ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

~~~~~~*~~~~~~

ĐỒ ÁN MÔN HỌC
TRUY XUẤT THÔNG TIN

MÔ HÌNH TRUY VẤN TF-IDF VÀ LATENT SEMANTIC ANALYSIS CHO CRANFIELD VÀ NFCORPUS

Giảng viên hướng dẫn	:	**Thầy Nguyễn Trọng Chỉnh**
Lớp	:	**CS419.M11.KHCL**
Sinh viên thực hiện	:	**Trần Việt Hoàng - 18520785**
		Bạch Văn Hiếu - 18520730
		Trần Minh Tiến - 18521492

TP.HCM – 2022
~~~~~~

Lời mở đầu

Trước tiên tập thể nhóm xin cảm ơn đến thầy Nguyễn Trọng Chỉnh, giảng viên phụ trách bộ môn CS419.M11.KHCL - Truy Xuất Thông Tin, đã góp ý và hỗ trợ nhóm chỉnh sửa trong quá trình hoàn thành đồ án môn học. Các kiến thức được thầy dạy trên lớp đã giúp nhóm linh hoạt hơn khi tích luỹ kiến thức xử lý ảnh trong suốt thời gian học tập cũng như tìm hiểu đồ án.

Mặc dù nhóm đã cố gắng hoàn thiện đồ án một cách tốt nhất, nhưng do kiến thức còn hạn hẹp và chưa hoàn hảo nên không tránh được những thiếu sót trong quá trình thực hiện. Kính mong sự góp ý và giúp đỡ của thầy để nhóm có thể hoàn thiện thêm đồ án hơn nữa trong tương lai.

Một lần nữa nhóm xin chân thành cảm ơn thầy!

Bảng phân công công việc và tỷ lệ hoàn thành

Công Việc	Thành viên		
	Tiến	Hoàng	Hiếu
Soạn slide	**33%**	**33%**	**33%**
Viết báo cáo	**33%**	**33%**	**33%**
Thuyết trình	**33%**	**33%**	**33%**
Tìm hiểu sâu lý thuyết về hai mô hình và các độ đo đánh giá	**-**	**50%**	**50%**
Tham khảo code và cài đặt thực nghiệm mô hình truy xuất thông tin	**100%**	**-**	**-**

Mục lục

Chương 1. Giới thiệu về bài toán Truy Xuất Thông Tin

1.1. Khái niệm về bài toán truy xuất thông tin

Truy xuất thông tin (information retrieval) trong khoa học máy tính và khoa học thông tin là quá trình thu thập các tài nguyên của hệ thống thông tin có liên quan đến thông tin mà ta có nhu cầu từ tập hợp các tài nguyên đó. Các thông tin thường ở dạng dữ liệu phi cấu trúc (ví dụ như âm thanh, hình ảnh, văn bản) hoặc bán cấu trúc (text theo định dạng XML).

Quá trình truy xuất thông tin bắt đầu khi người dùng nhập một truy vấn vào hệ thống. Truy vấn là những lệnh biểu thị nhu cầu thông tin, chúng có thể là một đoạn văn bản truy vấn, một ảnh hoặc một đoạn câu thoại (ví dụ công cụ tìm kiếm của google cho phép người dùng nhập một đoạn văn bản (google search), một ảnh (google images) hoặc một câu thoại (google voice)). Và các đối tượng đích được truy vấn hướng đến chính là các đơn vị thông tin trong bộ sưu tập nội dung hoặc cơ sở dữ liệu. Mô hình truy xuất thông tin không xác định duy nhất một đối tượng. Thay vào đó nó xác định một số đối tượng có thể phù hợp với truy vấn bằng cách tính toán và xếp hạng mức độ liên quan giữa truy vấn và đối tượng đích. Sự xếp hạng kết quả liên quan này chính là điểm khác biệt giữa truy xuất thông tin và tìm kiếm tài liệu thông thường.

Tùy thuộc vào ứng dụng mà bài toán truy xuất thông tin có truy vấn và đối tượng đích khác nhau. Chẳng hạn như bài toán truy xuất văn bản yêu cầu câu truy vấn là một chuỗi kí tự, đối tượng đích là các tài liệu liên quan. Dưới đây là bảng quy định đầu vào (input) và đầu ra (output) của bài toán truy xuất thông tin dạng văn bản:

INPUT	**OUTPUT**
• Một câu truy vấn (**query**) của người dùng. • Một bộ ngữ liệu (**corpus**).	• Một tập xếp hạng (**rank list**) các thông tin được cho là có liên quan (**relevant**) tới câu truy vấn (**query**).

Thông thường, bản thân tài liệu không được lưu trữ trực tiếp trong hệ thống truy xuất mà được đại diện bởi một dạng thức khác, chẳng hạn như vector đặc trưng tương ứng với tài liệu đó. Hầu hết các hệ thống truy xuất thông tin sẽ sử dụng dạng thức này để tính điểm và xếp hạng mức độ liên quan của các tài liệu với câu truy vấn. Những kết quả được xếp hạng hàng đầu được hiển thị cho người dùng đầu tiên.

1.2. Một số ứng dụng của truy xuất thông tin dạng văn bản

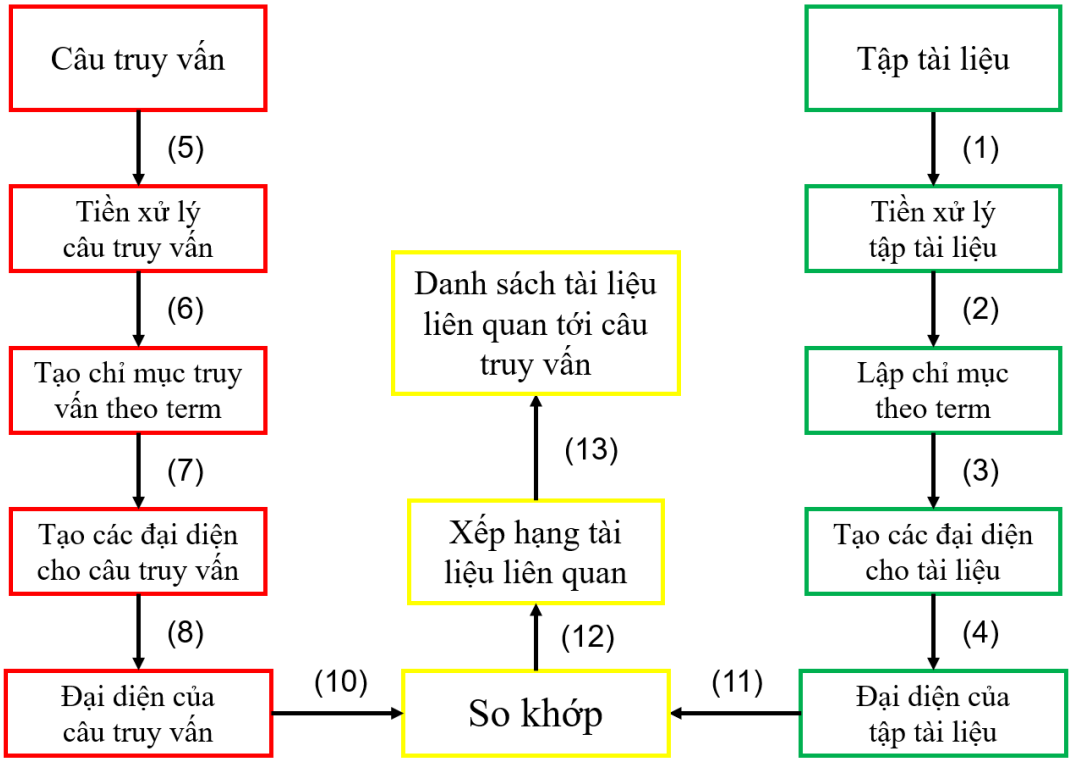
Một trong những ứng dụng phổ biến nhất của truy xuất thông tin là tìm kiếm trên web. Các công cụ tìm kiếm như google, bing, yahoo!... có thể tìm kiếm hàng tỷ tài liệu trên web để trả lời một truy vấn, chẳng hạn như "TP. Hồ Chí Minh có bao nhiêu quận " hoặc "lịch nghỉ Tết nguyên đán 2022".

Một ứng dụng truy xuất thông tin khác là tìm kiếm thông qua các diễn đàn thảo luận trực tuyến hoặc kho lưu trữ Hỏi & Đáp. Ở đây truy xuất thông tin được ứng dụng cùng với phân loại văn bản để đưa ra các chủ đề, bài đăng hoặc câu trả lời liên quan đến truy vấn của người dùng.

Trong bất kỳ hệ thống doanh nghiệp nào, người ta đều cần truy xuất các tài liệu bằng sáng chế, tài liệu nghiên cứu, các ấn phẩm có bản quyền hoặc các điều khoản pháp lý dựa trên một từ khóa hoặc cụm từ.

Trong thư viện, chúng ta có thể muốn lấy những cuốn sách có các từ cụ thể trong tiêu đề của nó, hoặc theo tên tác giả hay theo thể loại.

1.3. Mô hình truy xuất thông tin bằng văn bản



Hình 1: mô hình truy xuất thông tin bằng văn bản

Ở đầu vào là tập tài liệu, đầu tiên chúng ta tiền xử lý và xây dựng từ điển cho tập tài liệu (1). Tiếp theo chúng ta lập chỉ mục cho tài liệu bằng các từ trong tài liệu (2), từ đó chuyển

hóa từng tài liệu trong tập tài liệu sang dạng biểu diễn vector (3) với số chiều của vector có thể bằng số từ trong từ điển. Vì tập tài liệu ở dạng biểu diễn vector (4) được sử dụng để so khớp với câu truy vấn, cho nên mô hình truy xuất thông tin sẽ sử dụng dạng thức này để tính điểm và xếp hạng mức độ liên quan.

Ở đầu vào là câu truy vấn, đầu tiên chúng ta tiền xử lý câu truy vấn (5) và chuyển hóa câu truy vấn sang dạng biểu diễn vector (6, 7, 8) bằng mô hình truy xuất thông tin. Tiếp theo chúng ta tính độ tương đồng của vector truy vấn với từng vector tài liệu (12), xếp hạng các tài liệu có độ tương đồng giảm dần (13). Cuối cùng chọn các tài liệu có mức độ liên quan hàng đầu để hiển thị đến người dùng (14).

Chương 2. Mục tiêu của đồ án

- Hiểu một cách bao quát các khái niệm , cũng như hiểu rõ các vấn đề trong một bái toán truy xuất thông tin

- Áp dụng các kiến thức liên quan đã được học nhằm tối ưu cũng như đề xuất , cải tiến phương pháp phân tích văn bản dựa trên thực nghiệm qua thực tế cài đặt cũng như lý thuyết

- Cài đặt thử nghiệm các mô hình truy xuất thông tin cơ bản , từ đó nắm được quy trình chính của một bài toán truy xuất thông tin , cách xử lý băn bản đầu vào , đầu ra, cách đánh giá , cũng như nhận xét để trả về kết quả truy xuất hợp lý

- Từ lý thuyết và cài đặt thực nghiệm rút ra được kết quả trực quan, từ đó đánh giá và so sánh các phương pháp , mô hình truy xuất thông tin khác nhau

Chương 3. Tiền xử lý ngữ liệu

3.1. Tiền xử lý

Dữ liệu văn bản là một ví dụ điển hình của dữ liệu chuỗi. Một bài báo hay một đoạn văn có thể được coi là một chuỗi các từ, hoặc một chuỗi các ký tự. Dữ liệu dạng văn bản được xếp loại là dữ liệu phi cấu trúc hoặc bán cấu trúc, có thể mang nhiều yếu tố gây nhiễu khiến cho các mô hình truy xuất thông tin khó có thể làm việc trực tiếp. Vì vậy bước đầu tiên và không thể thiếu trong việc xử lý dữ liệu dạng văn bản đó chính là tiền xử lý ngữ liệu.

Sau đây là một số phương pháp phổ biến và hiệu quả để xây dựng mô hình truy xuất thông tin cũng như mô hình học máy từ dữ liệu dạng văn bản, ngôn ngữ tiếng Anh:

3.1.1. Loại bỏ kí tự đặc biệt

Văn bản thô được thu thập từ nhiều nguồn tài nguyên có thể chứa các thẻ tag của HTML và JS, cũng có thể là các ký tự không cần thiết như: !, @, #, $, %, ^, &, ",… hoặc các thẻ trích dẫn,… Ví dụ:

Input: The Lô River (sông Lô) is a major river flowing through 3 provinces of Vietnam [1], [2], [3].

Output: The Lô River sông Lô is a major river flowing through 3 provinces of Vietnam

3.1.2. Tách từ và chuyển đổi chữ cái in hoa

Tách từ (hay tokenization) là một trong những bước quan trọng nhất trong quá trình tiền xử lý văn bản. Nhìn chung, tokenization là quá trình tách một cụm từ, câu, đoạn văn, một hoặc nhiều tài liệu văn bản thành các đơn vị nhỏ hơn, mỗi đơn vị nhỏ hơn này được gọi là token. Có ba loại kỹ thuật tách từ phổ biến, đó là:

1. Tách từ dựa trên từ: tách chuỗi ra thành các từ, xác định bởi khoảng trắng.
2. Tách từ thành các ký tự: tách chuỗi ra thành các ký tự riêng lẻ, không phân biệt theo từ.
3. Tách từ dựa trên từ phụ: chia chuỗi thành các từ, chia các từ thành các từ khóa phụ, chẳng hạn như từ " unspecialized" sẽ được chia thành hai từ khóa phụ là: "un", "special", "ized".

Ở đây chúng tôi tách chuỗi văn bản ra thành các từ bằng kỹ thuật tách từ dựa trên từ (1) và chuyển đổi tất cả chữ cái in hoa thành chữ cái in thường (lowercase) và loại bỏ dấu của từ (de-accent). Sau đây là ví dụ về kỹ thuật tách từ:

Input: The Lô River sông Lô is a major river flowing through 3 provinces of Vietnam

Output: ["the", "lo", "river", "song", "lo", "is", "a", "major", "river", "flowing", "through", "3", "provinces", "of", "vietnam"].

3.1.3. Chuẩn hóa từ

Trong tiếng anh, một từ gốc thường được gắn thêm các tiền tố và hậu tố vào để đúng với ngữ pháp dùng trong ngữ cảnh nào đó, ví dụ từ "look" biến đổi thành "looked" hoặc "looks". Có hai kỹ thuật để đưa các từ đã bị biến đổi trở về nguyên mẫu đó là Stemming và Lemmatization.

1. Stemming: lược bỏ các ký tự của từ mà nó cho rằng đó là biến thể của từ. Ví dụ như ngắt bỏ kí tự "s" của từ "looks" thành từ nguyên mẫu là từ "look".

2. Lemmatization: cách làm của lemmatization thì thông minh hơn, nó có thể phát hiện được các biến thể bất quy tắc của từ hoặc những dạng thức đặc biệt của từ. Ví dụ như nó có thể biến đổi từ "goes", "went" thành từ nguyên mẫu là từ "go".

Chúng tôi sử dụng kỹ thuật Stemming thể chuẩn hóa từ vì kỹ thuật này được sử dụng rộng rãi, có tốc độ xử lý nhanh và trên thực tế thì nó không thua kém nhiều so với Lemmatization. Sau đây là ví dụ về kỹ thuật chuẩn hóa từ:

Input: ["the", "lo", "river", "song", "lo", "is", "a", "major", "river", "flowing", "through", "3", "provinces", "of", "vietnam"].

Output: ["the", "lo", "river", "song", "lo", "is", "a", "major", "river", "flow", "through", "3", "province", "of", "vietnam"].

3.1.4. Lược bỏ stopwords

Stopwords là các từ hầu như không thể hiện ý nghĩa khi xây dựng mô hình đặc trưng cho các văn bản. Đây thường là những từ giới từ, trợ từ có tần suất xuất hiện tương đối cao trong một văn bản thông thường, ví dụ như: "a", "an", "the".... Hoặc những từ phổ biến trong tập ngữ liệu chuyên ngành. Đối với trường hợp đầu tiên thì ta có thể sử dụng bộ stopwords được xây dựng sẵn cho tiếng Anh của các thư viện như nltk,... Đối với trường hợp thứ hai thì bên cạnh việc sử dụng bộ stopwords cho văn bản thông thường, ta phải tự tìm hiểu bộ stopword đặc thù cho ngữ liệu chuyên ngành đó. Ví dụ về việc lược bỏ stopwords:

Input: ["the", "lo", "river", "song", "lo", "is", "a", "major", "river", "flow", "through", "3", "province", "of", "vietnam"].

Output: ["lo", "river", "song", "lo", "major", "river", "flow", "3", "province", "vietnam"].

3.2. Xây dựng tập từ điển và danh sách posting

Tập từ điển (vocabulary) là tập hợp tất cả các từ duy nhất trong tập ngữ liệu đã được tiền xử lý, số lượng tài liệu chứa từ vựng và tổng số lần xuất hiện của từ vựng đó trong toàn bộ ngữ liệu. Danh sách posting (posting list) là danh sách chứa chỉ số tài liệu và số lần xuất hiện của một từ từ vựng trong tài liệu đó.

Mục đích của việc xây dựng tập từ điển và danh sách posting là để phục vụ tính toán cho các mô hình đại số. Ở chương tiếp theo chúng ta sẽ tìm hiểu về cách tính giá trị của sự xuất hiện của một từ trong tài liệu bằng các mô hình đại số như: Vector-Space model và Latent Semantic Analysis model.

Chương 4. Vector-Space model

4.1. Phần lý thuyết

4.1.1. Khái niệm về Vector-Space model

Vector space model (hay mô hình không gian vector) là một mô hình đại số (algebraic model) thể hiện thông tin của tài liệu (document) như một vector, các phần tử của vector này thể hiện mức độ quan trọng của một từ (term) và cả sự xuất hiện hay không xuất hiện của nó trong một tài liệu.

Mô hình này biểu diễn tài liệu như những vector trong không gian Euclid đa chiều, mỗi chiều tương ứng với một từ trong tập từ điển (vocabulary). Trọng số của phần tử thứ *i* trong vector tài liệu cho biết tầm quan trọng của phần tử thứ *I* trong tài liệu đó. Sự tương đồng của hai tài liệu thường được định nghĩa là khoảng cách cosine giữa những vector trong không gian.

Xét một tập các tài liệu $D = \{d_1, d_2, d_3, \ldots, d_N\}$ và các câu truy vấn lần lượt được biểu diễn thành dạng vector $d_i, q \in \mathbb{R}^v$ như sau:

$$d_i = \{w_{1,i}, w_{2,i}, w_{3,i}, \ldots, w_{v,i}\}$$

$$q = \{w_{1,q}, w_{2,q}, w_{3,q}, \ldots, w_{v,q}\}$$

Với d_i là biểu diễn của tài liệu thứ *i* và $w_{1,i}$ là trọng số của từ w_1 xuất hiện trong tài liệu d_i. Truy vấn q là một truy vấn để tìm kiếm thông tin trên tập các document và $w_{1,q}$ là trọng số của từ w_1 trong truy vấn này. Có một vài phương pháp để tính toán các giá trị này, trong đó thì TF-IDF (term frequency - inverse document frequency) là phương pháp thống kê được sử dụng rộng rãi nhất để xác định độ quan trọng của một từ trong đoạn tài liệu trong một tập nhiều đoạn tài liệu khác nhau.

4.1.2. Phương pháp tính TF-IDF

TF (Term-Frequency) - tần số xuất hiện của từ trong một tài liệu. Có nhiều cách để định nghĩa giá trị TF, dưới đây là một định nghĩa phổ biến:

$$\text{tf}(t) = f(t, d)$$

với $f(t,d)$ là số lần xuất hiện của từ t trong tài liệu d.

IDF (Inverse Document Frequency) - là phép đo thể hiện giá trị thông tin của một từ cung cấp. Khi chúng ta chỉ gán trọng số của một từ bằng giá trị TF, ta xem mỗi từ trong tài liệu đều quan trọng như nhau. Cách tính này gặp hạn chế khi gặp một số từ vốn xuất hiện nhiều nhưng rất ít quan trọng, ví dụ như "he", "about", "of",… trong tiếng Anh. Với IDF thì ta có thể biết thông tin mà từ đó cung cấp là phổ biến hay hiếm trong tất cả các tài liệu.

$$\text{idf}(t) = \frac{1}{|t \in D : t \in d|}$$

với $|t \in D : t \in d|$ là số lượng tài liệu có chứa từ t. Một từ có giá trị IDF cao khi và chỉ khi tần suất xuất hiện của nó thấp trong tất cả các tài liệu.

Từ đó giá trị **TF-IDF** được tính bằng:

$$\text{tf\_idf}(t) = \text{tf}(t) \times \text{idf}(t)$$

Một từ có giá trị tf-idf cao chứng tỏ rằng từ đó xuất hiện nhiều trong một hay một vài tài liệu nào đó và nó chỉ có mặt ở rất ít tài liệu trong toàn bộ tập tài liệu.

4.1.3. Độ đo tương đồng cosine

Độ đo tương đồng cosine của hai vector lần lượt đại diện cho tài liệu d và q có công thức như sau:

$$\text{cosine}(d, q) = \frac{d.q}{\|d\|_2 \, \|q\|_2}$$

4.2. Phần thực hành

Đầu vào:

Cho tập tài liệu như sau:

- D1: Students studying math
- D2: Math is an important subject
- D3: My brother is very hard working in math
- D4: I love math

Cho câu truy vấn như sau “math important subject”.

Yêu cầu: lập chỉ mục cho tập tài liệu và câu truy vấn bằng Vector space model, từ đó tính độ tương đồng và xếp hạng tài liệu theo độ tương đồng.

4.2.1. Tiền xử lý và lập chỉ mục tài liệu

B1, Tiền xử lý tập tài liệu bằng các bước đã nêu trong Chương 3:

- D1 [student, studi, math]
- D2 [math, import, subject]
- D3 [brother, hard, work, math]
- D4 [love, math]

B2, Thống kê các từ có trong tài liệu:

Work	Doc ID	Work	Doc ID
student	1	brother	3
studi	1	hard	3
math	1	work	3
math	2	math	3
import	2	love	4
subject	2	math	4

(bảng 1)

B3, Tạo tập từ điển

Vocabulary	Posting List (Doc ID)	Vocabulary	Posting List (Doc ID)
student	1	brother	3
studi	1	hark	3
math	1, 2, 3, 4	work	3
import	2	love	4
subject	2		

(bảng 2)

B4, Tạo PostingList kèm giá trị TF. Với TF của mỗi từ được tính bằng số lần xuất hiện của từ đó trong mỗi tài liệu. Và N.Doc là cột cho biết lượng tài liệu có chứa các từ trong tử điển.

Vocabulary	N.Doc	Posting List(Doc ID, tf)
student	1	(1,1)
studi	1	(1,1)
math	4	(1,1), (2,1), (3,1), (4,1)
import	1	(2,1)
subject	1	(2,1)
brother	1	(2,1)
hark	1	(3,1)
work	1	(3,1)
love	1	(4,1)

(bảng 3)

B5, Tính giá trị TF-IDF

Vocabulary	N.doc	IDF	Posting List (Doc ID, TF-IDF)
student	1	1	(1,1)
studi	1	1	(1,1)
math	4	¼	(1,¼), (2, ¼), (3, ¼),(4, ¼),
import	1	1	(2,1)
subject	1	1	(2,1)

brother	1	1	(2,1)
hark	1	1	(3,1)
work	1	1	(3,1)
love	1	1	(4,1)

(bảng 4)

B6. Tính hệ số chuẩn hóa TF-IDF cho các tài liệu

Norm doc 1	Norm doc 2	Norm doc 3	Norm doc 4
$\frac{\sqrt{33}}{4}$	$\frac{\sqrt{33}}{4}$	$\frac{7}{4}$	$\frac{\sqrt{17}}{4}$

(bảng 5)

B7. Tính TF-IDF chuẩn hóa

Vocabulary	N.Doc	IDF	Posting List(Doc ID, TF-IDF/Norm(n))
student	1	1	(1, 0.6936)
studi	1	1	(1, 0,6936)
math	4	¼	(1, 0.1740), (2, 0.1740), (3, 0.1429), (4,0.2425)
import	1	1	(2, 0.6963)
subject	1	1	(2, 0.6963)
brother	1	1	(3,0.571)
hark	1	1	(3,0.571)
work	1	1	(3,0.571)

love	1	1	(4,0.9701)

(bảng 6)

4.2.2. Lập chỉ mục cho câu truy vấn

B1, Tiền xử lý câu truy vấn bằng các bước đã nêu trong Chương 3:

Truy vấn sau khi qua tiền xử lý: [math, import, subject]

B2. Tính giá trị tf-idf cho câu truy vấn:

Ở đây, giá trị TF được tính nội bộ câu truy vấn và giá trị IDF được lấy ở trong bảng 4.

Term	TF	TF-IDF
math	1	¼
import	1	1
subject	1	1

(bảng 7)

B3. Tính hệ số chuẩn hóa TF-IDF cho câu truy vấn

Norm query
$\frac{\sqrt{33}}{4}$

B4. Tính TF-IDF chuẩn hóa cho câu truy vấn

Term	TF-IDF/Norm query
math	0.1740
import	0.6963
subject	0.6963

(bảng 8)

4.2.3. Tính độ tương đồng và xếp hạng tài liệu

B1. Nhân các giá trị TF-IDF của từng term trong câu truy vấn ở bảng 6 với các giá trị TF-IDF của từng term trong tập từ điển ở bảng 8.

Term	Doc ID, similarity
math	(1,0.03202), (2,0.03202), (3,0.0248), (4,0.0421)
import	(2, 0.4848)
subject	(2, 0.4848)

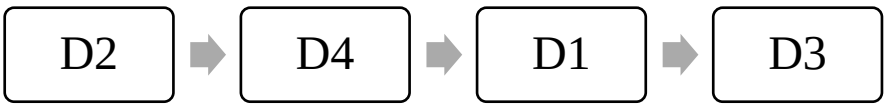
(bảng 9)

B2. Với mỗi tài liệu (doc id), cộng tất cả các giá trị similarity của mỗi term lại với nhau, ta thu được độ đo tương đồng cosine theo công thức lý thuyết.

D1	D2	D3	D4
0.03202	1	0.0248	0.0412

(bảng 10)

B3. Từ bảng 10 chúng ta xếp hạng các tài liệu liên quan tới câu truy vấn với độ tương đồng giảm dần.



Chương 5. Các bước thực hiện mô hình LSA

5.1. Xây dựng ma trận Term-document

Bước đầu tiên của phương pháp truy xuất theo mô hình SLA là thực hiện xây dựng ma trận Term-doc, trên lý thuyết thì chúng ta có thể thay ma trận Term-doc bằng các ma trận trọng số như TF, hay TF-IDF, mỗi phương pháp chúng ta chọn để xây dựng ma trận ban đầu hay còn gọi là ma trận A (với số cột tương ứng với số lượng Documents có trong tập tài liệu và số dòng tương ứng với số tập từ điển có trong toàn bộ tài liệu) sẽ cho ra kết quả truy xuất cũng như so khớp khác nhau, nên tùy vào đặc thù của tập tài liệu cũng như tập từ vận hay câu truy vấn mà ta nên chọn phương pháp tính trọng số tương ứng.

Ví dụ minh họa với tập tài liệu gồm 5 Documents là 1 câu query như sau:

Đầu vào:

Cho tập tài liệu

- D1 Students studying math
- D2 Math is an important subject
- D3 My brother is very hard working in school
- D4 Students in schoool
- D5 I love my brother

Cho câu truy vấn: "student math"

Yêu cầu: lập chỉ mục cho tập tài liệu và câu truy vấn bằng LSI model, từ đó tính độ tương đồng và xếp hạng tài liệu theo độ tương đồng.

Với tập tài liệu trên sau khi qua các bước tiền xử lý đã nêu ở phần trước thì ta sẽ có được tập từ vựng (posting list) từ đó ta có thể lập được bảng như sau:

		d1	d2	d3	d4	d5
0	student	1	0	0	1	0
1	studi	1	0	0	0	0
2	math	1	1	0	0	0
3	import	0	1	0	0	0
4	subject	0	1	0	0	0
5	brother	0	0	1	0	1
6	hard	0	0	1	0	0
7	work	0	0	1	0	0
8	school	0	0	1	1	0
9	love	0	0	0	0	1

(bảng 1)

Từ bảng Term-doc này ta có thể suy ra được ma trận Term-doc tương tự như trên:

```
A = [[1,0,0,1,0],
     [1,0,0,0,0],
     [1,1,0,0,0],
     [0,1,0,0,0],
     [0,1,0,0,0],
     [0,0,1,0,1],
     [0,0,1,0,0],
     [0,0,1,0,0],
     [0,0,1,1,0],
     [0,0,0,0,1],]
```

(bảng 2)

5.2. Tính Σ, Z, U^T

Trong mô hình LSA, chúng ta sẽ tiếp cận một thuật toán được gọi là đặc trưng của phương pháp này đó là thuật toán SVD:

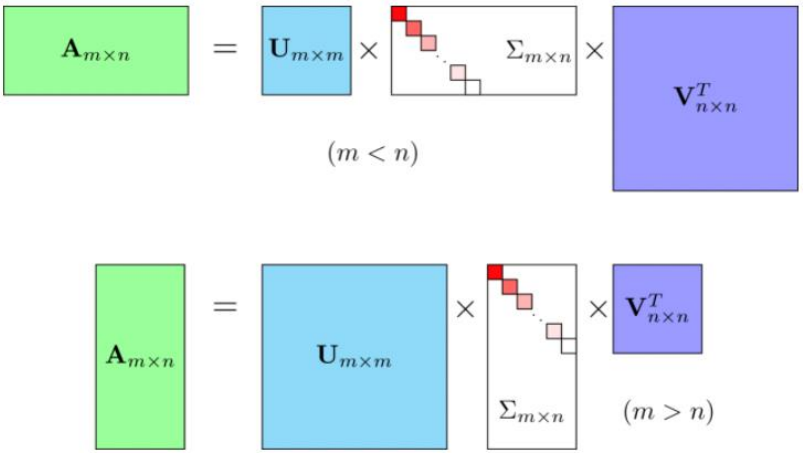
5.2.1 Phát biểu thuật toán SVD

Một ma trận A_{mxn} bất kỳ đều có thể phân tích thành dạng:

$$A_{mxn} = U_{m\times n}\,\Sigma_{m\times n}\,(V_{n\times n})^T \qquad (*)$$

Trong đó, U,V là các ma trận trực giao, Σ là ma trận đường chéo không vuông với các phần tử trên đường chéo giảm dần: $\sigma_1 \geq \sigma_2 \geq \cdots \geq$, $\sigma_r \geq 0 = 0 = \cdots =$ và r là rank của ma trận A. Lưu ý rằng mặc dù Σ không phải ma trận vuông, ta vẫn có thể coi nó là ma trận chéo nếu các thành phần khác không của nó chỉ nằm ở vị trí đường chéo, tức tại các vị trí có chỉ số hàng và chỉ số cột là như nhau. Số lượng các phần tử khác 0 trong Σ chính là rank của ma trận A.

Chú ý rằng cách biểu diễn (*) không là duy nhất vì ta chỉ cần đổi dấu của cả U và V thì (*) vẫn thoả mãn. Tuy vậy, người ta vẫn thường dùng 'the SVD' thay vì 'a SVD'. Hình 1 mô tả SVD của ma trận A_{mxn} trong hai trường hợp: m < n và m > n. Trường hợp m – n có thể xếp vào một trong hay trường hợp trên.



Hình 2: SVD cho ma trận A khi: m < n (hình trên), và m > n (hình dưới). Σ là một ma trận đường chéo với các phần tử trên đó giảm dần và không âm. Màu đỏ càng đậm thể hiện giá trị càng cao. Các ô màu trắng trên ma trận này thể hiện giá trị 0.

5.2.2 Một số ứng dụng của SVD

Như ta đã biết, theo sơ sở lý thuyết của thuật toán SVD thì nó sẽ giúp chúng ta loại bỏ đi những phần, được gọi là mang thông tin không quan trọng trong một ma trận để giảm được số chiều của ma trận đó mà không làm ma trận ban đầu thay đổi quá nhiều vì thế nên chúng ta thấy người ta sẽ ứng dụng thuật toán SVD vào các bài toán nén hình ảnh (có mất mát).

5.2.3 Sử dụng SVD để tính các ma trận Σ, Z, và U^T

Dựa trên cơ sở thuật toán SVD ta sẽ tính được ba ma trận trên như sau:

```
S [[-0.31477849  0.3970884  -0.56073601  0.11156553  0.28814966  0.20098842
   0.33650043  0.33650043 -0.21933629 -0.13551201]
 [-0.13729361  0.32912903 -0.2386758   0.26339617 -0.47590006  0.23412924
  -0.11309927 -0.11309927  0.56483004  0.34722851]
 [-0.21202825  0.62602778  0.2102724   0.06519349 -0.24765952 -0.43511765
  -0.22340116 -0.22340116 -0.34549375 -0.2117165 ]
 [-0.07473464  0.29689875  0.4489482  -0.19820268  0.22824054 -0.06900811
   0.2541582   0.2541582   0.61232506 -0.32316631]
 [-0.07473464  0.29689875  0.4489482  -0.19820268  0.22824054  0.50412576
  -0.03075704 -0.03075704 -0.2668313   0.5348828 ]
 [-0.49533777 -0.27396559  0.27536499  0.43549667 -0.02769095  0.44019768
  -0.13269991 -0.13269991 -0.04386726 -0.4271024 ]
 [-0.36624544 -0.18583285  0.08783351 -0.21345553 -0.35315593 -0.11960463
   0.73460017 -0.26539983 -0.08773451  0.14579519]
 [-0.36624544 -0.18583285  0.08783351 -0.21345553 -0.35315593 -0.11960463
  -0.26539983  0.73460017 -0.08773451  0.14579519]
 [-0.54373033 -0.11787348 -0.2342267  -0.36528616  0.4108938  -0.20098842
  -0.33650043 -0.33650043  0.21933629  0.13551201]
 [-0.12909233 -0.08813274  0.18753148  0.64895219  0.32546498 -0.44019768
   0.13269991  0.13269991  0.04386726  0.4271024 ]]
```

Z =

Σ = `Sigma [2.19933656 2.02695745 1.57110367 1.2927013  0.95651388]`

```
U chuyen vi [[-0.30195485 -0.16436662 -0.80549699 -0.390349   -0.28391748]
 [ 0.66713054  0.60180112 -0.37667528  0.13775076 -0.1786413 ]
 [-0.37498442  0.70534417  0.13799555 -0.50598998  0.2946314 ]
 [ 0.34049257 -0.25621686 -0.27593424 -0.19627166  0.83890134]
 [-0.45520502  0.21831525 -0.33779855  0.73082417  0.31131177]]
```

U^T=

Các ma trận trên tương ứng Σ, Z, U^T = SVD(A)

Với A là ma trận Term-doc đã xác đinh được ở bước 1 (A là kết quả của bảng 2).

5.3. Chọn lại số chiều cho các ma trận trên

Số chiều k được coi như là một siêu tham số, vì với mỗi cách chọn k khác nhau thì chúng ta sẽ thu được kết quả cũng như độ chính xác khác nhau cho mô hình, vì thế nên việc hơn K nên sử dụng phương pháp thực nghiệm, từ đó chọn số chiều k với k cho ra kết quả cao nhất.

Vì là ví dụ minh họa nên chúng tôi sẽ chọn số k bằng 2 để dễ dàng thực hiện mà tính toán được dễ hình dung. Dưới đây là ba ma trận σ, S, và U^T sau khi chọn lại số chiều k=2:

```
[[2.19933656 0.        ]
 [0.         2.02695745]];
```

S=

```
array([[-0.31477849,  0.3970884 ],
       [-0.13729361,  0.32912903],
       [-0.21202825,  0.62602778],
       [-0.07473464,  0.29689875],
       [-0.07473464,  0.29689875],
       [-0.49533777, -0.27396559],
       [-0.36624544, -0.18583285],
       [-0.36624544, -0.18583285],
       [-0.54373033, -0.11787348],
       [-0.12909233, -0.08813274]]);
```

σ=

```
array([[-0.30195485, -0.16436662],
       [ 0.66713054,  0.60180112],
       [-0.37498442,  0.70534417],
       [ 0.34049257, -0.25621686],
       [-0.45520502,  0.21831525]])
```

U^T =

5.3.1. Tính ma trận từ khác theo công thức

K = S*Z

```
array([[-0.69230385,  0.8048813 ],
       [-0.30195485,  0.66713054],
       [-0.46632148,  1.26893166],
       [-0.16436662,  0.60180112],
       [-0.16436662,  0.60180112],
       [-1.08941448, -0.55531658],
       [-0.80549699, -0.37667528],
       [-0.80549699, -0.37667528],
       [-1.19584599, -0.23892452],
       [-0.28391748, -0.1786413 ]])
```

K =

5.3.2. Tính ma trận Document theo công thức

$D = U^T * Z$

```
D = array([[-0.66410035, -0.33316415],
           [ 1.46724458,  1.21982527],
           [-0.82471695,  1.42970261],
           [ 0.74885775, -0.51934067],
           [-1.00114904,  0.44251572]])
```

5.4. Tính vector truy vấn

Sau khi đã có vector doc thì ta phải thực hiện tính vector truy vấn đế tiến hành so khớp, thì với mô hình LSA chúng ta sẽ tính vector truy vấn theo công thức $q = K[q_0] + K[q_1] + \dots K[q_n]$ với K là chỉ số tương ứng của ma trận từ khóa đã tính ở phần vector document, K[q] là các từ xuất hiện trọng câu query.

0	student	5	brother
1	studi	6	hard
2	math	7	work
3	import	8	school
4	subject	9	love

Ở ví dụ này ta thấy, câu truy vấn "Student math" sẽ tương ứng với K[0] và K[2]. Nên q = K[0] + K[2], suy ra ta có vector truy vấn tương ứng

```
q = array([-1.15862533,  2.07381296])
```

Sau đó tiến hành xếp hạng các vector truy vấn với vector doc với đầu đo cosine đặc trưc cho sự tương đồng của vector để tính toán và xếp hạng. Chọn kết quả theo độ chính xác giảm dần (ở đâu là từ 1 trở về) sau đó trả về kết quả:

```
0.04448965604129056
0.18304838515099536
0.9999057886116011
-0.8982847280399011
0.7990302993240597
```

Với kết quả trên thì ta sẽ xếp hạng các Doc theo thứ tự như sau D3, D4, D2, D1, D0 về trường hợp này ta nhận thấy D3 không có từ nào giống như câu query nhưng mà lại được xếp đầu, điều này có thể giải thích được bởi vì từ student với math có ở câu truy vấn nằm khá gần nhóm từ hark, work, school nên có thể sẽ được đưa lên đầu.

5.5. Lập chỉ mục với mô hình VSM + LSI để xếp hạng

Theo lý thuyết thì qua 4 bước trên ta có thể cho ra được kết quả xếp hạng cũng nhưu hoàn thành được yêu cầu của truy xuất, tuy nhiên với số lượng Doc lớn thì lúc so khớp sẽ mất rất chiều thời gian, cũng như đặc trưng của mô hình LSA là sẽ lấy theo các từ ngữ có mặc ngữ nghĩa tiềm ẩn, hay nói cách khác là nó sẽ có xu hướng gom các từ gần nghĩa, cũng nghĩa lại với nhau nên sẽ cho ra kết quả không liên quan khá là nhiều, vì thế để tối ưu thì ta có thể thực hiện chọn các Doc có liên quan trước bằng mô hình VSM sau đó tiến hành chọn lại và tính toán trên tập Doc đã được chọn lọc.

Ta sẽ lưu lại vector từ khóa K và vector tài liệu D tương ứng, sau đó tiến hạnh chọn các tài liệu có liên quan trước:

Dic	
student	1,4
studi	1
math	1,2
import	2
subject	2

brother	3,5
hard	3
work	3
school	3,4
love	5

Từ bảng trên ta thấy hai từ student và math sẽ xuất hiện ở các Doc: 1, 2, 4 .

Tiếp theo đó ta tiến hành xếp hạng lại trên 3 Doc: 1,2,4 thay vì 5 Doc như ban đầu, qua đó ta thấy sẽ giảm được rất nhiều tài nguyên cho việc so khớp:

```
0.04448965604129056
0.18304838515099536
-0.8982847280399011
```

Đây là kết quả sau khi tiến hành so khớp bằng đầu đo cosine trên 3 doc 1,2,4 theo thứ tự Do đó ta sẽ thấy keeys quả sẽ có thay đổi với phương pháp không sử dụng VSM để chọn Doc trước đó là D2, D1, D4 thay vì D3, D4, D2, D1, D0.

Chương 6. Đánh giá mô hình truy xuất thông tin

Mục tiêu của việc đánh giá hệ thống truy vấn là để kiểm tra mức độ thỏa mãn của kết quả tìm kiếm đối với mục đích truy vấn của người dùng. Precision (độ chính xác) và recall (độ phủ) là chỉ tiêu đánh giá cơ bản để kiểm tra hiệu năng của mô hình truy xuất thông tin, tức là khả năng trả về thông tin liên quan tới câu truy vấn của người dùng. Nhờ những thông số đánh giá này mà chúng ta có thể cải thiện hoặc tìm những mô hình truy vấn thông tin mang kết quả tốt nhất để thực nghiệm.

6.1. Độ chính xác và độ phủ

6.1.1. Độ chính xác (precision):

$$\text{precision} = \frac{\text{số tài liệu mô hình trả về liên quan}}{\text{tổng số tài liệu mô hình trả về}}$$

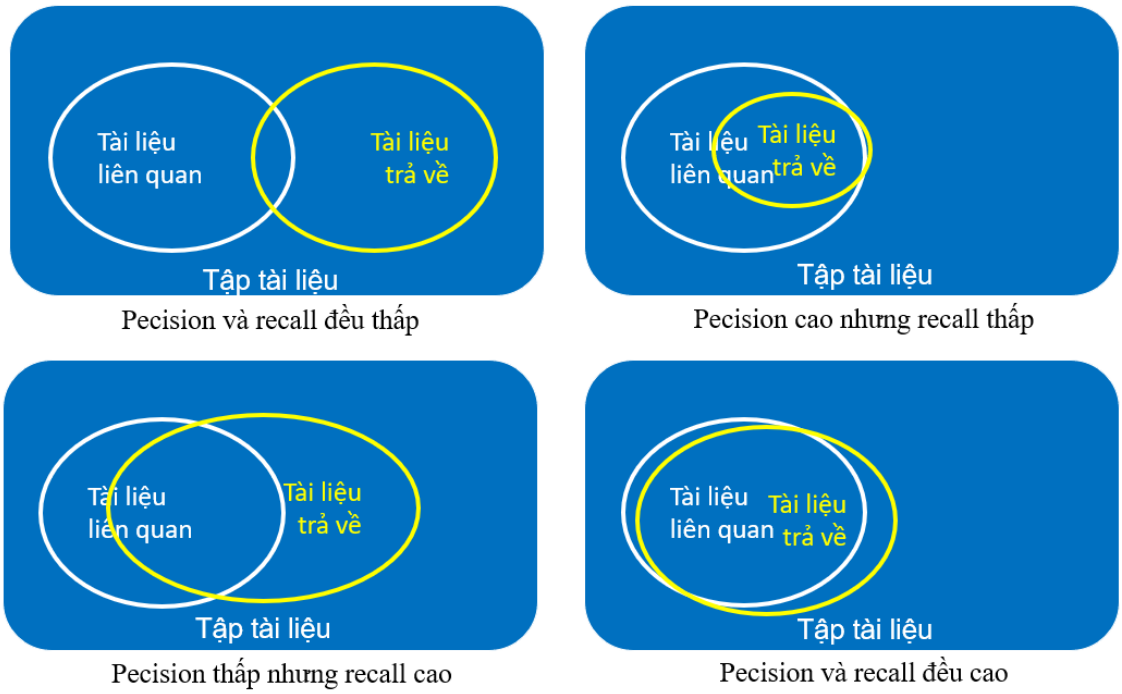
6.1.2. Độ phủ (recall)

$$\text{recall} = \frac{\text{số tài liệu mô hình trả về liên quan}}{\text{tổng số tài liệu liên quan}}$$

Precision và recall là hai đại lượng có quan hệ mật thiết với nhau, việc đánh giá một mô hình truy xuất thông tin cho toàn bộ tài liệu trả về phải dựa trên cùng lúc hai độ đo này. Chẳng hạn như chẳng có gì để nói khi một mô hình đạt 100% recall khi mô hình đó trả về toàn bộ tài liệu trong tập tài liệu, vì thế việc đánh giá thêm trên precision là cần thiết.

Độ đo precision và recall cũng có thể được đánh giá ở một thứ hạng giới hạn nhất định, có nghĩa là nó sẽ chỉ xem xét một vài các kết quả cao nhất được hệ thống trả về. Precision hay recall được tính cho *n* kết quả trả về đầu tiên được viết là *R@n* hay *P@n*.

Dưới đây là các minh họa về precision và recall:



Pecision và recall đều thấp — Pecision cao nhưng recall thấp

Pecision thấp nhưng recall cao — Pecision và recall đều cao

Hình 3: Minh họa hiệu năng của mô hình truy xuất thông tin qua trường hợp

Độ chính xác và độ phủ có quan hệ chặt chẽ với nhau, và quan hệ này là quan hệ đánh đổi (trade-off). Để minh họa cho điều này, ta xét trường hợp "Precision thấp nhưng recall cao", nếu tìm cách thu hẹp khu vực "Tài liệu trả về" về phía "Tài liệu liên quan", ta sẽ gặp trường hợp ở hình "Precision cao nhưng hiệu năng thấp".

6.2. Average Precision

Độ chính xác và độ phủ là các chỉ số giá trị đơn lẻ dựa trên toàn bộ hoặc một phần danh sách tài liệu được hệ thống trả về. Đối với các hệ thống trả về một chuỗi tài liệu được xếp hạng, ta cần một độ đo hiệu năng có thể kết hợp tính chất cả hai độ đo trên và dựa trên thứ tự trình bày các tài liệu trả về.

Bằng cách tính toán độ chính xác và độ phủ ở mọi vị trí trong chuỗi tài liệu đã được xếp hạng, người ta có thể vẽ một precision-recall curve, vẽ biểu đồ độ chính xác P(r) là hàm số của độ phủ r trong khoảng [0, 1], gọi là Average Precision.

$$\text{AvgP} = \int_0^1 \text{P}(r)\, dr$$

Trong đó Average Precision (AvgP) là diện tích hình phẳng giới hạn bởi P(r). Trong thực tế, diện tích này được xấp xỉ bằng một tổng hữu hạn trên mọi vị trí trong chuỗi tài liệu được xếp hạng:

$$\text{AvgP} = \sum_{k=1}^{n} \text{P}(k)\, \Delta r(k)$$

Với k là thứ hạng tài liệu trong danh sách truy vấn tải về, n là tổng số tài liệu trả về, P(k) là độ chính xác được tính cho k tài liệu đầu tiên (hay P@k), $\Delta r(k)$ là sự thay đổi của độ phủ từ tài liệu thứ k-1 đến k.

Ví dụ: Xét tập tài liệu có 5 tài liệu và một câu truy vấn, cùng với đó là hai hệ thống truy vấn A và B có kết quả truy vấn trả về lần lượt là {R, R, N, R, N} và {N, N, R, R, R} (với R là tài liệu liên quan tới câu truy vấn và N là tài liệu không liên quan).

Thực nghiệm với ví dụ đề cập ở trên, ta có bảng tính $P(k)$ và $\Delta r(k)$ của hai mô hình A và B như sau:

	Mô hình truy xuất A			Mô hình truy xuất B		
Tài liệu (k)	R hay N	P(k)	Δr(k)	R hay N	P(k)	Δr(k)
1	R	1/1	1/3	N	0	0
2	R	2/2	1/3	N	0	0
3	N	2/3	0	R	1/3	1/3
4	R	3/4	1/3	R	1/4	1/3
5	N	3/5	0	R	1/5	1/3

Từ đó ta tính được:

$$\text{AvgP(A)} = \sum_{k=1}^{10} \text{P}(k)\,\Delta r(k) = \left(\frac{1}{1}+\frac{2}{2}+\frac{3}{3}\right)\times\frac{1}{3} \approx 0.92$$

$$\text{AvgP(B)} = \sum_{k=1}^{10} \text{P}(k)\,\Delta r(k) = \left(\frac{1}{3}+\frac{1}{4}+\frac{1}{5}\right)\times\frac{1}{3} \approx 0.26$$

6.3. Interpolated Average Precision

mAP được thực hiện theo phương pháp nội suy 11 điểm trên toàn bộ dữ liệu, chính là lấy trung bình của các giá trị P' ở 11 điểm recall với các mức *r'* trải dài từ 0.1 đến 1 với bước nhảy là 0.1.

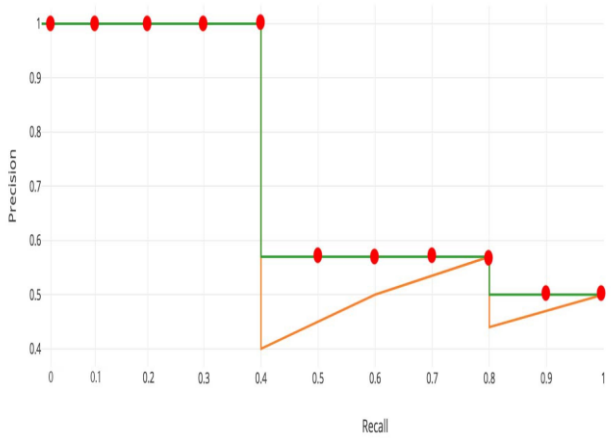
P' ở độ phủ *r* được định nghĩa là *p* cao nhất được tìm thấy ở bất kì độ mức phủ nào. Giá trị nội suy precision được tính theo công thức sau:

$$\text{P}'(r) = \max_{r':r' \geq b} p(r')$$

Từ đó ta tính giá trị nội suy 11 điểm chính là trung bình các giá trị P' ở 11 điểm phủ *r*:

$$\text{AvgP} = \frac{1}{11} \sum_{r\in\{0,\,0.1,\ldots,1\}} \text{P}'(r)$$

Hình 4 là ví dụ về sự khác biệt của Interpolated Average Precision và Average Precision:



Hình 4 [6]: Minh họa về độ chính xác (precision) của phương pháp đánh giá Interpolated Average Precision (giới hạn bởi các đoạn thẳng màu xanh) và Average Precision (giới hạn bởi các đoạn thẳng màu vàng) trên 11 mức độ phủ (recall) với giá trị nội suy P'(r) là các chấm màu đỏ.

6.4. Mean Average Precision

Trung bình của trung bình bộ chính xác trung bình (MAP) cho một tập hợp các truy vấn là giá trị trung bình của điểm độ chính xác trung bình (AvgP) cho mỗi truy vấn.

$$\text{MAP} = \frac{\Sigma_{q=1}^{Q} \text{AvgP}(q)}{Q}$$

Trong đó Q là số lượng các truy vấn.

Chương 7. Cài đặt thử nghiệm

7.1. Bộ ngữ liệu

7.1.1. Bộ ngữ liệu CRANFIELD

Bộ ngữ liệu theo ngôn ngữ tiếng Anh, được sử dụng rộng rãi trong các thực nghiệm về bài toán truy xuất thông tin. Bộ ngữ liệu này chứa tập nội dung tài liệu của 1400 bản tóm tắt ở đa dạng chủ đề khác nhau và tập các câu truy vấn kèm theo độ tương đồng giữa các tài liệu liên quan đến câu truy vấn đó để thực hiện đánh giá mô hình truy xuất. Vỡi mỗi tập tài liệu của CRANFIELD sẽ gồm các trường:

- ‘.I’: Số thứ tự của tài liệu
- ‘.T.’: Tiêu đề của tài liệu
- ‘.A’: Tác giả của tài liệu
- ‘.B’: Thông tin lề của tài liệu
- ‘.W’: Nội dung tóm tắt của tài liệu

Về phần thực nghiệm, nhóm chỉ lấy toàn bộ phần '.W' làm tài liệu cho mô hình truy xuất thông tin vì dòng đầu tiên của nội dung trường này bao gồm cả tiêu đề '.T' và nội dung tóm tắt chính của tài liệu. Có 2 tài liệu không chứa bất kì nội dung chữ nào là '.I 471' và ".I 995" nhưng nhóm vẫn lưu dưới dạng tài liệu rỗng để không ảnh hưởng đến thứ tự các tài liệu dùng để truy vấn. Khảo sát cho thấy tài liệu 995 cũng có trong các tài liệu liên quan (relevant) trong các kết quả trả về của câu truy vấn, có thể là thành phần gây nhiễu cho kết quả truy vấn, song cũng không ảnh hưởng nhiều đến kết quả. Tập tài liệu dùng để truy vấn gồm 225 câu liên quan đến tài liệu.

7.1.2. Bộ ngữ liệu NFCORPUS:

Bộ ngữ liệu theo ngôn ngữ tiếng Anh, với nội dung xoay quanh chủ đề y học thường được sử dụng cho bài toán Medical Information Retrieval (Truy xuất thông tin y học). Bộ ngữ liệu được chia thành 3 tập dữ liệu: train, test, dev theo tỉ lệ lần lượt là 80%, 10%, 10%. Mỗi tập dữ liệu đều kèm theo tập các câu truy vấn cùng với độ tương đồng giữa các tài liệu liên quan đến câu truy vấn đó để thực hiện đánh giá mô hình truy xuất. Với mỗi tập tài liệu NFCORPUS sẽ bao gồm: ID - số thứ tự tài liệu và TEXT - nội dung tài liệu đó.

Về phần thực nghiệm, nhóm sử dụng "file.docs" làm tài liệu xây dựng mô hình truy xuất và "file.all.queries" làm tập truy vấn bao gồm các thông tin: tiêu đề, mô tả, chủ đề, nội dung và bình luận. Thực nghiệm của nhóm cho thấy sử dụng "file.all.queries" làm tập truy vấn thay vì sử dụng "file.titles.queries" vì kết quả thực nghiệm cho thấy sử dụng hết thông tin văn bản sẽ cho ra kết quả tốt hơn, sẽ được đề cập ở phần nội dung sau.

Stopwords sẽ được loại bỏ bằng bộ stopwords được NFCORPUS cung cấp sẵn thay vì sử dụng thư viện để thực hiện. Tập truy vấn gồm 2594 câu ở tập train, 325 câu ở tập test và dev.

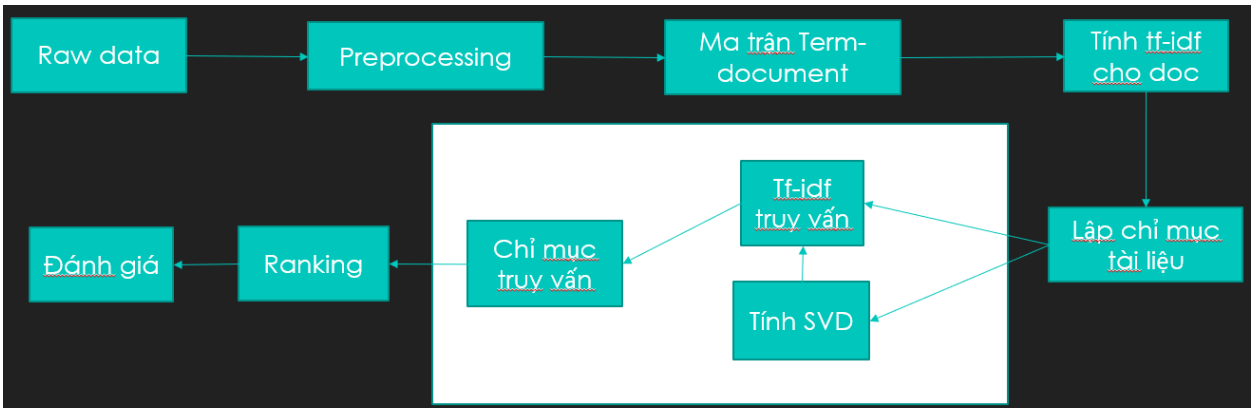
7.2. Thư viện sử dụng

Thư viện nhóm sử dụng bao gồm:

- nltk (Natural Language Toolkit): là thư viện Python hỗ trợ xử lý ngôn ngữ tự nhiên phổ biến nhất, giúp việc thực hiện xử lý bài toán ngôn ngữ dễ dàng và nhanh chóng hơn.
- gensim: là thư viện Python mã nguồn mở dùng cho nhiều tác vụ khác nhau như: mô hình hoá, lập chỉ mục tài liệu và rút trích tính tương tự với ngữ liệu lớn. Gensim có chức năng phân tích ngữ nghĩa tiềm ẩn, phân tích ma trận LSA và ma trận TF-IDF.
- numpy (Numeric Python): là thư viện toán học phổ biến và mạnh mẽ của Python, cho phép làm việc hiệu quả và tốc độ xử lý nhanh với tính toán ma trận và mảng.

7.3. Pipeline

Sau đây là tổng quan của mô hình truy xuất nhóm đã thực hiện:



Hình 5: Pipeline của mô hình nhóm thực hiện

Dữ liệu thô ban đầu sẽ được tiền xử lý để tìm tập term của tài liệu: với mỗi nội dung tài liệu, sẽ được xử loại bỏ các kí tự đặc biệt bằng gensim, sau đó tiến hành loại bỏ stopwords (CRANFIELD sử dụng thư viện nltk và NFCORPUS sử dụng stopwords được cung cấp sẵn), và cuối cùng là Steamming term bằng nltk.

Ma trận term-document được tạo ra từ tập term, sau đó tính trọng số TF-IDF và lập chỉ mục cho tập tài liệu này.

```
[[(32, 0.002315523664277359),
  (33, 0.0005577226369853424),
  (60, 0.003298172977800566),
  (87, 0.0019176328267168077),
  (103, 0.00207108917231462),
  (111, 0.001247234390497613),
  (207, 0.003075566934775168),
  (266, 0.0035873045759648625),
  (267, 0.0026167671804609737),
  (268, 0.0032805904593704973),
  (269, 0.0027680788973944437),
```

Hình 6: Minh hoạ về lập chỉ mục tf-idf cho tài liệu

Bước tiếp theo, nhóm thực nghiệm hai mô hình là Vector Space Model TF-IDF và mô hình LSI sử dụng thuật toán SVD ngay trên tập tài liệu TF-IDF, tức sử dụng LSI trên mô hình TF-IDF. Nhằm xây dựng mô hình LSI, nhóm sử dụng thư viện gensim để lập chỉ mục cho tập tài liều.

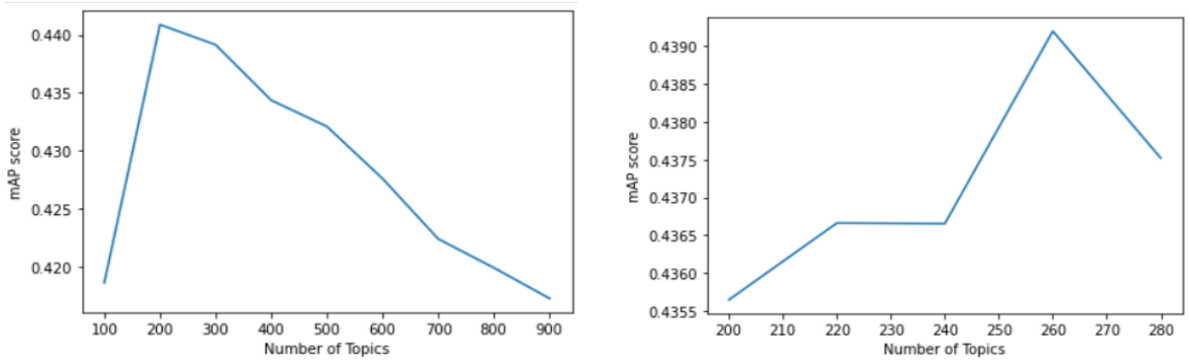
```
[(0, 0.16800342273043603),
 (1, -0.015095275558128278),
 (2, 0.09843736164586996),
 (3, 0.14586960249330438),
 (4, -0.0407101632104013),
 (5, -0.03785180177562449),
 (6, -0.02584938097604107),
 (7, -0.0037506982612062444),
 (8, -0.022297153848275623),
 (9, 0.05669339240658127)]
```

Hình 7: Minh hoạ về lập chỉ mục tài liệu cho mô hình LSI

Cuối cùng cả hai mô hình lập chỉ mục cho câu truy vấn của mỗi mô hình và xếp hạng dựa trên độ tương đồng cosine similarity. Kết quả cuối cùng của mô hình được đánh giá bằng mAP nội suy 11 điểm trên toàn bộ dữ liệu.

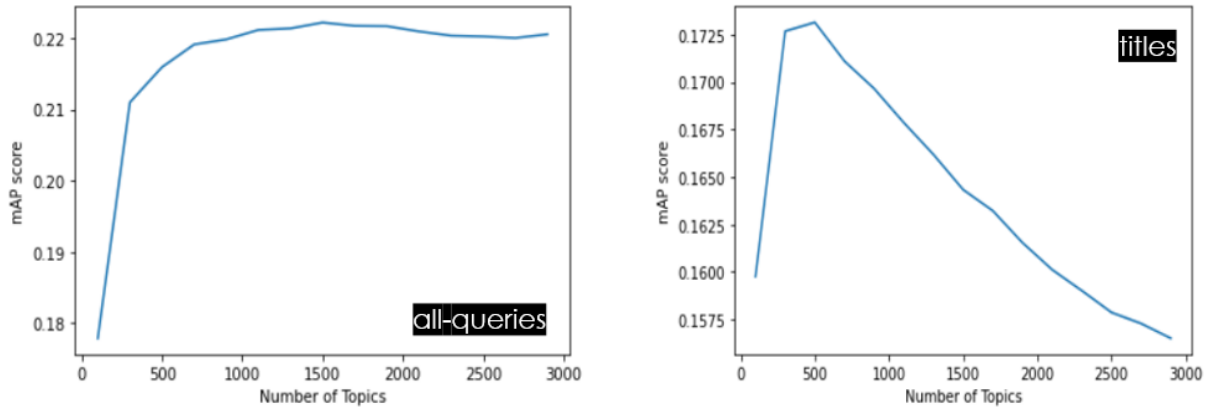
7.4. Chọn số chiều cho LSI

Tham số chính để điều chỉnh mô hình LSI chính là số k, hay là số topic (ngữ nghĩa tiềm ẩn) của tài liệu. Với ngữ liệu CRANFIELD, nhóm tiến hành chạy thực nghiệm so sánh mức k từ 100 đến 1000. Kết quả đánh giá mAP cao nhất ở mức 200, cụ thể là 260:



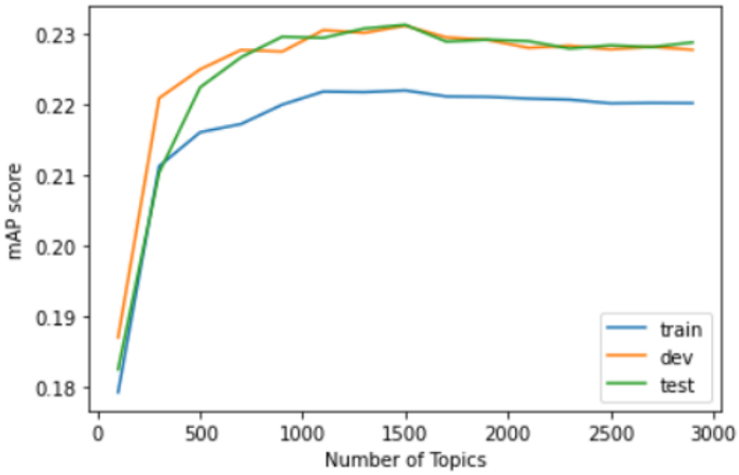
Hình 8: Duyệt số k để tìm mAP cao nhất

Với ngữ liệu NFCORPUS, nhóm tiến hành chạy thực nghiệm so sánh ở mức k từ 100 đến 3000 vì số lượng terms của NFCORPUS lớn hơn nhiều so với CRANFIELD. Nhóm tiến hành đánh giá trên hai tập truy vấn “file.all.queries” và “file.titles.queries”, kết quả cho thấy nếu chỉ lấy titles có thể khiến cho kết quả truy xuất giảm đi đáng kể so với việc lấy toàn bộ nội dung truy vấn.



Hình 9: So sánh giữa chọn toàn bộ nội dung và tiêu đề làm câu truy vấn

Sau khi chọn toàn bộ nội dung làm câu truy vấn, nhóm tiến hành chọn ra số k cho ra kết quả mAP cao nhất. Số k có thể tiếp tục tăng tới 5000 nhưng thời gian chạy sẽ lâu hơn và cũng như kết quả mAP sẽ giảm đi và bão hoà nên mốc 3000 là hoàn toàn chấp nhận được.



Hình 10: Duyệt số k để tìm mAP cao nhất

So sánh với ba tập train, dev, test ta thấy kết quả hai tập test và dev tương đồng nhau ở mốc cao nhất vượt mức 0.23, còn tập train cho kết quả thấp hơn. Nguyên nhân có thể do số câu truy vấn ở hai tập còn lại ít hơn so với tập train nên kết quả sẽ cao hơn là tập train.

7.5. Kết quả đánh giá

Sau đây là kết quả cuối cùng của phần thực nghiệm của nhóm:

	Vector Space Model	LSI
Cranfield	0.3894719	0.440868
Nfcorpus - train	0.2168682	0.221982
Nfcorpus - dev	0.2221909	0.231624
Nfcorpus - test	0.2258749	0.231236

Hình 11: Kết quả thực nghiệm của nhóm

Kết quả cao nhất của hai ngữ liệu đều thuộc về mô hình LSI với CRANFIELD là 0.44 và NFCORPUS là 0.23. So sánh ta thấy kết quả của CRANFIELD có sự khác biệt đáng kể trong khi NFCORPUS cho thấy hai mô hình Vector Space Model TF-IDF và LSI chỉ lệch nhau 0.01.

Chương 8. Kết luận và hướng phát triển

Từ thực nghiệm nhóm thấy đúc kết một số kết luận:

- Do kế thừa mô hình TF-IDF của Vector Space Model nên mô hình LSI cho ra kết quả cao hơn.
- Đối với bộ dữ liệu NFCORPUS, vì tập dev và tập test có số lượng câu truy vấn ít hơn tập train nên một phần cũng ảnh hưởng đến kết cuối cùng. Song sự sai số chỉ ở mức 0.01.
- Về kết quả NFCORPUS cả hai mô hình đều cho ra kết quả chưa thực sự cao do vấn đề ngữ cảnh vẫn là một vấn đề khó cho các bài toán truy xuất thông tin nói riêng và bài toán xử lý ngôn ngữ tự nhiên nói chung.

Với sự phát triển của các hướng nghiên cứu về bài toán truy xuất thông tin cũng như bài toán xử lý ngôn ngữ tự nhiên, đã và đang có nhiều sự tiến bộ rõ rệt, đặc biệt là các mô hình mạng học sâu. Các thư viện cũng đang ngày càng được cung cấp sẵn và dễ dàng thực hiện các mô hình mạng học sâu mà không cần tốn nhiều thời gian và công sức. Trong đó hai thư viện NLTK và Gensim đã và đang được cải thiện nhằm giải quyết vấn đề ngữ cảnh trong ngôn ngữ.

Tài liệu tham khảo

[1] Wayback Machine (archive.org)

[2] https://medium.com/betacom/latent-semantic-indexing-in-python-85880414b4de

[3] https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python

[4] https://blogcuabuicaodoanh.wordpress.com/2020/02/22/mean-average-precision-map-trong-bai-toan-object-detection/

[5] Gensim: Topic modelling for humans (radimrehurek.com)

[6] https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173

[7] https://machinelearningcoban.com/2017/06/07/svd/

[8] https://blog.marketmuse.com/glossary/latent-semantic-analysis-definition/