Elasticsearch, Solr ….

Elasticsearch phương pháp hỗ trợ được xây dựng trên Lucene và cung cấp API REST dựa trên JSON để tham khảo các tính năng của Lucene. Elasticsearch cung cấp một hệ thống phân tán trên đỉnh Lucene . Một hệ thống phân tán không phải là thứ mà Lucene nhận thức được hoặc được xây dựng cho. Elasticsearch cung cấp sự trừu tượng hóa của cấu trúc phân tán.

Elasticsearch cung cấp các tính năng hỗ trợ khác như nhóm luồng, hàng đợi, API giám sát nút / cụm, API giám sát dữ liệu, quản lý cụm, v.v.

Tính sẵn sàng cao: Elasticsearch được phân phối, để nó có thể quản lý sao chép dữ liệu, có nghĩa là có nhiều bản sao dữ liệu trong cụm của bạn. Điều này cho phép tính sẵn sàng cao.

* DSL truy vấn mạnh mẽ : Elasticsearch cung cấp cho chúng tôi, giao diện JSON để đọc và viết các truy vấn trên Lucene. Nhờ Elasticsearch, bạn có thể viết các truy vấn phức tạp mà không cần biết cú pháp Lucene.
* Schemaless : Các trường (tên, cặp giá trị) schema không phải được xác định trước. Khi bạn lập chỉ mục dữ liệu, Elasticsearch có thể tự động tạo lược đồ khi chạy, như ma thuật.

Lucene là một thư viện công cụ tìm kiếm . Sử dụng nó để xây dựng công cụ tìm kiếm của riêng mình: một đối thủ cạnh tranh mới của Elasticsearch hoặc Solr hoặc một cái gì đó hẹp cho trường hợp sử dụng của bạn

Elasticsearch là một công cụ tìm kiếm . Hầu hết mọi người sử dụng nó để tổng hợp nhật ký, tìm kiếm sản phẩm hoặc một biến thể.

Thành phần chức năng chính của Lucene bao gồm hai phần: Thành phần tạo chỉ mục và thành phần tìm kiếm. Đây là hai thành phần quan trọng cho một hệ thống tìm kiếm thông tin.

Thành phần Tạo chỉ mục: Bao gồm các chức năng xử lý và phân tích dữ liệu để đánh chỉ mục. Lucene cho phép thiết lập các trường thông tin cần thiết để đánh chỉ mục phục vụ cho thành phần tìm kiếm, các thư viện phục vụ đánh chỉ mục mà Lucene hỗ trợ.

Thành phần Tìm kiếm: bao gồm các phần chức năng xử lý tìm kiếm, trả về kết quả tìm kiếm cho người dùng, thông qua biên dịch và so khớp để lấy về kết quả tốt nhất.

Đánh chỉ mục (Index): Đầu tiên, Lucene giúp bạn phân loại chỉ mục, quản lý như các Document. Việc đánh chỉ mục được thực hiện qua những bước sau.

* Đầu tiên, thu thập dữ liệu (Acquire content), ở bước này thường là crawler hoặc spider để thu thập dữ liệu đánh chỉ mục.
* Tiếp theo là bước xây dựng tài liệu (Build document). Ở bước này thì dữ liệu thô thu nhập ở trên được chia thành các document và các trường text, nhưng các search engine vẫn chưa đánh chỉ mục luôn được mà cần được chuyển sang phân tích tài liệu.
* Phân tích tài liệu (Analyze document): Ở đây text được chia nhỏ thành các token, mỗi token được hiểu là một từ trong tài liệu. Document được xác định bởi dãy các token
* Bước cuối cùng là đánh chỉ mục. Lucene cung cấp API đầy đủ cho việc đánh chỉ mục trở nên dễ dàng hơn.

Truy vấn (query): Khi người sử dụng gửi lên 1 submit thì hệ thống search engine cần phân tích thành câu query cho hệ thống có thể hiểu, bước này gọi là (Build query). Lucene cung cấp gói QueryParser để biến câu truy vấn của người sử dụng thành câu query của hệ thống.Truy vấn có thể là các toán tử boolean hoặc các phép toán liên quan, hoặc tổng hợp, phân tích,...

Tìm kiếm (search): Là quá trình tra cứu và tìm ra kết quả phù hợp với câu truy vấn của người sử dụng và đưa ra kết quả. Ở đây thường query theo 3 mô hình: Mô hình toán tử Boolean, mô hình vector, mô hình xác suất. Những mô hình này đã được giới thiệu trong bài truy vấn thông tin (IR).

Display: Render các kết quả tìm kiếm sắp xếp theo thứ tự phù hợp và hiển thị cho người sử dụng. Ở đây, kết thúc một phiên tìm kiếm.

## 2.2. Mô hình đánh chỉ mục ngược

Hình 2.2 Mô hình đánh chỉ mục của Lucene [7].

Trong mô hình đánh chỉ mục của Lucene để tiến hành đánh chỉ mục (index) trước tiên tác giả chuyển đổi dữ liệu thành dạng văn bản thuần túy plain text, như file .txt. Điều này là quan trọng, bởi vì dữ liệu được lưu trữ dưới nhiều dạng file khác nhau (pdf, word, excel, powerpoint, html…), trong khi đó Lucene chỉ hỗ trợ bạn index các Field dạng String , Date hoặc đối tượng Reader. Mỗi khi bạn chuẩn bị cho việc index và tạo ra đối tượng Document với các Field, thì Lucene sẽ phân tích dữ liệu này sao cho phù hợp nhất với việc index.

Để làm điều này Lucene sẽ phân chia dữ liệu thành các chuỗi hoặc là các kí tự thông qua việc lựa chọn các toán tử thực thi trên chúng. Chẳng hạn như việc tác giả phân tích thành các kí tự thường, hoặc bỏ đi các từ ngữ không có nghĩa… Sau khi dữ liệu được phân tích, nó sẽ sẵn sang cho việc index Lucene sẽ chứa dữ liệu này theo cấu trúc inverted index (chỉ mục có thể nghịch đảo ). Cấu trúc này sẽ có hiệu quả để tiết kiệm dung lượng ổ đĩa và cho phép tìm kiếm nhanh hơn các từ khóa trong quá trình search. Nguyên tắc của nó là thay vì phải tìm kiếm các từ nào chứa trong tài liệu đó thì với cấu trúc này sẽ tối ưu hóa việc tìm ra câu trả lời “tài liệu nào chứa từ khóa này”.

#### **Khung nhìn logic của việc đánh chỉ mục**

Một chỉ mục Lucene được lưu trong một thư mục trong hệ thống tệp trên đĩa cứng. Những thành phần quan trọng của một chỉ mục Lucene là các phân đoạn (segment), các tài liệu (document), các trường (field) và các term. Mọi chỉ mục đều bao gồm một hoặc nhiểu phân đoạn. Mỗi phân đoạn chứa một hoặc nhiều tài liệu. Mỗi tài liệu có một hoặc nhiều trường và mỗi trường chứa một hoặc nhiều term. Một trường có thể là văn bản tự do, các từ khoá, ngày tháng. Mỗi term là một cặp xâu biểu diễn một tên trường và một giá trị. Một phân đoạn bao gồm một chuỗi các tệp. Số lượng chính xác các tệp hình thành nên phân đoạn thay đổi tuỳ chỉ mục và phụ thuộc vào số các trường mà chỉ mục chứa. Tất cả các tệp thuộc về cùng một phân đoạn sẽ có chung tiền tố và chỉ khác nhau phần hậu tố. Có thể coi một phân đoạn là một chỉ mục phụ (sub-index), mặc dù mỗi phân đoạn không phải là một chỉ mục hoàn toàn độc lập. Tất cả các thành phần trong một chỉ mục được cố định một khi chỉ mục được xây dựng hoàn chỉnh, còn trong quá trình lập chỉ mục thì con số này biến đổi.

Hình 2.3 Diagram, text

Description automatically generatedKhung nhìn logic của việc đánh chỉ mục Lucene [7].

#### **Chỉ mục ngược**

Phát triển và duy trì một *inverted index* là vấn đề chủ chốt khi xây dựng một hệ thống tìm kiếm thông tin hiệu quả. Để lập chỉ mục một tài liệu văn bản, trước hết phải kiểm tra qua từng tài liệu để xây dựng một danh sách các *posting*. Các *posting* thực hiện mô tả cách thể hiện của một từ trong một tài liệu văn bản. *Posting* thường gồm một từ, một tệp tài liệu có gán mã định danh và có thể có cả các vị trí hoặc tần số của từ đó trong tài liệu thu thập.

Nếu coi các *posting* là những bộ (*tuple*) có dạng <từ, mã tài liệu> thì một tập các tài liệu sẽ hình thành một danh sách các *posting* được sắp xếp theo mã tài liệu. Nhưng để truy vấn hiệu quả các tài liệu văn bản có chứa những từ cụ thể thì cần phải sắp xếp các *posting* theo từ (hoặc bằng cả từ và tài liệu, sẽ làm cho việc tìm kiếm nhiều từ nhanh hơn). Theo hướng này, việc xây dựng một chỉ mục tìm kiếm về cơ bản là một bài toán sắp xếp. Chỉ mục tìm kiếm là một danh mục các posting được sắp xếp theo từ khóa.

Hầu hết các hệ thống truy vấn thông tin đều sử dụng cấu trúc B-tree để duy trì chỉ mục. Lucene có hướng tiếp cận khác một chút: thay vì duy trì một chỉ mục duy nhất, Lucene được xây dựng từ nhiều thành phần đã được đánh chỉ mục và kết hợp các phân đoạn đó lại với nhau một cách thống nhất. Với bộ tài liệu văn bản mới được đánh chỉ mục, Lucene lại tiếp tục tạo một thành phần đánh chỉ mục mới, thành phần đó sẽ nhanh chóng kết hợp các thành phần chỉ mục nhỏ hoặc với những thành phần chỉ mục lớn hơn, việc này giữ cho tất cả các thành phần nhỏ nhằm đảm bảo tốc độ truy vấn thông tin nhanh hơn. Việc tối ưu đánh chỉ mục hỗ trợ cho việc tìm kiếm thông tin được nhanh nhất, Lucene có thể kết hợp tất cả các thành phần chỉ mục đã được đánh index thành một, điều này có ích với các chỉ mục hiếm khi được cập nhật. Nhằm phòng ngừa xung đột giữa phần đọc và phần ghi chỉ mục, Lucene không bao giờ thay đổi ngay các thành phần chỉ mục mà nó chỉ tạo ra phân đoạn (thành phần đánh chỉ mục) mới. Việc kết hợp các thành phần chỉ mục, Lucene thực hiện ghi một thành phần mới và xoá những thành phần cũ – sau khi bất kỳ bộ đọc nào đã đóng nó.



Hình 2.4 Ví dụ minh hoạ cho định dạng đánh chỉ chỉ mục của Lucene [7].

Một phân đoạn Lucene gồm một số tệp:

• Một chỉ mục từ điển chứa một mục cho mỗi 100 mục trong từ điển

• Một từ điển chứa một mục cho mỗi từ duy nhất

• Một tệp các posting chứa một mục cho từng posting

Vì Lucene không cập nhật “tại chỗ” các phân đoạn nên các phân đoạn có thể lưu được trong các tệp flat đơn giản thay vì cấu trúc B-trees phức tạp.

Hình 2.4 biểu diễn một thành phần chỉ mục ví dụ. Phần này là của một phân đoạn duy nhất (trong trường hợp có thực hiện đánh chỉ mục được chuẩn hoá với một thành phần duy nhất). Một thành phần được đặt một tiền tố tên tệp dữ liệu duy nhất.

* **Tên trường (.fnm)**

File dữ liệu có đuôi .fnm bao gồm tất cả các tên trường được sử dụng bởi các tài liệu trong một phân đoạn. Các trường dữ liệu đã được đánh dấu để thể hiện các thuộc tính dữ liệu của nó. Sắp xếp các tên trường dữ liệu trong tệp .fnm được xác định trong quá trình đánh chỉ mục và việc này không cần thiết phải tuân theo thứ tự. Vị trí của một trường trong tệp .fnm được dùng để liên kết nó với các tệp được chuẩn hoá (các tệp với hậu tố .f[0-9]\*). Trong ví dụ này, chỉ có trường *subject* được vectơ hoá. Trường *url* được thêm vào như một trường *Field.UnIndexed*, không được lập chỉ mục mà cũng không được vectơ hoá.

* **Từ điển term (.tis)**

Tất cả các term (các bộ gồm tên trường và giá trị) trong một phân đoạn được lưu trữ trong tệp .tis. Danh sách các từ được sắp xếp trước hết phải tuân theo thứ tự quy định của tên trường tương ứng với bản chữ cái và rồi theo giá trị dữ liệu trong một trường dữ liệu. Mỗi mục term có chứa tần số thể hiện tài liệu (*document frequency*) của nó: là số các tài liệu văn bản chứa các từ này trong phân đoạn.

Hình 2.4 chỉ minh hoạ một phần các term trong chỉ mục, ít nhất một term ứng với mỗi trường. Chú ý rằng trường *url* được bỏ qua vì nó được thêm vào dưới dạng một trường *UnIndexed*, chỉ được lưu trữ và không thể được coi như các term. Đối với mỗi term trong tệp .tis, tệp .frq chứa các mục cho từng tài liệu có chứa term đó.

* **Tần số term**

Các tần số term trong mỗi tài liệu được liệt kê trong tệp .frq. Tần số của một từ trong một tài liệu văn bản được dùng làm thừa số để tính toán xếp hạng và thường tăng độ liên quan của một tài liệu văn bản khi nó có tần số cao nhất.

Đối với mỗi tài liệu được liệt kê trong tệp .frq, tệp vị trí (*.prx*) chứa các mục cho từng thể hiện dữ liệu của từ trong một tài liệu văn bản.

* **Vị trí term**

Tệp dữ liệu .prx thực hiện liệt kê vị trí của từng từ trong một tài liệu văn bản. Thông tin dữ liệu vị trí này được dùng khi các truy vấn cần tới, chẳng hạn như các truy vấn cụm từ. Thông tin vị trí của các trường đã được tách được lấy trực tiếp từ các trường dữ liệu vị trí từ tăng dần được chỉ định trong quá trình phân tích văn bản.

Rõ ràng, *cấu trúc chỉ mục của Lucene đáp ứng hai yêu cầu: tối đa hiệu suất và tối thiểu tài nguyên cần sử dụng*. Thật vậy, nếu một trường không được lập chỉ mục thì có thể nhanh chóng loại bỏ hoàn toàn nó khỏi các truy vấn dựa trên cờ indexed trong tệp .fnm. Tệp .tii, được lưu trên RAM, cho phép truy xuất ngẫu nhiên vào tệp từ điển term .tis. Các truy vấn cụm từ không cần phải tìm thông tin vị trí nếu như term đó không xuất hiện. Việc tổ chức thông tin hợp lý là tối quan trọng và việc tối thiểu hoá số lần truy nhập tệp trong quá trình truy vấn thông tin là vấn đề quyết định. Đó là một số ví dụ thể hiện ưu điểm của cách thiết kế cấu trúc chỉ mục trong Lucene.

## 2.3. Tìm kiếm

#### **Mô hình biểu diễn không gian vectơ**

Mô hình biểu diễn không gian vectơ là sự biểu diễn tập các tài liệu văn bản dưới dạng các vectơ trong một không gian vectơ. Cách biểu diễn này là trọng tâm của các thao tác truy xuất thông tin, đặc biệt là việc xếp hạng độ liên quan văn bản của các tài liệu đối với các câu truy vấn. Bước then chốt của hướng tiếp cận này là việc xem các truy vấn là các vectơ trong cùng hệ không gian vectơ của tập tài liệu.

* **Tích nội**

Ký hiệu *(d)* là vectơ xuất phát từ tài liệu *d*. Giả sử các phần tử được xử lý tính toán bằng lược đồ gán trọng số *tf-idf*. Tập các tài liệu văn bản trong một tập hợp được chuyển thành một không gian vectơ mà mỗi chiều là mỗi trục ứng với một term. Cách biểu diễn này không thể hiện được thứ tự giữa các term trong từng tài liệu.

Diagram

Description automatically generated

Hình 2.5 Minh họa độ tương tự côsin.

Để định lượng giữa hai tài liệu có độ tương tự trong không gian vectơ này, có thể xét độ lớn của hiệu vectơ giữa hai vectơ. Phép đo này có hai nhược điểm: hai tài liệu có sự phân bố term rất giống nhau có thể có hiệu vectơ lớn đáng kể chỉ bởi vì một vectơ lớn hơn nhiều so với vectơ còn lại. Vì vậy, sự phân bố liên quan giữa các term có thể giống nhau trên cả hai tài liệu nhưng tần số term thực sự của vectơ thứ nhất có thể lớn hơn nhiều.

Đề bù lại tác động của độ dài tài liệu, phương pháp chuẩn để định lượng giữa hai tài liệu có độ tương tự *d1* và *d2* là phải tính độ tương đồng côsin của các biểu diễn vectơ tương ứng của chúng (*d1*) và (*d2*):

|  |  |
| --- | --- |
|  | (1) |

Trong đó, tử số biểu biễn tích trong (còn gọi là tích chấm) của các vectơ (*d1*) và (*d2*), còn tử số là tích độ dài của hai vectơ. Vai trò của tử số là để chuẩn hoá các vectơ (*d1*) và (*d2*) thành các vectơ đơn vị và . Tác giả có thể thay thế công thức chung (1) như sau:

|  |  |
| --- | --- |
|  | (2) |

Có thể coi công thức (2) là tích trong của phiên bản đã được chuẩn hoá của hai vectơ tài liệu. Xét tập n vectơ đã chuẩn hoá biểu diễn các tài liệu trong một bộ dữ liệu. Với một tài liệu văn bản d (một trong các tài liệu văn bản *d* của tập hợp), xét việc truy vấn thông tin trong tập hợp những tài liệu giống d nhất. Việc tìm kiếm như vậy có ích trong một mô hình tìm kiếm mà ở đó người dùng có thể xác định một tài liệu và tìm kiếm những tài liệu khác giống với nó. Vấn đề tìm kiếm những tài liệu giống d nhất trở thành việc tìm các di có các tích trong (giá trị sim) lớn nhất. Điều này tất nhiên thực hiện được bằng cách tính các tích trọng giữa với từng , rồi chọn ra những vectơ có chỉ số sim lớn nhất.

Một bộ tập hợp n tài liệu có thể xem là một bộ tập hợp các vectơ, điều này dẫn tới cách nhìn một tập hợp là một ma trận term-tài liệu: một ma trận m x n mà các hàng của nó biểu diễn m term (các chiều) của n cột, mỗi cột ứng với một tài liệu.

* **Biểu diễn truy vấn**

Lý do chính để tài liệu được biễu diễn dưới dạng các vectơ là ý tưởng coi như một truy xuất là một vectơ. Xét truy xuất *q*, với thể hiện q dưới dạng vectơ đơn vị , có số chiều bằng với số *m* các term phân biệt được lập chỉ mục. Ý tưởng then chốt là: gán cho mỗi tài liệu *d* một giá trị bằng với tích trong:

Như vậy, bằng cách xem một truy vấn là một “túi từ”, có thể coi truy vấn là một tài liệu ngắn. Nhờ vậy, có thể sử dụng côsin có độ tương tự giữa vectơ truy xuất với một vectơ văn bản làm độ đo độ cụ thể về việc có liên quan với tài liệu với truy xuất đó. Tiếp theo, các giá trị độ liên quan có thể sử dụng để chọn ra những tài liệu văn bản có độ liên quan cao nhất đối với câu truy vấn.

#### **Xếp hạng**

Mô hình truy xuất của Lucene là sự biến đổi dựa trên mô hình không gian vectơ. Thực tế, đó là sự liên kết và kết hợp của mô hình không gian vectơ và mô hình Boolean nhằm xác định độ liên quan của bộ tài liệu với một tài liệu đối với một câu truy vấn. Ý tưởng đằng sau mô hình không gian vectơ là một term truy vấn xuất hiện càng nhiều lần trong một bộ tài liệu, cân đối với số lần term đó xuất hiện trong tất cả các tài liệu của bộ tài liệu liên quan thì tài liệu đó càng có độ liên quan cao với truy vấn. Lucence sử dụng mô hình Boolean nhằm trước hết thu hẹp số lượng tài liệu cần xếp hạng dựa vào việc sử dụng lôgíc boolean trong đặc tả truy vấn. Lucene cũng bổ sung một số khả năng và sự tinh lọc đối với mô hình này để hỗ trợ tìm kiếm mờ và tìm kiếm boolean, nhưng về bản chất nó vẫn là một hệ thống dựa vào mô hình không gian vectơ.

Thật vậy, câu truy vấn q có độ tương tự đối với tài liệu d tương quan với khoảng cách côsin hay tích chấm giữa vectơ tài liệu và vectơ truy vấn trong một mô hình không gian vectơ. Một tài liệu văn bản có vectơ gần hơn với vectơ của truy vấn trong mô hình mô hình không gian vectơ thì được xếp hạng cao hơn. Độ tương tự này được tính toán theo công thức 3:

|  |  |
| --- | --- |
|  | (3) |

Trong đó:

tf(t trong d) tương quan với tần số term (f), được định nghĩa là số lần term t xuất hiện trong bộ tài liệu đang được xét d. Những tài liệu có càng nhiều thể hiện của một term thì nhận điểm xếp hạng càng cao. Công thức mặc định để tính tf(t trong d) là:

|  |  |
| --- | --- |
|  | (4) |

idf(t) tương quan với nghịch đảo của docFreq (số các tài liệu mà trong đó term t xuất hiện). Nghĩa là các term càng hiếm khi xuất hiện thì điểm xếp hạng tổng thể càng cao. Công thức mặc định để tính idf(t) là (với numDocs là số lượng tài liệu có trong tập hợp):

|  |  |
| --- | --- |
|  | (5) |

coord(q,d) là thừa số tính điểm xếp hạng dựa trên số lượng các term của truy vấn q được tìm thấy trong một tài liệu cụ thể d. Thông thường, một tài liệu chứa càng nhiều các term truy vấn thì sẽ nhận điểm cao hơn so với tài liệu khác có chứa ít term hơn. Thừa số này được tính khi thực hiện việc tìm kiếm.

queryNorm(q) là thừa số được chuẩn hoá để dùng làm cho các điểm xếp hạng ứng với các truy vấn q khác nhau (hoặc thậm chí là ứng với các chỉ mục khác nhau) và có thể so sánh được. Thừa số này không tác động tới việc xếp hạng các tài liệu văn bản (vì tất cả các bộ tài liệu văn bản được xếp hạng đều được nhân với cùng một thừa số). Thừa số này được tính khi thực hiện việc tìm kiếm. Công thức tính mặc định là:

|  |  |
| --- | --- |
|  | (6) |

t.getBoost() là giá trị tăng mức độ quan trọng (boost) tại thời điểm tìm kiếm của term t trong truy vấn q.

norm(t,d) bao gồm một số thừa số tăng mức độ quan trọng và độ dài được tính vào thời gian tìm kiếm:

Thừa số tăng mức độ quan trọng của tài liệu doc: được thiết lập trước khi thêm tài liệu vào chỉ mục.

Thừa số tăng mức độ quan trọng của trường f: được thiết lập trước khi thêm trường vào một tài liệu.

lengthNorm(field) được tính khi tài liệu được thêm vào chỉ mục, phù hợp với số lượng các token của trường này trong tài liệu. Do đó, những trường càng ngắn thì càng làm tăng điểm xếp hạng.

Khi một tài liệu được bổ sung vào chỉ mục thì tất cả các thừa số trên được nhân với nhau. Nếu như tài liệu d có nhiều trường f trùng tên thì tất cả các giá trị tăng mức độ quan trọng của chúng được nhân cùng với nhau:

|  |  |
| --- | --- |
|  | (7) |

Tuy nhiên, giá trị norm được mã hoá thành một byte trước khi lưu. Vào thời gian tìm kiếm, giá trị byte đó được đọc từ thư mục chỉ mục và giải mã trở lại thành một giá trị norm. Việc mã hoá/giải mã này, trong khi làm giảm kích thước chỉ mục thì lại làm giảm độ chính xác, không thể đảm bảo rằng giải mã(mã hoá(x)) = x. Chẳng hạn giải mã(mã hoá(0.89)) = 0.75.

Các thừa số tăng mức độ quan trọng được đưa vào phương trình để cho phép điều chỉnh ảnh hưởng của một truy vấn hoặc trường tới độ tương tự. Các thừa số tăng mức độ quan trọng của các trường xuất hiện tường minh trong phương trình dưới dạng thừa số getBoost(t.field trong d) được thiết lập trong quá trình đánh chỉ mục. Giá trị mặc định của các thừa số tăng mức độ quan trọng của các trường, về mặt lôgíc, là 1.0.

Ngoài những thừa số được thể hiện tường minh trong phương trình, các thừa số khác có thể được tính trên cơ sở từng truy vấn như là thành phần của thừa số queryNorm. Bản thân các truy vấn có thể có ảnh hưởng tới độ tương tự của tài liệu. Tăng tầm quan trọng để thể hiện của Query chỉ có ích trong một truy vấn nhiều mệnh đề; nếu chỉ có một term duy nhất được dùng để tìm kiếm thì việc tăng tầm quan trọng của nó sẽ tăng độ liên quan của tất cả các tài liệu văn bản có liên quan theo tỷ lệ ngang nhau. Trong một truy vấn boolean nhiều mệnh đề, một số tài liệu có thể tương ứng với một mệnh đề nhưng không tương ứng với các mệnh đề còn lại, cho phép thừa số Boost phân biệt giữa các truy vấn với nhau. Các truy vấn cũng mặc định có thừa số Boost bằng 1.0.

## 2.4. Ví dụ minh họa

Chúng tôi thực hiện dựng mô hình truy vấn thông tin theo mã nguồn mở của Lucence 7.04 và tiến hành thực hiện đánh index và truy vấn với dữ liệu mà chúng tôi đã thu thập được trên thư viện pháp luật và có những kết quả như sau:

Bảng 2.1 Ví dụ tập văn bản pháp luật dùng để dánh chỉ mục.

|  |  |
| --- | --- |
| DocID | Tên văn bản |
| 1 | QUY ĐỊNH VỀ DỰ BÁO, CẢNH BÁO, TRUYỀN TIN THIÊN TAI VÀ CẤP ĐỘ RỦI RO THIÊN TAI |
| 2 | PHÊ DUYỆT CHIẾN LƯỢC QUỐC GIA PHÒNG, CHỐNG THIÊN TAI ĐẾN NĂM 2030, TẦM NHÌN ĐẾN NĂM 2050 |
| 2 | BAN HÀNH QUY CHẾ HOẠT ĐỘNG ỨNG PHÓ SỰ CỐ TRÀN DẦU |

Sau khi Lucene có được tập văn bản thì tiếp tục phân tích các tập văn bản đó thành token theo DocID và thực hiện đánh index như hình sau.

Table

Description automatically generated

Hình 2.6 Ví dụ mô tả việc đánh chỉ mục ngược dựa vào tập tài liệu văn bản thu thập.

Ở Hình 2.6 thể hiện cách thực hiện là những đối tượng của cùng 1 từ chỉ mục được thực hiện tổng hợp chúng lại với nhau, sau đó thực hiện tách thành Dictionary và Postings và cuối cùng thực hiện lưu lại tần số xuất hiện tài liệu văn bản, cũng chính là độ lớn của mỗi posting list được thêm vào danh mục.

Lucene là mã nguồn mở hỗ trợ mô hình truy xuất thông tin có sử dụng đến lý thuyết tập hợp và lý thuyết của toán tử Boolean như AND, OR. Tác giả thực tiếp tục với ví dụ trên với từ khóa truy vấn “cảnh báo thiên thai” tương ứng với bộ dữ liệu văn bản đã lập chỉ mục, Lucene thực hiện phân tích từ khóa truy vấn và biểu diễn (Một độ phù hợp với từ đó có trong tài liệu, Không phù hợp với từ đó không có trong tài liệu) ma trận từ chỉ mục như sau:

Bảng 2.2 Ma trận thể hiện từ chỉ mục.

|  |  |  |  |
| --- | --- | --- | --- |
| Term | DocID 1 | DocID 2 | DocID 3 |
| Cảnh báo | 1 | 0 | 0 |
| Thiên tai | 1 | 1 | 0 |

Lucene thực hiện phép toán AND ta có: 100 (Cảnh báo) AND 110 (Thiên tai) = 110. Kết quả mà Lucene sẽ trả về là Doc 1 và Doc 2.

# PHƯƠNG PHÁP THỰC HIỆN

## Mô hình thực hiện

#### **Mô hình thực hiện tổng quan**

Trong mô hình thực hiện của bài toán này, chúng tôi thực hiện với hai công đoạn chính là công đoạn xây dựng hệ thống tìm kiếm có cải tiến phần câu truy vấn dựa vào thư viện Lucene và công đoạn xây dựng đánh giá dựa vào tập câu truy vấn liên quan.

Trong công đoạn xây dựng hệ thống truy vấn thông tin về văn bản pháp luật gồm có các thành phần cơ bản sau: Tập các văn bản pháp luật đã thu thập trước đó từ trang thư viện pháp luật, thư viện mã nguồn mở Lucene, tập các văn bản sau khi đánh chỉ mục.

Trong công đoạn đánh giá hệ thống tìm kiếm gồm có: câu truy vấn, tập các văn bản được trả về, tập các văn bản liên quan, phân hệ đánh giá và kết quả thu được.

|  |
| --- |
|  |

Hình 3.1 Quy trình thực hiện bài toán.

#### **Mô hình thực hiện chi tiết**

* Thực hiện thu thập dữ liệu và tiền xử lý dữ liệu văn bản pháp luật
  + Mục đích: Thu thập số lượng lớn văn bản pháp luật từ trang điện tử văn bản pháp luật thông dụng nhất hiện nay (<https://thuvienphapluat.vn/>). Tiền xử lý dữ liệu với tập văn bản đã thu thập về.
  + Đầu vào: Tập dữ liệu chưa được tiền xử lý
  + Đầu ra: Tập dữ liệu văn bản đã được tiền xử lý. (Tập dữ liệu thu thập và tiền xử lý có cấu trúc như ở Mục 4.2.1).
* Thực hiện đánh chỉ mục dữ liệu
  + Mục đích: Đánh chỉ mục các tập tin văn bản dùng để phục vụ cho việc truy vấn thông tin và đánh giá kết quả trả về (Tập văn bản sử dụng đánh chỉ mục đã được tiền xử lý dữ liệu ở Mục 3.2.1).
  + Đầu vào: Tập tài liệu văn bản pháp luật đã thực hiện tiền xử lý dữ liệu.
  + Đầu ra: Tập tài liệu văn bản pháp luật đã được đánh chỉ mục dữ liệu văn bản pháp luật.
  + Mô hình xử lý: Tiến trình đánh chỉ mục là một tiến trình gồm nhiều bước thao tác phức tạp, có thể xem qua mô hình như sau:

|  |
| --- |
|  |

Hình 3.2 Tạo chỉ mục các tập tài liệu văn bản pháp luật đầu vào.

* Thực hiện tìm kiếm
  + Mục đích: Tìm kiếm ra những văn bản liên quan đến nội dung tìm kiếm của câu truy vấn theo ngôn ngữ tự nhiên
  + Đầu vào: Câu truy vấn đã được tiền xử lý trước khi truyền vào core lõi của Lucene thực hiện tìm kiếm. Câu truy vấn được tiền xử lý như Mục 3.2.1 và Mục 3.2.3
  + Đầu ra: Tập các kết quả liên quan có xếp hạng
  + Mô hình xử lý:

|  |
| --- |
|  |

Hình 3.3 Tìm kiếm văn bản dựa trên câu hỏi truy vấn.

* Thực hiện đánh giá kết quả trả về
  + Mục đích: Đánh giá kết quả trả về khi cải tiến mở rộng câu truy vấn so với kết quả tìm kiếm mặc định của Lucene khi chưa thực hiện xử lý mở rộng câu truy vấn
  + Đầu vào: Kết quả tìm kiếm có liên quan và tập văn bản liên quan nội dung truy vấn
  + Kết quả: Kết quả đánh giá đã được so sánh
  + Xử lý: Khi có tập tài liệu kết quả liên quan, tiến hành xử lý đánh giá hệ thống dựa trên tập tài liệu đánh giá thông qua các độ đo (độ chính xác, độ bao phủ ở Mục 4.3). Đồng thời tính độ chính xác ở thứ hạng k và độ chính xác trung bình như sau:
    - Độ chính xác của kết quả ở thứ hạng k (P(k)): Giả sử Retrieved(k) là tập gồm k tài liệu văn bản pháp luật ban đầu mà hệ thống tìm được. Độ chính xác ở hạng k (P(k)) có thể được định nghĩa như sau:
* Độ chính xác trung bình của kết quả: Là một độ đo được kết hợp độ chính xác ở các mức bao phủ và được tính toán bởi công thức sau:

## Quy trình thực hiện

#### **Công đoạn tiền xử lý dữ liệu**

Trong công đoạn tiền xử lý, hệ thống IR tạo ra biểu diễn bên trong của thông tin trong từng tài liệu văn bản pháp luật thông qua quy trình đánh chỉ mục mặc định của Lucene, công đoạn này được tác giả kế thừa xử lý dữ liệu thông qua Lucene. Trước hết, tập tài liệu văn bản pháp luật được tiền xử lý dữ liệu bằng một số phương pháp như phân tích từ vựng, loại bỏ túc từ, lấy từ gốc từ văn bản đơn giản của tài liệu. Kết quả nhận được là tập các từ hay còn được hiểu cách khác là các khái niệm, được coi là khung nhìn logic [8] của bộ tài liệu văn bản pháp luật.

Diagram

Description automatically generated

Hình 3.4 Khung nhìn lôgíc của công đoạn tiền xử lý dữ liệu [8].

Dữ liệu thực nghiệm cũng như để làm bộ dữ liệu có mức độ liên quan dùng để đánh giá kết quả, được sử dụng trong luận văn này được lấy từ nguồn dữ liệu trực tuyến các văn bản pháp luật <https://thuvienphapluat.vn/>, thực hiện lấy về 177.311 văn bản với nhiều chuyên mục và lĩnh vực khác nhau. (Các văn bản lấy về được giới hạn ở dạng text)

Xử lý dữ liệu thô được xem là một bước không thể thiếu trong việc phát triển  
một bộ phân lớp nhằm cải thiện độ chính xác phân lớp. Vì văn bản thô vốn được thu  
thập lưu trữ mà không có cấu trúc nếu giữ nguyên sẽ rất khó khăn trong xử lý. Đặc  
biệt là các thể loại văn bản được thu thập từ nhiều nguồn website sẽ lẫn chứa các HTML code, code đây gọi là nhiễu dữ liệu, dữ liệu thô và cần được xử lý làm sạch dữ liệu.

Về cơ bản cũng dựa vào khung nhìn logic ở trên, chúng tôi thực hiện tiền xử lý dữ liệu văn bản pháp luật kế thừa vào mô hình của Lucene trong luận văn này sẽ bao gồm các bước:

* Làm sạch văn bản.
* Tách từ.
* Phân loại từ.
* Loại bỏ stopwords.

**Làm sạch văn bản:** mục đích bước này là loại bỏ nhiễu trong dữ liệu, đưa dữ liệu này từ dữ liệu nhiễu sang dữ liệu có hệ thống, có tổ chức thông qua quy ước của file json. Đa phần dữ liệu nhiễu là dữ liệu chứa các thẻ HTML và JavaScript. Ví dụ câu “Văn bản <b>Pháp Luật</b>” sau khi làm sạch sẽ thu được câu “Văn bản Pháp Luật”.

**Tách từ:** trong Tiếng Việt dấu cách (space) chỉ mang ý nghĩa phân cách âm tiết với nhau, vì vậy để xử lý trong Tiếng Việt công đoạn tách từ (word segmentation) là một trong những bài toán quan trọng. Ví dụ trong câu “Giảng viên dạy tốt” nếu tách thành 4 từ độc lập thì từ “Giảng” và “viên” sẽ mang ý nghĩa riêng biệt nhau so với khi chúng đứng cùng nhau trong từ “Giảng viên”.

**Loại bỏ hư từ (stopwords):** là loại bỏ những từ xuất hiện nhiều trong ngôn ngữ tự nhiên, ngôn ngữ đời thường, tuy nhiên lại không mang nhiều ý nghĩa mà còn làm cho kết quả truy vấn không được chính xác. Trong Tiếng Việt, hư từ thường là những từ như sau: ở, thì, để, là, này, kia, v.v.

Với những vấn đề ở trên, tôi đã quyết định sử dụng bộ thư viện VnCoreNLP: A Vietnamese Natural Language Processing Toolkit [9], đây là một bộ thư viện mã nguồn mở trong python. Thư viện này đang được dùng rất rộng rãi và có độ hiệu quả cao trong việc tiền xử lý văn bản Tiếng Việt.

#### **Lập chỉ mục**

Bước lập chỉ mục cũng được chúng tôi sử dụng kế thừa từ Lucene để thực hiện đánh index dữ liệu sau khi thực hiện tiền xử lý dữ liệu. Chúng tôi sử dụng các từ như mô tả khung nhìn lôgic của các tài liệu văn bản pháp luật để thực hiện đánh chỉ mục. Như trên chúng tôi cũng đã nêu, cấu trúc chỉ mục phổ biến nhất được sử dụng hiện nay là Inverted Files, trong đó tập tài liệu đã thu thập được biến đổi thành một tập các từ kèm theo một danh sách các từ và các tài liệu văn bản pháp luật mà chúng xuất hiện. Trong một Inverted File, mỗi từ trỏ tới một danh sách tất cả các tài liệu liên quan mà nó xuất hiện trong văn bản đó. Cấu trúc chỉ mục như vậy đóng vai trò rất quan trọng vì nó "cho phép tìm kiếm nhanh trên tập dữ liệu lớn" [8] hoặc một miền dữ liệu xác định trước.

Quy trình này có thể được thực hiện thủ công (đòi hỏi sức người nên rất tốn kém) như ví dụ ở “Chương: Cơ sở lý thuyết ở mục 2.4” hoặc tự động bằng cách tách các term từ văn bản của phần tử thông tin văn bản, sử dụng một thủ tục dựa trên thống kê hoặc ngôn ngữ địa phương.

Một số từ có thể biểu diễn tốt hơn lĩnh vực hoặc chủ đề của tài liệu thu thập. Do đó, mỗi từ có thể được gán một trọng số cho thấy tầm quan trọng của từ trong tài liệu thu thập. Như vậy, cấu trúc đánh chỉ mục bao gồm một tập các từ đã được xử lý, kèm theo một danh sách các tài liệu văn bản chứa các từ đó và trọng số của từ đó được thể hiện lặp lại bao nhiêu lần trong một văn bản pháp luật. Trọng số của một từ trong một tài liệu văn bản pháp luật có thể chỉ đơn giản là số lần xuất hiện của từ đó trong tài liệu. Tần số xuất hiện càng lớn thì tầm quan trọng của từ đó càng lớn. Điều này được gọi là gán trọng số theo tần số từ (term frequency weighting – tf). Số lượng tài liệu mà một từ xuất hiện trong đó cũng có thể được dùng làm yếu tố chính đáng trong việc gán trọng số. Một từ xuất hiện trong càng nhiều văn bản thì khả năng phân biệt của từ đó đối với càng nhiều văn bản càng cao. Điều này gọi là tần số tài liệu đảo ngược (inverse document frequency – idf). Lược đồ gán trọng số tf-idf được sử dụng phổ biến trong các hệ thống truy xuất văn bản nói chung và tài liệu văn bản ở một miền dữ liệu hẹp nói riêng.

#### **Xử lý câu truy vấn**

Đối với bài toán, truy vấn thông tin văn bản ở miền dữ liệu hẹp là văn bản pháp luật mà chúng tôi đang tiến hành thực nghiệm, chúng tôi đã đề xuất xử lý cải tiến ở công đoạn tiền xử lý câu truy vấn, với mục đích là đem lại kết quả tốt nhất ở miền dữ liệu hẹp. Trước khi xử lý cải tiến chúng tôi cũng đã thực hiện kế thừa các nội dung đã đề cập ở trên. Đề tài được lấy ý tưởng từ việc truy vấn thông tin ở nội bộ cơ quan nhà nước với ngôn ngữ đời thường Tiếng Việt, hiện đang cho kết quả không như mong đợi, vấn đề này cũng là nhu cầu cấp bách trong thời buổi hiện nay.

Đặt vấn đề, nhu cầu tìm kiếm thông tin của người dung được xem là một câu truy vấn (request), cùng là đầu vào cho hệ thống IR. Một yêu cầu có thể được viết ở dạng ngôn ngữ tự nhiên, nó là một tập các từ khóa với từ vựng theo ngữ nghĩa của Tiếng Việt, hoặc có thể được phát biểu với các toán tử Boolean mặc định Lucene. Ví dụ: câu truy vấn “*điều kiện trở thành giảng viên*” mặc định đưa vào tập từ khóa để truy vấn là *“điều OR kiện OR trở OR thành OR giảng OR viên”* mà chưa được xử lý phân loại từ cũng như gán các toán tử phù hợp với loại từ.

Ở các bước xử lý ở công đoạn đầu về truy vấn thông tin, hệ thống IR thực hiện các thao tác xử lý đối với truy vấn của người dùng tương tự như đối với các tài liệu ban đầu trong quá trình tiền xử lý dữ liệu văn bản, tuy nhiên để có kết quả mong muốn tốt hơn chúng tôi thực hiện xử lý thêm về việc tiền xử lý câu truy vấn, cũng như gán luật cho từ khóa của truy vấn theo các toán tử Boolean (AND, OR). Chúng tôi thực hiện quy ước cho tập từ khóa truy vấn sau tiền xử lý của người dùng như sau:

* Trường hợp các từ khóa truy vấn đều được phân loại là N, Np (Danh từ) thì gán với toán tử AND cho tất cả các từ khóa.
* Trường hợp các từ khóa truy vấn đều phân loại là V (Động từ) và một số từ loại khác N, Np thì gán với toán tử OR cho tất cả các từ khóa
* Trường hợp các từ khóa truy cấn có cả N, Np, V và các từ loại khác thì gán với toán tử AND cho các từ loại N, Np đồng thời kết hợp gán các từ loại còn lại với toán tử OR

Các thao tác xử lý mở rộng câu truy vấn bằng toán tử kết hợp với phân loại từ là những phương thức cải tiến mở rộng chính mà chúng tôi cải tiến thêm để biểu diễn những nhu cầu truy vấn của người dùng sao cho trả về tập kết quả liên quan gần nhất với người dùng.

#### **Tìm kiếm**

Trong công đoạn tìm kiếm, chúng tôi tiếp tục kế thừa xử lý của Lucene sau khi thực hiện hiện cái tiến mở rộng câu truy vấn,chúng tôi dùng kết quả đã cải tiến để thực hiện truyền xuống cho hệ thống Lucene thực hiện xử lý công đoạn tiếp theo là công đoạn tiềm kiếm. Và chúng tôi thấy, Lucene xử lý mỗi từ thu thập được từ thao tác xử lý dữ liệu đều được dùng để xác định, thông qua tập đánh chỉ mục, và một danh sách các tài liệu pháp luật liên quan mà trong đó từ đó xuất hiện. Nếu có nhiều từ xuất hiện trong câu truy vấn thì công đoạn tìm kiếm sẽ trả về tập hợp của các tài liệu văn bản pháp luật được theo tất cả các từ hoặc một số từ, tùy theo kiểu truy vấn của người dùng. Tóm lại, theo xử lý của Lucene thì tìm kiếm là quá trình đối sánh (matching) các từ trong các tài liệu với các từ trong truy vấn trước khi xếp hạng.

Ta thấy hệ thống IR thực hiện so sánh giữa câu truy vấn với từng biểu diễn của tài liệu liên quan để đánh giá độ liên quan của nó với nhu cầu tìm kiếm thông tin. Kết quả so sánh có thể là liên quan tuyệt đối với câu truy vấn hoặc liên quan một phần nào đó, nhưng do tính không rõ ràng vốn có của quá trình truy vấn thông tin cũng như ngôn ngữ truy vấn nên phù hợp một phần ngày càng được ưa chuộng hơn, do vậy việc so sánh tuyệt đối không được xét trong luận văn. Trong trường hợp so sánh một phần, các tài liệu thường được chuyển tới người dùng theo thứ tự giảm dần của độ liên quan. Mục đích của việc so sánh một phần là để trình bày các tài liệu có liên quan với nhu cầu thông tin ở phần đầu tiên trong danh sách được xếp hạng.

#### **Xếp hạng**

Tương tự cho các công đoạn trên, công đoạn này chúng tôi cũng thực hiện kế thừa việc sếp hạng mặc định của Lucene. Mọi tài liệu văn bản pháp luật thu thập được sẽ được đánh giá theo độ liên quan nhất định của chúng đối với câu truy vấn. Thông thường, đánh giá này phụ thuộc vào một giải thuật xếp hạng, giải thuật này tính toán ra kết quả là một con số thực cho từng tài liệu. Tài liệu ứng với một số giá trị càng lớn thì càng được xem là có khả năng liên quan nhiều hơn. Tiếp theo, các tài liệu văn bản pháp luật được thu thập sẽ được trả về theo thứ tự giảm dần theo kết quả của giải thuật xếp hạng. Nhờ vậy, người dùng có cơ hội để xem xét kỹ hơn các tài liệu liên quan nhất, nằm ở phần trên trong thứ tự sắp xếp, so với các tài liệu không liên quan. Do đó, giải thuật xếp hạng được xem là phần cốt yếu của một hệ thống IR, tuy nhiên nó cũng phụ thuộc phần lớn vào bộ dữ liệu thu thập đã được tiền xử lý cũng như đề xuất tiền xử lý mở rộng câu truy vấn. Và mấu chốt quan trọng nhất vẫn là bộ tài liệu văn bản liên quan tương ứng với từng câu truy vấn, để có được kết quả gần với mong muốn của người dùng nhất.

# THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

## Môi trường thực nghiệm

#### **Môi trường thực nghiệm**

Các thuật toán và mô hình xử lý được phát triển dựa trên ngôn ngữ Java phiên bản 7.4.0 và môi trường Windows 10 Professional 64bit, Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz 1.80 GHz, RAM 12GB và bộ nhớ 1TB.

#### **Công cụ sử dụng**

Trong luận văn này, tác giả xây dựng mô hình tìm kiếm và mô hình thực nghiệm trên IDE là Eclipse phiên bản 2020-12. Chúng tôi thực hiện các công việc như xây dựng core lõi của Apache Lucene phiên bản 7.04. Từ đây, tác giả tìm hiểu và thực hiện thay đổi cấu trúc câu truy vấn cũng như gán từ loại cho câu truy vấn trước khi truyền vào core lõi của Apache Lucene thực hiện truy vấn.

Và trước khi sử dụng Eclipse để chạy thực nghiệm thì tác giả cần một công cụ tải dữ liệu trên các trang web có văn bản pháp luật về để sử dụng làm dữ liệu thực nghiệm cho luận văn. Công cụ đó có tên gọi là Scrapy (Cào dữ liệu phân trang), công cụ này đã hỗ trợ cho việc thu thập văn bản pháp luật, từ dữ liệu thô cho đến dữ liệu theo định dạng có cấu trúc JSON. Chúng tôi sử dụng không chỉ Scrapy mà còn sử dụng một công cụ nữa là BeautifulSoup

BeautifulSoup là một công cụ được phát triển bằng Python, nó được sử dụng để tách dữ liệu ra khỏi các file HTML và XML. Nó hoạt động cùng với trình phân tích cú pháp để cung cấp cách điều hướng, tìm kiếm và sửa đổi cây phân tích cú pháp. Nó thường tiết kiệm khác nhiều thời gian cho công việc bóc tách dữ liệu như này.

Tiếp theo là công cụ giải quyết cho việc tách từ, phân loại từ có tên gọi là VncoreNLP. VnCoreNLP là một bộ công cụ xử lý ngôn ngữ đời thường cho Tiếng Việt, được xây dựng bởi tác giả Nguyễn Quốc Đạt và cộng sự [9], nhằm mục đích tạo điều kiện cho nghiên cứu xử lý ngôn ngữ tự nhiên (NLP) của Việt Nam bằng cách cung cấp các chú thích ngôn ngữ phong phú thông qua các thành phần chính của NLP về phân tách từ, gán nhãn từ loại, nhận dạng thực thể có tên (NER) và phân tích phụ thuộc [9].

## Dữ liệu thực nghiệm

Dữ liệu thực nghiệm cho bài toán đặt ra được thực hiện hai công đoạn:

* Công đoạn thu thập dữ liệu văn bản pháp luật được thực hiện theo lưu đồ xử lý ở mục 3.1.1. Mô hình lưu đồ này chỉ rõ việc thu thập dữ liệu văn bản cũng như việc tiền xử lý dữ liệu sau khi thu thập dự liệu thô từ một trang điện tử về văn bản pháp luật
* Công đoạn thực hiện tổng hợp tập dữ liệu liên quan, công đoạn này thực hiện dựa vào tập danh mục lĩnh vực đã thu thập ở mục 4.2.1, phối hợp với nhóm sinh viên để thực hiện công việc nghiên cứu và tìm hiểu các văn bản pháp luật, đưa ra tập câu truy vấn kèm theo tập kết quả tương ứng với câu truy vấn được thực hiện ở mục 4.2.2.

#### **Công đoạn thu thập dữ liệu**

Dữ liệu thực nghiệm trong luận văn này là được thu thập chọn lọc các văn bản pháp luật trên thư viện pháp luật. Tập dữ liệu thu thập gồm hơn 2.000 văn bản bao gồm nhiều chuyên mục khác nhau và được liệt kê trong Bảng 5.

Bảng 4.1 Danh mục dữ liệu thực nghiệm đã thu thập theo lĩnh vực.

|  |  |
| --- | --- |
| **Danh mục lĩnh vực** | **# văn bản** |
| Bộ máy hành chính | 700 |
| Tài chính nhà nước | 600 |
| Văn hóa – Xã hội | 100 |
| Tài nguyên – Môi trường | 200 |
| Xây dựng – Đô thị | 500 |
| Công nghệ thông tin | 100 |
| Thể thao | 100 |
| Y tế | 300 |
| Lĩnh vực khác | 400 |

#### **Công đoạn thực hiện tổng hợp tập dữ liệu liên quan**

Để có độ chính xác khách quan về kết quả truy vấn, dựa vào tập dữ liệu đã thu thập ở trên. Chúng tôi đã phối hợp với một số sinh viên trường Đại Học Công Nghệ TP. Hồ Chí Minh, đặt ra tiêu chí mỗi thành viên sẽ thực hiện nghiên cứu dữ liệu trong một lĩnh vực khác nhau trong một miền pháp luật. Trình tự thực hiện như sau:

**Bước 1:** Thực hiện nghiên cứu các văn bản trong đã được thu thập về, trong lĩnh vực mình đã chọn để đưa ra câu truy vấn

**Bước 2:** Từ câu truy vấn đó phải tổng hợp được 5 văn bản có kết quả liên quan đến câu truy vấn.

Bảng 4.2 Kết quả tổng hợp của 1 câu truy vấn với 5 kết quả liên quan

|  |  |  |
| --- | --- | --- |
| <ID> | DungNT006 | |
| <Query> | Kế hoạch thực hiện công tác phòng chống ma túy địa phương | |
| <Results> | <DocID> | VB\_445185 |
| <Loại văn bản> | LUẬT |
| <Tên văn bản> | PHÒNG, CHỐNG MA TÚY |
| <DocID> | VB\_469889 |
| <Loại văn bản> | KẾ HOẠCH |
| <Tên văn bản> | PHÒNG, CHỐNG HIV/AIDS GIAI ĐOẠN 2021 – 2025 TRÊN ĐỊA BÀN TỈNH SƠN LA |
| <DocID> | VB\_471262 |
| <Loại văn bản> | KẾ HOẠCH |
| <Tên văn bản> | THỰC HIỆN CÔNG TÁC PHÒNG, CHỐNG MA TÚY NĂM 2021 |
| <DocID> | VB\_469004 |
| <Loại văn bản> | KẾ HOẠCH |
| <Tên văn bản> | THỰC HIỆN CÁC GIẢI PHÁP GIỮ VỮNG XÃ KHÔNG CÓ MA TÚY VÀ KÉO GIẢM XÃ, PHƯỜNG, THỊ TRẤN CÓ MA TÚY NĂM 2021 |
| <DocID> | VB\_470182 |
| <Loại văn bản> | KẾ HOẠCH |
| <Tên văn bản> | PHÒNG, CHỐNG HIV/AIDS TRÊN ĐỊA BÀN THÀNH PHỐ HÀ NỘI NĂM 2021 |

Ở bước này mỗi cá nhân cần thực hiện 100 câu truy vấn khác nhau.

**Bước 3:** Sau khi thực hiện tổng hợp được tập văn bản liên quan với các câu truy vấn và kết quả tương ứng khác nhau. Chúng tôi thực hiện hoán đổi các câu truy vấn của nhau. Ví dụ: Cá nhân A thực hiện được 100 câu truy vấn của lĩnh vực hành chính công, Cá nhân B thực hiện được 100 câu truy vấn của lĩnh vực xã hội hóa. Chúng tôi thực hiện hóa đổi 2 lĩnh vực này của 2 cá nhân A và B, mục đích để lấy ra được kết quả liên quan nhất có thể.

**Bước 4:** Sau khi có tập kết quả liên quan cuối cùng đã hoán đối ở Bước 3, chúng tôi thực hiện giao các kết quả đó để lấy ra phần chung. Phần này được đánh giá là tập dữ liệu liên quan nhất.

Bảng 4.3 Kết quả tổng hợp của một câu truy vấn của cá nhân B.

|  |  |  |
| --- | --- | --- |
| <ID> | HUNGNT005 | |
| <Query> | Kế hoạch thực hiện công tác phòng chống ma túy địa phương | |
| <Results> | <DocID> | VB\_471262 |
| <Loại văn bản> | KẾ HOẠCH |
| <Tên văn bản> | THỰC HIỆN CÔNG TÁC PHÒNG, CHỐNG MA TÚY NĂM 2021 |
| <DocID> | VB\_2451 |
| <Loại văn bản> | QUYẾT\_ĐỊNH |
| <Tên văn bản> | VỀ VIỆC PHÊ DUYỆT "ĐỀ ÁN TỔNG THỂ KIỂM SOÁT MA TUÝ QUA BIÊN GIỚI ĐẾN NĂM 2010" |
| <DocID> | VB\_469004 |
| <Loại văn bản> | KẾ HOẠCH |
| <Tên văn bản> | THỰC HIỆN CÁC GIẢI PHÁP GIỮ VỮNG XÃ KHÔNG CÓ MA TÚY VÀ KÉO GIẢM XÃ, PHƯỜNG, THỊ TRẤN CÓ MA TÚY NĂM 2021 |
| <DocID> | VB\_445185 |
| <Loại văn bản> | LUẬT |
| <Tên văn bản> | PHÒNG, CHỐNG MA TÚY |
| <DocID> | VB\_470182 |
| <Loại văn bản> | KẾ HOẠCH |
| <Tên văn bản> | PHÒNG, CHỐNG HIV/AIDS TRÊN ĐỊA BÀN THÀNH PHỐ HÀ NỘI NĂM 2021 |

Ví dụ: Ở Bảng 6 là kết quả của Cá nhân A và ở Bảng 7 là kết quả của cá nhân B, 2 cá nhân đã hoán đổi lĩnh vực cho nhau để tìm ra văn bản liên quan. Và Bảng 8 là kết quả đã được chúng tôi thực hiện lấy phần chung.

Bảng 4.4 Tập văn bản liên quan dùng để đánh giá độ chính xác

|  |  |  |
| --- | --- | --- |
| <ID> | DungNT006 | |
| <Query> | Kế hoạch thực hiện công tác phòng chống ma túy địa phương | |
| <Results> | <DocID> | VB\_445185 |
| <Loại văn bản> | LUẬT |
| <Tên văn bản> | PHÒNG, CHỐNG MA TÚY |
| <DocID> | VB\_471262 |
| <Loại văn bản> | KẾ HOẠCH |
| <Tên văn bản> | THỰC HIỆN CÔNG TÁC PHÒNG, CHỐNG MA TÚY NĂM 2021 |
| <DocID> | VB\_469004 |
| <Loại văn bản> | KẾ HOẠCH |
| <Tên văn bản> | THỰC HIỆN CÁC GIẢI PHÁP GIỮ VỮNG XÃ KHÔNG CÓ MA TÚY VÀ KÉO GIẢM XÃ, PHƯỜNG, THỊ TRẤN CÓ MA TÚY NĂM 2021 |
| <DocID> | VB\_470182 |
| <Loại văn bản> | KẾ HOẠCH |
| <Tên văn bản> | PHÒNG, CHỐNG HIV/AIDS TRÊN ĐỊA BÀN THÀNH PHỐ HÀ NỘI NĂM 2021 |

## Phương pháp đánh giá kết quả

* **Phép đo độ chính xác và độ bao phủ**

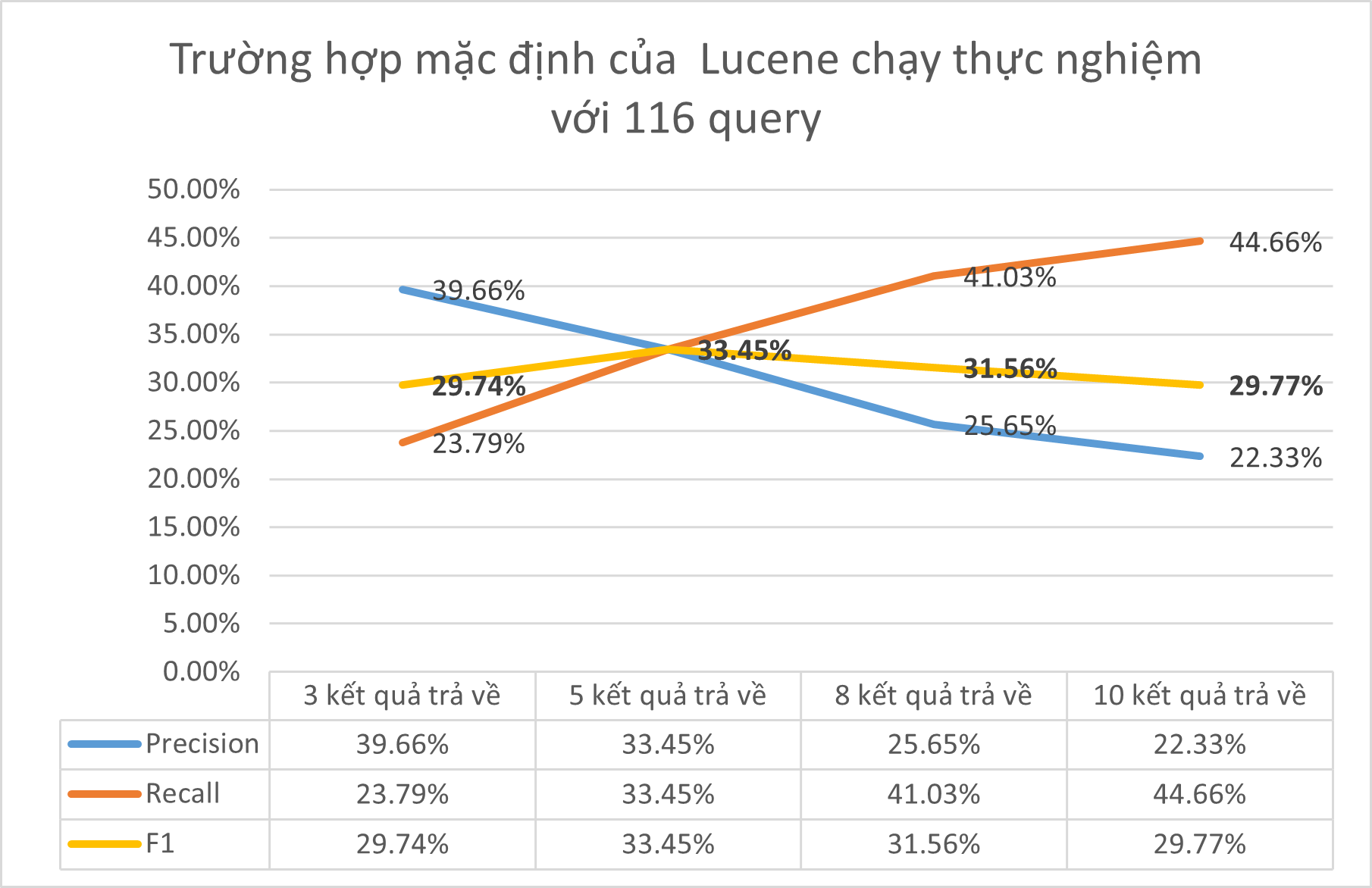
Hiệu quả của một hệ thống IR thể hiện khả năng đáp ứng của kết quả truy xuất đối với một yêu cầu thông tin. Các độ đo hiệu quả truy xuất phổ biến là độ chính xác (precision) và độ bao phủ (recall) [10]. Cả hai độ đo đều dựa trên các đánh giá của người dùng theo quá trình truy xuất. Độ chính xác (P) là tỷ trọng tài liệu văn bản được thu thập có liên quan, nó cho biết khả năng thu thập những tài liệu được xếp hạng trên cùng mà phần lớn là có liên quan. Độ bao phủ (R) là tỷ trọng tài liệu văn bản có liên quan được thu thập, nó cho biết khả năng truy vấn tất cả các tài liệu văn bản có liên quan đến tập hợp. P và R lần lượt được tính theo công thức (8) và (9)

|  |  |
| --- | --- |
|  | (8) |
|  | (9) |

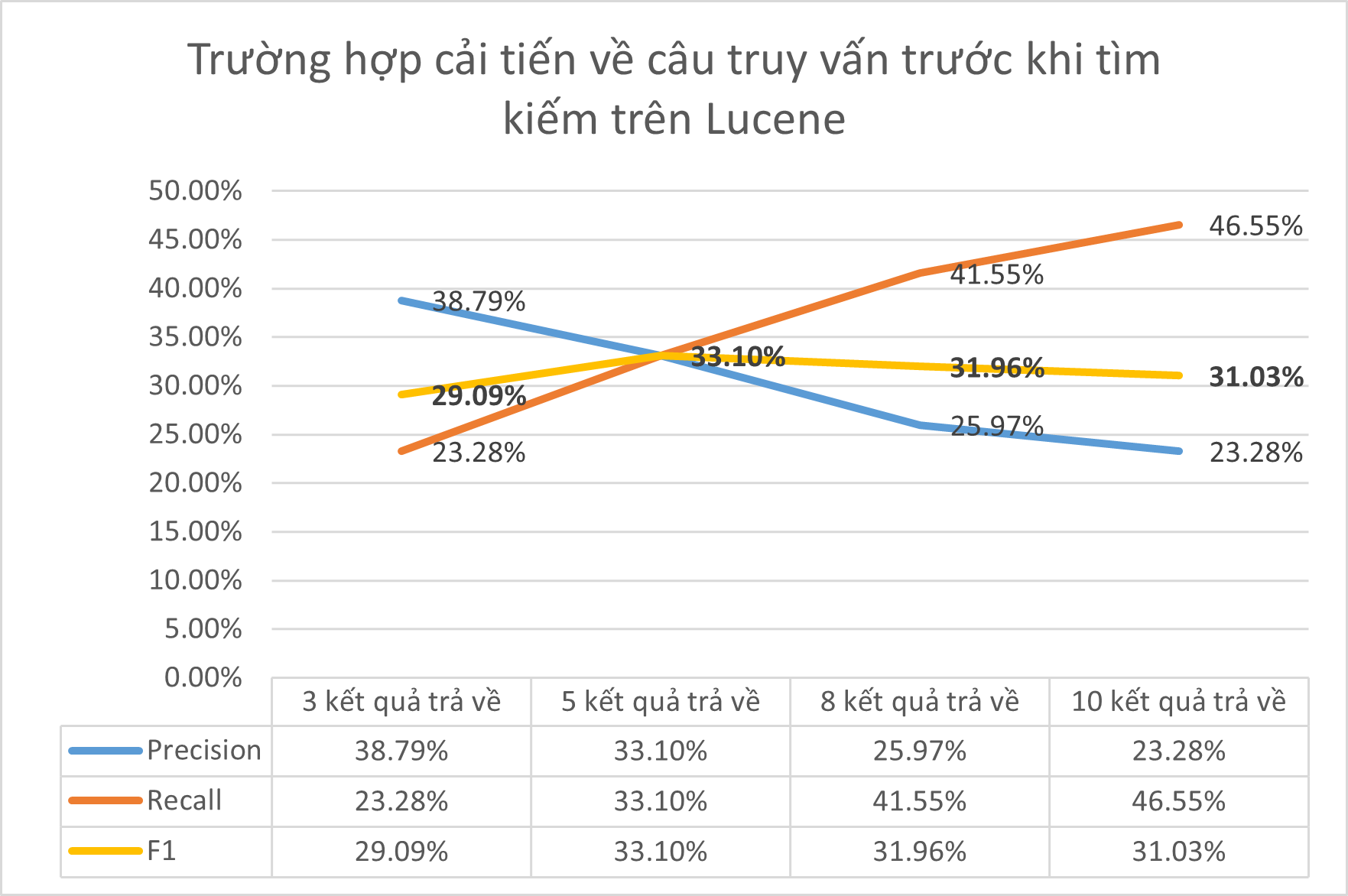
Nhược điểm của hai độ đo này là thực tế thì việc kiểm tra tất cả bộ tài liệu văn bản của tập kết quả. Tuy nhiên, điều này đối lập với thực tiễn sử dụng, trong đó người dùng chỉ xem một phần của tập tài liệu đã thu thập, xếp hạng kết quả theo độ liên quan. Do đó, các giá trị của phép đo độ chính xác và độ bao phủ biến đổi khi người dùng tiến hành kiểm tra các tài liệu được thu thập. Cách thức phổ biến để biểu diễn sự biến đổi đó, nói cách khác là biểu diễn hiệu quả truy xuất là sử dụng phép đo có độ bao phủ, độ chính xác [10].

Để vẽ đồ thị đó, phép đo có độ chính xác được tính dựa vào 11 mức chuẩn của phép đo độ bao phủ là 0%, 10%, 20%, ... 100%. Ví dụ, phép đo độ chính xác của các tài liệu văn bản được thu thập được tính tới khi hệ thống IR trả về 10% số lượng các tài liệu có liên quan. Nếu đạt tới độ bao phủ 10% với tài liệu thứ hai của tập tài liệu được thu thập thì độ chính xác ứng với mức này sẽ là 50%. Đối với mức bao phủ 0%, độ chính xác đạt được thông qua một thủ tục nội suy. Mức bao phủ của một yêu cầu ví dụ có thể khác với 11 mức bao phủ chuẩn, khi đó cũng cần phải sử dụng thủ tục nội suy nhưng cách này ít được dùng để đánh giá hệ thống IR.

Dựa vào dữ liệu văn bản đã thu thập cũng như tập dữ liệu liên quan (tập văn bản có độ chính xác phù hợp đến câu truy vấn), tác giả bắt đầu thực nghiệm nghiệm với các góc độ đánh giá kết quả trả về tương ứng cho 3, 5, 8, 10 kết quả trả về được thể hiện trên đồ thị độ chính xác (P) và độ bao phủ (R) như sau:



Hình 4.1 Kết quả thể hiện độ chính xác (P) và độ bao phủ (R) theo trường hợp mặc định của Lucene.



Hình 4.2 Kết quả thể hiện độ chính xác (P) và độ bao phủ (R) theo trường hợp đã cải tiến câu query.

Các giá trị của thước đo độ chính xác và độ bao phủ, và kết quả của việc tính trung bình nhiều truy vấn, thường được sử dụng để so sánh hiệu năng của các hệ thống IR. Những đồ thị độ có phép đo độ chính xác và độ bao phủ trung bình như vậy mang lại cái nhìn tốt hơn cho hiệu quả truy xuất tổng thể.

* **Độ chính xác trung bình**

Thông thường, trong một hệ thống IR, độ chính xác quan trọng hơn độ bao phủ nếu như người dùng muốn tìm câu trả lời gần đúng nhất cho một câu truy vấn theo ngôn ngữ thông thường chứ không phải tất cả các câu trả lời có thể có. Độ bao phủ có thể quan trọng khi người dùng cần biết tất cả nội dung có liên quan về một chủ đề. Trong các hệ thống IR thông thường, độ bao phủ có thể được tăng lên khi số lượng các phần tử được thu thập tăng lên, trong khi đó độ chính xác lại có khả năng giảm đi. Vì lý do này, các kỹ thuật cải thiện độ bao phủ sẽ tác động tới độ chính xác và ngược lại. Cần phải có sự cân bằng giữa các kỹ thuật đó để điều chỉnh hiệu quả truy xuất.

Thực tế, phép đo của độ chính xác và độ bao phủ không đủ để đánh giá các hệ thống IR. Ví dụ, nếu có hai hệ thống, mỗi hệ thống thu thập được 10 tài liêu, 5 tài liệu có mức độ liên quan tốt và 5 tài liệu văn bản không có mức độ liên quan cao, nhưng cả hai đều có độ chính xác là 0.5, nhưng một hệ thống có 5 tài liệu đầu tiên có liên quan và 5 tài liệu tiếp theo không liên quan thì tốt hơn một hệ thống có 5 tài liệu đầu tiên không liên quan và 5 tài liệu tiếp theo có liên quan (bởi lẽ người dùng sẽ không hài lòng nếu phải kiểm tra những tài liệu không liên quan trước). Do đó, cần có những phép đo kết hợp độ chính xác và độ bao phủ có xét tới thứ tự các tài liệu được thu thập.

Một hướng tiếp cận là sử dụng độ đo R-Precision. Cách tiếp cận này dùng phép đo của độ chính xác thứ R trong bảng xếp hạng các kết quả ứng với một truy vấn, với R là tổng số các tài liệu văn bản có liên quan đã biết. Giá trị thu được có lợi ích để quan sát hoạt động của một chiến lược truy vấn đối với từng truy vấn trong tập kiểm tra.

Để so sánh hai giải thuật, có thể sử dụng kết quả của độ chính xác. Biểu đồ độ chính xác là một biểu đồ R-Precision đối với nhiều truy vấn.

|  |  |
| --- | --- |
|  | (10) |

Trong đó, RPA (i) và RPB (i) lần lượt là các giá trị R-Precision của giải thuật truy xuất A và B đối với tìm kiếm thứ i.

• RPA/B = 0: cả hai giải thuật đều có hiệu quả truy xuất ngang nhau đối với tìm kiêm thứ i.

• RPA/B > 0: giải thuật A có hiệu quả tìm kiếm tốt hơn đối với tìm kiếm thứ i.

• RPA/B <0: giải thuật B có hiệu quả tìm kiếm tốt hơn đối với tìm kiếm thứ i.

Độ đo được dùng phổ biến nhất là độ chính xác trung bình (mean average precision – **MAP**). Độ đo MAP được tính bằng độ chính xác tại mỗi điểm trong bảng xếp hạng mà ở đó tìm thấy một tài liệu văn bản có liên quan, sau đó lấy trung bình của các giá trị đó (và cuối cùng là trung bình đối với tất cả các truy vấn).

* **Độ đo F1 (F-measure)**

Độ đo F (*F-measure*) kết hợp với phép đo độ chính xác và độ bao phủ để lấy trung bình điều hòa của chúng.

|  |  |
| --- | --- |
|  | (11) |

Trong đó r(j) là giá trị của độ bao phủ và P(j) là độ chính xác của tài liệu thứ i trong tập hợp. Các giá trị tỷ lệ của F(j) nằm trong vùng [0,1], bằng 0 nếu không có tài liệu văn bản có liên quan nào được thu thập và bằng 1 nếu tất cả các tài liệu văn bản được thu thập đều có liên quan. Hơn nữa, F(j) nhận giá trị cao khi và chỉ khi cả độ chính xác và độ bao phủ đều cao, điều này biểu thị sự cân bằng tốt giữa độ chính xác và độ bao phủ.

Đối với các tập văn bản lớn trên Web, rất khó để tìm tổng số các tài liệu văn bản có liên quan tuyệt đối. Đối với trường hợp này, phải đưa ra giải pháp là lấy mẫu trên tập hợp và thực hiện đánh giá độ liên quan dựa trên các tài liệu mẫu (tập tài liệu có liên quan).

Hệ thống IR khác nhau khi dùng một tập hợp dùng cùng truy vấn, sau đó kết hợp tất cả các tài liệu văn bản có liên quan và thực hiện đánh giá độ liên quan dựa trên tập này.

## Kết quả thực nghiệm và đánh giá

Đánh giá được thực hiện trên 61 câu query và 116 câu query cho 2 tập kết quả trả về gồm 5 kết quả trả về 10 kết quả trả tương ứng với 61 và 116 câu query (tạm gọi là tập dữ liệu liên quan).

Chỉ số đánh giá được chạy với tập dữ liệu liên quan trên và tương ứng với 2 trường hợp như sau:

- Trường hợp 1: Thực nghiệm với phần core mặc định của Lucene 7.04

- Trường hợp 2: Tiền xử lý dữ liệu văn bản trước khi đánh index (Thực hiện tách từ Tiếng Việt; Gán nhãn phân loại từ; Thực hiện xóa túc từ; Thực hiện loại bỏ dấu Tiếng Việt); Tiền xử lý câu truy vấn (Tư tượng cho các xử lý ở dữ liệu trước khi index và xử lý thêm phần sắp xếp từ loại theo toán tử như Mục 3.2.3); Thực hiện cải tiến mở rộng câu truy vấn.

Chúng tôi có bảng kết quả đánh giá như sau:

Bảng 4.5 Kết quả thực nghiệm so sánh 2 trường hợp với 5 kết quả trả về và 10 kết quả trả về.

|  |  |  |  |
| --- | --- | --- | --- |
| Tập 116 queries | | Lấy 5 kết quả trả về | Lấy 10 kết quả trả về |
| Trường hợp mặc định của Lucene | Precision (P) | 33.45 | 22.32 |
| Recall (R) | 33.45 | 44.65 |
| F1 | **33.45** | **29.77** |
| Trường hợp tiền xử lý câu truy vấn | Precision (P) | 33.10 | 23.27 |
| Recall (R) | 33.10 | 46.55 |
| F1 | 33.10 | **31.03** |

Chúng tôi sử dụng bộ dữ liệu thực nghiệm gồm 2000 văn bản pháp luật Tiếng Việt (trong số 177.311 văn bản với nhiều chuyên mục khác nhau) đã được xử lý trước khi đánh index dựa vào mã nguồn mở Lucene. Các tài liệu ở định dạng TXT, HTML, DOC, PDF. Trước khi tiến hành thực nghiệm, chúng tôi dựa vào 2000 văn bản đã đưa ra 116 câu query, và tương ứng một câu query sẽ có 5 kết quả trả về tương ứng (gọi là tập dữ liệu văn bản liên quan).

Chúng tôi tiến hành thực nghiệm với tập dữ liệu 2000 văn bản đã tiền xử lý và 116 câu query ứng với tập kết quả liên quan trong 2 môi trường khác nhau để có sự đánh giá khách quan (Môi trường mặc định của mã nguồn mở Lucene và Môi trường có tiền xử lý câu truy vấn trước khi áp dụng mã nguồn mở Lucene) đã mang lại kết quả như Bảng 4.5.

Qua bảng đánh giá (Bảng 4.5), ta có thể so sánh trường hợp mặc định và trường hợp đã tiền xử lý câu truy vấn thì kết quả trả về không cao đối với trường hợp đã xử lý tương ứng với tập 5 kết quả trả về. Còn đối với tập 10 kết quả trả về thì trường hợp đã tiền xử lý câu truy vấn có kết quả đánh giá cao hơn trường hợp mặc định của Lucene khi chưa xử lý và có độ chính xác F1 tăng hơn so với mặc định của Lucene là **1.26%**.

Với kết quả thực nghiệm trên miền dữ liệu văn bản pháp luật có kết quả tương đối chính xác cao và hiệu quả. Như vậy chúng tôi đã kế thừa mã nguồn mở Lucene và thực hiện cải tiến câu truy vấn đã đáp ứng được mục tiêu đặt ra của đề tài. Qua kết quả thực nghiệm này, chúng tôi có thể áp dụng mô hình cải tiến này cho việc tìm kiếm tài liệu văn bản pháp luật cho các cơ quan nhà nước hay một cơ quan được tổ chức khép kín.

# CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Trong bài toán này, chúng tôi đã thể hiện các bước cơ bản cần thực hiện để xây dựng hệ thống truy vấn thông tin; nghiên cứu tìm hiểu Lucene và so sánh các kỹ thuật đánh chỉ mục và các mô hình truy vấn thông tin; trình bày cơ chế đánh chỉ mục Inverted Files và mô hình truy vấn thông tin của thư viện mã nguồn mở Lucene; kế thừa mô hình truy vấn thông tin của Lucene và cải tiến thêm về mở rộng câu truy vấn trước khi thực hiện tìm kiếm. Cụ thể:

Chúng tôi đã tiến hành các bước của quá trình truy vấn thông tin văn bản pháp luật, bao gồm hai công đoạn chính là tiền xử lý tài liệu và thu thập thông tin văn bản pháp luật. Chúng tôi cũng xây dựng các phương pháp đánh giá hiệu quả truy vấn theo hai hướng chính là tiếp cận hướng mô hình và tiếp cận đánh giá lấy người sử dụng làm yếu tố quyết định.

Chúng tôi đã cải tiến câu query trước khi tìm kiếm trên sự kế thừa một phần của mã nguồn mở Lucene để kiểm nghiệm yếu tố đúng đắn của dữ liệu và đánh giá kết quả trả về có thật sự tốt hơn trong một miền dữ liệu hẹp. Chúng tôi đã chạy thực nghiệm nhiều lần với nhiều truy vấn liên quan đến pháp luật khác nhau. Đồng thời chúng tôi chạy cho hệ thống tìm kiếm mặc định được sử dụng mã nguồn mở Lucene để thực nghiệm với cùng các truy vấn trong cùng điều kiện, cùng ngôn ngữ liên quan. Kết quả thực nghiệm cho thấy sau khi cải tiến về câu query trước khi sử dụng Lucene để thực hiện tìm kiếm có kết quả thật sự mong muốn hơn và có độ chính xác F1 tăng hơn so với mặc định là 1.26%.

Luận văn đã trình bày cơ sở lý thuyết, cũng như tổng quan về Lucene và nguyên lý vận hành của một hệ thống truy vấn thông tin văn bản, trình bày một cách khái quát về các tính năng chính và hoạt động của mã nguồn mở Lucene. Và luận văn đã kế thừa và đóng góp một phần vào việc cải tiến về câu truy vấn trước khi thực hiện truy vấn thành công trong công việc truy vấn văn bản pháp luật theo ngôn ngữ đời thường trên hệ thống quản lý văn bản pháp luật với ngôn ngữ Tiếng Việt.

Bên cạnh những kết quả đạt được đáp ứng ở một miền dữ liệu văn bản nhất định thì đề tài còn có những mặt hạn chế như: Phần thực nghiệm bài toán hiện tại chỉ dừng lại ở phạm vi nhỏ trong miền pháp luật với số lượng văn bản pháp luật còn hạn chế. Với miền dữ liệu khác thì cần phải đánh giá lại về tập dữ liệu liên quan cho việc truy vấn, về hạn chế này cho thấy được để trả về tập kết quả mong muốn cần phụ thuộc vào nhiều miền dữ liệu, nhưng quan trọng nhất vẫn là tập dữ liệu liên quan tạm gọi là tập dữ liệu vàng.

## Hướng phát triển

Hiện tại, luận văn mới xử lý cải tiến nhỏ ở việc cải tiến câu truy vấn trước khi truyền câu query vào hệ thống Lucene và chúng tôi đã có một số kết quả nhất định về việc tập kết quả trả về nhưng chưa phải là tuyệt đối nhất. Bởi vậy, chúng tôi đã đưa ra hướng phát triển của luận văn sẽ là nghiên cứu kỹ thuật mở rộng câu truy vấn bằng các hình thức như: mở rộng về xử lý từ đồng nghĩa của Tiếng Việt, mở rộng về xử lý phân loại từ với Tiếng Việt không dấu. Vì Tiếng Việt có những đặc điểm riêng về cấu tạo của từ, phân loại từ, cấu trúc ngữ pháp theo ngôn ngữ đời thường. Vì vậy cần có cải tiến thêm trong việc xử lý từ vựng, cũng như các đề xuất nói ở trên.

Luận văn này là một bước đầu tiên để chúng tôi làm quen với việc nghiên cứu về bài toán truy vấn thông tin văn bản pháp luật. Chúng tôi mong muốn được cộng tác và làm việc với những người quan tâm, yêu thích về lĩnh vực truy vấn thông tin này để cùng nghiên cứu và giải quyết bài toán truy vấn thống tin, đặc biệt là bài toán tìm kiếm tài liệu văn bản ở ngôn ngữ đời thường nhất và ở một miền dữ liệu tổng quát.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | M. Saravanan, B. Ravindran and S. Raman, "Improving legal information retrieval using an ontological framework," *Artif Intell Law,* p. 17:101–124, 2009. |
| [2] | P. K. Bhatia, T. Mathur and T. Gupta, "Survey Paper on Information Retrieval Algorithms and Personalized Information Retrieval Concept," *International Journal of Computer Applications,* vol. 66– No.6, 2013. |
| [3] | R. Nanda, A. K. John, L. D. Caro, G. Boella and L. Robaldo, "Legal Information Retrieval Using Topic Clustering and Neural Networks," *EPiC Series in Computing,* vol. 47, pp. 68-78, 2017. |
| [4] | D. S. Carvalho, N. T. Minh, T. X. Chien and N. L. Minh, "Lexical-Morphological Modeling for Legal Text Analysis," in *Springer International Publishing AG 2017*, 2017. |
| [5] | K. Sugathadasa, B. Ayesha, N. d. Silva, A. S. Perera, V. Jayawardana, D. Lakmal and M. Perera, "Legal Document Retrieval Using Document Vector Embeddings and Deep Learning," *Springer Nature Switzerland AG 2019,* pp. 160-175, 2019. |
| [6] | A. Kanapala, S. Jannu and R. Pamula, "Passage-Based Text Summarization for Legal Information Retrieval," *Arabian Journal for Science and Engineering,* no. 44, p. 9159–9169, 2019. |
| [7] | M. MCCANDLESS, E. HATCHER and O. GOSPODNETIĆ, "Lucene in Action," Printed in the United States of America, Manning Publications Co, 2010. |
| [8] | R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, New York: Addison Wesley, 1999. |
| [9] | V. Thanh, N. Q. Dat, N. Q. Dai, M. Dras and M. Johnson, "VnCoreNLP: A Vietnamese Natural Language Processing Toolkit," in *Proceedings of NAACL-HLT 2018*, New Orleans, Louisiana, 2018. |
| [10] | Z. Qiu, "Hyperstructure-Based Search Methods for the World Wide Web," *Technischen Universität Darmstadt,* 2004. |
| [11] | D. Inkpen and A. Professor, "Information Retrieval on the Internet," vol. III (Part 3), no. 213, 2007. |
| [12] | K. Sugathadasa, B. Ayesha, N. d. Silva, A. S. Perera, V. Jayawardana, D. Lakmal and M. Perera, "Legal Document Retrieval Using Document Vector Embeddings and Deep Learning," *Springer Nature Switzerland AG,* p. 160–175, 2019. |
| [13] | M. v. Opijnen và C. Santos, “On the concept of relevance in legal information retrieval,” *Artif Intell Law,* p. 25:65–87, 2017. |
| [14] | R. Nanda, A. K. John, L. D. Caro, G. Boella and L. Robaldo, "Legal Information Retrieval Using Topic Clustering and Neural Networks," *EPiC Series in Computing,* vol. 47, pp. 68-78, 2017. |
| [15] | D. S. Carvalho, T. D. Vu, T. V. Khanh and N. L. Minh, "Improving Legal Information Retrieval by Distributional Composition with Term Order Probabilities," *EPiC Series in Computing,* vol. 47, pp. 43-56, 2017. |