

A blue and white logo

Description automatically generated with medium confidence

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN MÔN HỌC**

CS419 – TRUY VẤN THÔNG TIN

VECTOR SPACE MODEL (VSM) VÀ BOOLEAN INDEPENDENCE MODEL (BIM)

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và tên** | **Email liên hệ** |
| 1 | 20520461 | Huỳnh Thái Dương | [20520461@gm.uit.edu.vn](mailto:20520461@gm.uit.edu.vn) |
| 2 | 20521072 | Nguyễn Thị Kim Anh | [20521072@gm.uit.edu.vn](mailto:20521072@gm.uit.edu.vn) |
| 3 | 205201268 | Đặng Công Hà | [20521268@gm.uit.edu.vn](mailto:20521268@gm.uit.edu.vn) |

**

**Giảng viên hướng dẫn:** ThS. Nguyễn Trọng Chỉnh

**Nhóm sinh viên thực hiện:** CS419.KHCL

**THÔNG TIN CHUNG**

|  |  |
| --- | --- |
| **Đề tài:** | Vector Space Model (VSM) và  Boolean Independence Model (BIM) |
| **Môn học:** | CS419 – Truy Vấn Thông Tin |
| **Lớp:** | CS419.N11.KHCL. |
| **Giảng viên hướng dẫn:** | ThS. Nguyễn Trọng Chỉnh. |
| **Thời gian thực hiện:** | Học kỳ 1. Năm học: 2021 – 2022. |
| **Sinh viên thực hiện:** | Huỳnh Thái Dương – 20520461.  Đoàn Công Hà – 20521268.  Nguyễn Thị Kim Anh – 20521072. |
| **Nội dung đề tài:** | Sử dụng 2 mô hình truy vấn thông tin là Vector Space Model (VSM) và Boolean Independence Model (BIM) trong truy vấn thông tin.  Thực hiện đánh giá kết quả truy vấn giữa 2 mô hình trên và áp dụng trên một mô hình truy vấn nữa là WHOOSE. |
| **Kế hoạch thực hiện:** | Tuần 1, 2: Thành lập nhóm. Chọn đề tài. Xây dựng kế hoạch thực hiện và phân công nhiệm vụ.  Tuần 3, 4, 5, 6, 7: Khảo sát tình hình thực tế. Thu thập các thông tin liên quan và tìm nguồn tài liệu tham khảo. Tìm hiểu các kiến thức cần thiết cho quá trình thực hiện đề tài.  Tuần 8, 9, 10, 11, 12: Tìm hiểu các công nghệ và công cụ có thể sử dụng trong đề tài. Thiết kế, xây dựng và viết chương trình cài đặt. Đánh giá và kiểm thử chương trình cài đặt.  Tuần 13, 14: Chuẩn bị slide trình bày và tập duyệt báo cáo.  Tuần 15: Báo cáo đồ án. Tuần Dự trữ: Chỉnh sửa báo cáo và chương trình cài đặt sau khi báo cáo. |

**

**LỜI CẢM ƠN**

Nhóm xin chân thành gửi lời cảm ơn đến ThS. Nguyễn Trọng Chỉnh – Giảng viên khoa Khoa học máy tính, Trường Đại học Công nghệ thông tin, Đại học Quốc gia thành phố Hồ Chí Minh, đồng thời là giản viên giảng dạy lớp CS419.N11.KHCL – Môn Truy vấn thông tin , trong thời gian đã tận tình hướng dẫn và định hướng cho nhóm trong suốt quá trình thực hiện và hoàn thành đồ án.

Trong quá trình thực hiện đồ án nhóm đã cố gắng rất nhiều để hoàn thành đồ án một cách tốt nhất và hoàn thiện nhất, song cũng sẽ không tránh khỏi được những sai sót ngoài ý muốn. Nhóm mong rằng sẽ nhận được những lời nhận xét và những lời góp ý chân thành từ quý thầy/cô và các bạn trong quá trình thực hiện chương trình của nhóm để chương trình ngày càng hoàn thiện hơn. Mọi thắc mắc cũng như mọi góp ý của mọi người xin gửi email về một trong các địa chỉ email sau: [20520461@gm.uit.edu.vn](mailto:20520461@gm.uit.edu.vn) (Huỳnh Thái Dương), [20521072@gm.uit.edu.vn](mailto:20521072@gm.uit.edu.vn) (Nguyễn Thị Kim Anh) và [20521268@gm.uit.edu.vn](mailto:20521268@gm.uit.edu.vn) (Đoàn Công Hà) . Mỗi ý kiến đóng góp sẽ là một nguồn động lực to lớn đối với nhóm để nhóm có thể cố gắng cải tiến chương trình ngày càng hoàn thiện và phát triển đồ án lên một mức cao hơn, nhóm cũng sẽ dựa vào đó để phát triển hơn những ưu điểm và cải thiện được phần nào đó những nhược điểm của chương trình. Hy vọng đề tài “Áp dụng hai mô hình VSM và BIM trong truy vấn thông tin” do nhóm thực hiện sẽ trở thành một công cụ hữu ích và có thể ứng dụng được trong lĩnh vực Truy Vấn Thông Tin.

Thành phố Hồ Chí Minh, tháng 1 năm 2022

**Nhóm sinh viên thực hiện**

Huỳnh Thái Dương

Nguyễn Thị Kim Anh

Đoàn Công Hà

**MỤC LỤC**

Trang

[Chương 1. TỔNG QUAN 6](#_Toc126756470)

[**1. Giới thiệu** 6](#_Toc126756471)

[**2. Mô hình tổng quát** 10](#_Toc126756472)

[**3. Các thách thức của bài toán** 11](#_Toc126756473)

[**4. Đối tượng và phạm vi nghiên cứu** 11](#_Toc126756474)

[**5. Mục tiêu bài toán** 12](#_Toc126756475)

[Chương 2. CƠ SỞ LÝ THUYẾT 12](#_Toc126756476)

[I. Tiền xử lý 12](#_Toc126756477)

[**Giới thiệu** 12](#_Toc126756478)

[**2. Các bước tiền xử lý dữ liệu** 13](#_Toc126756479)

[II. Lập chỉ mục 14](#_Toc126756480)

[III. Các bước tiền xử lý dữ liệu 14](#_Toc126756480)

[II. Ngữ liệu 14](#_Toc126756480)

[Chương 3. VectorSpace Model (VSM) 17](#_Toc126756481)

[I. Giới thiệu 12](#_Toc126756477)

[II. Quá trình lập chỉ mục 14](#_Toc126756480)

[III. Quá trình truy vấn 12](#_Toc126756477)

[II. Generalised Vector Space Model 14](#_Toc126756480)

[Chương 4. Binary Independent Model (BIM) 17](#_Toc126756481)

[I. Giới thiệu 12](#_Toc126756477)

[II. Quá trình lập chỉ mục 14](#_Toc126756480)

[III. Quá trình truy vấn 12](#_Toc126756477)

[II. Relevant Feedback 14](#_Toc126756480)

[Chương 5. CÁC PHƯƠNG PHÁP ĐÁNH GIÁ 17](#_Toc126756482)

[Chương 6. CÀI ĐẶT CHƯƠNG TRÌNH 17](#_Toc126756482)

[Chương 7. KẾT LUẬN 39](#_Toc126756489)

[**1. Nhận xét về mô hình** 39](#_Toc126756490)

[TÀI LIỆU THAM KHẢO 39](#_Toc126756491)

# **Chương 1. TỔNG QUAN**

### **1. Giới thiệu**

**Khái niệm truy vấn thông tin**

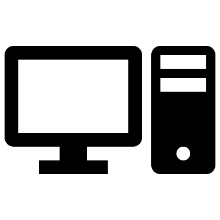
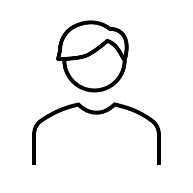
Truy vấn thông tin là một nhánh trong ngành khoa học máy tính, nghiên cứu và áp dụng máy tính vào việc phân tích một văn bản ( tài liệu có cấu trúc hoặc không có cấu trúc ) hoặc tập lưu trữ lớn nhằm mục đích tìm được một thông tin cần thiết thoả yêu cầu của người dùng từ văn bản đó hoặc đơn vị dữ liệu đó. Mục đích của những người nghiên cứu về lĩnh vực này là tận dụng những kiến thức về việc truy vấn thông tin để phát triển những công cụ và kỹ thuật phù hợp vào hệ thống máy tính sao cho máy tính có thể giải quyết những vấn đề thực tế được đặt ra có liên quan đến việc tìm kiếm thông tin trong thực tế mà mọi người đều có thể sử dụng được, đồng thời kết quả trả về phải có độ tương đồng cao so với nhu cầu của người dùng.



QUERY



??????





**TEXT**

**PROCESSING**

**QUERY**

**Mô phòng quá trình truy vấn thông tin trong một tệp văn bản.**

Nói ngắn gọn, truy vấn thông tin là một công việc cho máy tính lấy thông tin và người dùng và đi tìm thông tin đó. Ví dụ một công cụ thực tế trong truy vấn thông tin như: Google, Yahoo, …

Graphical user interface, website

Description automatically generated

**Một số ứng dụng của truy vấn thông tin hiện nay**

**Các cơ sở khoa học của xử lý ngôn ngữ tự nhiên**

* Câu: là một đơn vị ngữ pháp gồm một hay nhiều từ có liên kết ngữ pháp với nhau. Một câu có thể bao gồm các từ được nhóm lại để thể hiện một khẳng định, nghi vấn, cảm thán, yêu cầu, mệnh lệnh, hoặc đề nghị.Bản chất của ngôn ngữ tự nhiên:
* Token: Là một dạng hình thức của một từ ( cụm từ hoặc cụm kí tự ) được tách ra từ một câu hoặc từ một đoạn văn bản. Trong đó mỗi văn bản ta gồm có nhiều loại kí tự cũng như nhiều loại từ, từ đó ta tách đoạn văn đó thành nhưng Token khác nhau, quá trình này được gọi là Tokenization.
* Term: Term là một tập hợp của nhiều token có nội dung giống nhau. Tuy nhiên, một token không được xem là một term mà khi một sau quá trình lặp chỉ mục thì những token có cấu trúc giống nhau được xem là một term.

**Quá trình truy vấn thông tin**

Truy vấn thông tin gồm 4 giai đoạn: Tách tokens (Tokenization), Lập chỉ mục đảo ngược(Inverse indexed), Tách tokens trong câu truy vấn(Query tokenization), Truy vấn**(Retrieval)**:

* Tách tokens (Tokenization): Đây là bước đầu tiên trong truy vấn thông tin, trong bước này ta thức hiện tách cách một câu thành các token khác nhau, thường là các từ.
* Lập chỉ mục đảo ngược(Inverse indexed): Là quá trình bao gồm việc thực hiện lập chỉ mục cho từng token trong một câu và thực hiện tổng hợp các token thành một term.
* Tách token trong câu truy vấn(Query tokenization): Tương tự như quá trình tách token ở trên ta sẽ thực hiện tách token và ở đây ta không cần thực hiện lập chỉ mục cho từng term.
* **Truy vấn(Retrieval):** Dựa trên từng token trong câu truy vấn ta sẽ lấy chỉ mục ra từ những term có trong danh sách các term đã lập chỉ mục ở trên và từ đó tuỳ vào từng giải thuật ta sẽ có cách xử lý khác nhau.

**

1. **bước trong truy vấn thông tin**

**Một số ứng dụng của truy vấn thông tin**

* **Ứng dụng chính của truy vấn thông tin là việc tìm kiếm thông tin mà người dùng muốn như tìm kiếm trên nhưng trang web bán hàng hoặc nhưng ứng dụng, ngoài ra còn có một số việc mà truy vấn có thể làm là dựa trên nhưng nội dung trong một đoạn văn hay thậm chí là trong một thư mục thì ta có thể tìm kiếm được nội dung tương ứng,…**
* **Ngoài ra truy vấn thông tin có thể được sử dụng kết hợp với sử lý ngôn ngữ tự nhiên nhằm nâng cấp chất lượng của việc truy vấn thay vì ta chỉ lấy query bằng những gì ta nhập vào thì ở đây ta có thể kết hợp với âm thanh,…**

****Những thách thức của truy vấn thông tin**

Truy vấn thông tin tuy là một nhánh trong lĩnh vực trí tuệ nhân tạo mới phát triển gần đây nhưng đã là công cụ mạnh mẽ với những lợi ích to lớn. Tuy nhiên, vẫn còn một số hạn chế và vấn đề trong xử lý ngôn ngữ tự nhiên:

### **2. Mô hình tổng quát**

Trong báo cáo này, mô hình được sử dụng dùng để truy vấn là hai mô hình VSM( Vector Space Model ) và mô hình BIM( Boolean Indepence Model ) như đã nhắc đến ngay trong tên của tập báo cáo này. Các phương pháp đánh giá độ chính xác của mô hình cả về phần tách từ lẫn gán nhãn sẽ được trình bày trong phần sau của báo cáo.

Bài toán có thể được phát biểu một cách trực quan và ngắn gọn dưới dạng mô hình như sau:

Diagram

Description automatically generated

**Mô hình tổng quát của bài toán**

### **3. Các thách thức của bài toán**

Việc xây dựng một mô hình có khả năng giải quyết yêu cầu của bài toán đặt ra tuy có nhiều thuận lợi nhờ vào sự phát triển của máy tính hiện đại, ngôn ngữ lập trình và các công cụ lập trình. Tuy nhiên, theo nhóm nhận thấy, việc xây dựng mô hình này vẫn còn gặp phải một số thách thức trong quá trình thực hiện như “Tách term”, “Việc có sai sót ở phần tách term dẫn đến việc sai sót trong lúc truy vấn.”, “Các trường hợp xây dựng lưu trữ sai có thể dẫn đến việc hao tốn thời gian trong lúc truy vấn cũng như trong lúc xây dựng chương trình”. Đó cũng chính là những trở ngại mà chúng tôi gặp phải trong quá trình thực hiện đề tài.

### **5. Mục tiêu bài toán**

Dựa trên cơ sở kiến thức về phương pháp VSM và mô hình BIM, đề tài đưa ra định hướng xây dựng một mô hình tách term và thực hiện truy vấn dựa trên những term đã được tách. Từ các câu, đoạn văn hoặc là một tệp các văn bản cho trước. Mô hình được xây dựng trong đề tài này sẽ trả về một kết quả được đánh giá dựa trên sự liên quan giữa câu truy vấn và cả những văn bản hoặc các câu đã được cho ở trên. Và để đám bảo hơn về sự chính xác khi xét về chất lượng và độ chính xác của kết quả trả về dựa trên việc đánh giá mô hình thông qua các phương pháp khác nhau như accuracy, precision và recall.

Thông qua mô hình được xây dựng trong đề tài này, nhóm hi vọng rằng sẽ mang lại những nghiên cứu cơ bản trong việc truy vấn thông tin từ đó có thể phát triển thành những mô hình phát triển hơn.

# **Chương 2. CƠ SỞ LÝ THUYẾT**

## **I. Tiền xử lý dữ liệu**

Trong giai đoạn tiền xử lý, hệ thống IR tạo ra biểu diễn bên trong của mục. Trước hết, tập tài liệu văn bản thông tin trong từng tài liệu thông qua quy trình đánh chỉ được tiền xử lý bằng một số phương pháp thao tác văn bản tự động như phân tích từ vựng (lexical analysis), loại bỏ từ dừng (stopword removing), lấy gốc từ (stemming) từ dạng văn bản đơn giản (plain text) của tài liệu. Kết quả nhận được là tập các từ (term) hay còn được hiểu là các khái niệm (concept), được coi là khung nhìn logic của tài liệu.

Một term có thể là một từ (word), nhưng cũng có thể là một gốc từ (stem), một cụm danh từ hoặc một cụm từ (phrase). Gốc từ là một từ được rút gọn thành gốc của nó sau khi loại bỏ các phụ tố: ví dụ, ‘system2” và ‘component\_123’sẽ trở thành ‘system’ và ‘component’ sau bước lấy gốc từ. Giả thiết cơ bản (đôi khi bị nghi ngờ) nằm sau việc lấy gốc từ là không có sự khác biệt đáng kể về ý nghĩa giữa các từ có chung một gốc. Một cụm từ chứa

ít nhất hai từ liên tiếp có nghĩa rõ ràng, ví dụ ‘office application’ hoặc “Hanoi University of Technology”. Nếu có thể, những từ khóa (keyword) được chỉ định một cách thủ công, mô tả nội dung của tài liệu cũng có thể được dùng cho việc lập chỉ mục (ví dụ Google).

1. **Phân tích từ vựng**

Là quá trình biến đổi các ký tự trong tài liệu thành một tập các từ được đề cử để chọn làm từ chỉ mục bằng cách xử lý các chữ số, dấu nối, các ký hiệu chấm câu và chữ viết hoa viết thường.

Bước đầu tiên là lọc ra các ký tự không mong muốn và ký hiệu (các thẻ HTML, dấu chấm câu, các con số...). Tiếp theo, văn bản cần được chia thành các thẻ (token, còn gọi là từ khóa) sử dụng khoảng trắng phân tách và các ký tự kết thúc câu. Việc này không đơn giản vì các từ trong các văn bản không phải lúc nào cũng được phân tách rõ ràng (ví dụ, nếu văn bản là I can’t go thì có thể xem dấu nháy là một dấu phân tách từ để có hai từ là can và t, hoặc có thể không coi nó là dấu phân tách và xem là một từ can’t, hoặc có thể mở rộng dạng liên kết thành hai từ can và not và sử dụng khoảng trắng để phân tách.

1. **Loại bỏ từ dừng**

Việc loại bỏ từ dừng có ý nghĩa làm giảm kích cỡ của cấu trúc chỉ mục. Đây là bước tiền xử lý nhằm loại bỏ những từ có tần suất xuất hiện cao trong hầu hết các tài liệu mà lại không mang nội dung có ý nghĩa. Những từ như vậy được gọi là từ dừng (stopword), bao gồm các mạo từ, giới từ, liên từ, chẳng hạn như a, the. It, of, could...

·

Quá trình loại bỏ từ dừng có thể được chia thành hai loại :

Từ cần loại bỏ nằm trong danh sách từ dừng, quá trình này sẽ được thực hiện trong phần nhận dạng từ, có nghĩa là từ này sẽ không thể đi

qua bộ nhận dạng từ và vì thế chúng không được lập chỉ mục. Từ cần loại bỏ không nằm trong danh sách từ dùng nhưng nó xảy ra quá thường xuyên trong tập tài liệu của ta (từ quá thường xuyên ở đây có nghĩa là nó xuất hiện vượt quá một ngưỡng qui định của ta, ví dụ như có mặt trên 80% số lượng File, hoặc trên 200 file), quá trình này được thực hiện sau khi ta đã lập chỉ mục song toàn bộ tài liệu và bảng chỉ mục đang được lưu trong bộ đệm bộ nhớ chính. Khi đó ta thực hiện việc duyệt trên bảng băm để tìm các từ dừng, thêm chúng vào

danh sách danh sách từ dừng và loại phần tử chứa từ đó khỏi bảng

băm.

1. **Lấy gốc từ**

Lấy gốc từ là quá trình thu gọn một từ về dạng ngữ pháp gốc của nó. Nhằm xác định và nhóm các từ có chung ngữ nghĩa sao cho người dùng không phải xác định quá cụ thể trong truy vấn. Ví dụ: computes, computing, computer có gốc là comput.

*Việc lấy gốc từ trước khi xây dựng chỉ mục có ưu điểm là làm giảm kích thước chỉ mục và cho phép truy xuất các tài liệu với nhiều dạng biến tố của cùng một từ (ví dụ, khi tìm kiếm với từ computation, kết quả sẽ bao gồm các tài liệu có chứa từ computations và computing).*

Như vậy, giai đoạn tiền xử lý tài liệu tập trung vào việc chắt lọc tập các khái niệm mô tả cho từng tài liệu. Các khái niệm đó thường được gán trọng số

riêng, thể hiện độ liên quan của chúng đối với chủ đề của tài liệu. Thông thường, nếu một term xuất hiện càng nhiều lần trong một tài liệu và ít xuất hiện hơn trong các tài liệu khác của tập hợp thì nó càng là ký hiệu mô tả tốt hơn cho tài liệu đó. Những tiêu chí khác như vị trí của từ khóa, phần logic mà từ đó nó đã được chắt lọc hoặc độ dài của tài liệu có thể được dùng để tính toán trọng số của khái niệm liên quan trong tài liệu.

3.1. Lemmatization

Lemmatization là quá trình nhóm các dạng biến cách khác nhau của một từ lại với nhau để chúng có thể được phân tích thành một mục duy nhất. Lemmatization tương tự như từ gốc nhưng nó mang lại ngữ cảnh cho các từ. Vì vậy, nó liên kết các từ có nghĩa tương tự thành một từ.

Tiền xử lý văn bản bao gồm cả Từ gốc cũng như Từ vựng. Nhiều khi mọi người thấy hai thuật ngữ này khó hiểu. Một số coi hai điều này là như nhau. Trên thực tế, từ vựng hóa được ưa thích hơn Stemming vì từ vựng hóa thực hiện phân tích hình thái học của các từ.

Các ứng dụng của từ vựng là:

Được sử dụng trong các hệ thống truy xuất toàn diện như công cụ tìm kiếm.

Được sử dụng trong lập chỉ mục nhỏ gọn

Ví dụ về từ vựng:

-> rocks : rock

-> corpora : corpus

-> better : good

3.2. Stemming

Stemming là quá trình tạo ra các biến thể hình thái của từ gốc/từ gốc. Các chương trình gốc thường được gọi là các thuật toán gốc hoặc các chương trình gốc. Thuật toán tạo gốc rút gọn các từ“chocolates”, “chocolatey”, “choco”thành từ gốc, “chocolate” and “retrieval”, “retrieved”, “retrieves” rút gọn thành từ gốc “retrieve”. Từ gốc là một phần quan trọng của quy trình pipelining trong Xử lý ngôn ngữ tự nhiên.

3.2.1 Thuật toán Stemmer của Porter

Đây là một trong những phương pháp tạo từ gốc phổ biến nhất được đề xuất vào năm 1980. Nó dựa trên ý tưởng rằng các hậu tố trong ngôn ngữ tiếng Anh được tạo thành từ sự kết hợp của các hậu tố nhỏ hơn và đơn giản hơn. Stemmer này được biết đến với tốc độ và sự đơn giản của nó. Các ứng dụng chính của Porter Stemmer bao gồm khai thác dữ liệu và truy xuất thông tin. Tuy nhiên, các ứng dụng của nó chỉ giới hạn trong các từ tiếng Anh. Ngoài ra, nhóm các nhánh được ánh xạ trên cùng một gốc và gốc đầu ra không nhất thiết phải là một từ có nghĩa. Các thuật toán về bản chất khá dài và được biết đến là thuật toán gốc lâu đời nhất.

3.2.2 Snowball Stemmer

Khi so sánh với Porter Stemmer, Snowball Stemmer cũng có thể ánh xạ các từ không phải tiếng Anh. Vì nó hỗ trợ các ngôn ngữ khác nên Snowball Stemmers có thể được gọi là một trình tạo gốc đa ngôn ngữ. Bộ tạo gốc Snowball cũng được nhập từ gói nltk. Trình tạo gốc này dựa trên ngôn ngữ lập trình có tên là 'Snowball' xử lý các chuỗi nhỏ và là trình tạo gốc được sử dụng rộng rãi nhất. Snowball Stemmer mạnh hơn nhiều so với Porter Stemmer và còn được gọi là Porter2 Stemmer. Do có những cải tiến được thêm vào khi so sánh với Porter Stemmer, Snowball Stemmer có tốc độ tính toán cao hơn.

Và còn nhiều loại stemming nữa…

## **II. Lập chỉ mục**

Bước lập chỉ mục sử dụng các term mô tả khung nhìn logic của các tài liệu để xây dựng chỉ mục. Như trên đã nêu, cấu trúc chỉ mục phổ biến nhất là Inverted Files, trong đó tập tài liệu được biến đổi thành một tập các term kèm theo một danh sách tương ứng các tài liệu mà chúng xuất hiện. Trong một Inverted File, mỗi term trỏ tới một danh sách tất cả các tài liệu mà nó xuất hiện. Cấu trúc chỉ mục như vậy đóng vai trò rất quan trọng vì nó cho phép tìm kiếm nhanh trên tập dữ liệu lớn.

Quy trình này có thể được thực hiện thủ công (đòi hỏi sức người nên

rất tốn kém) hoặc tự động bằng cách tách các term từ văn bản của phần tử thông tin, sử dụng một thủ tục đựa trên thông kê hoặc ngôn ngữ.

Một số term có thể biểu diễn tốt hơn chủ đề của tài liệu. Do đó, mỗi term có thể được gán một trọng số thể hiện tầm quan trọng của nó trong tài liệu. Như vậy, cấu trúc chỉ mục bao gồm một tập các term đã được xử lý, kèm theo một danh sách các tài liệu chứa chúng và trọng số của chúng. Trọng số của một term trong một tài liệu có thể chỉ đơn giản là số lần xuất hiện của chúng trong tài liệu. Tần số càng lớn thì tầm quan trọng của nó càng lớn. Điều này được gọi là gán trọng số theo tần số từ (term frequency weighting – tf). Số lượng tài liệu mà một term xuất hiện trong đó cũng có thể được sử dụng làm yếu tố trong việc gán trọng số. Một term xuất hiện trong càng nhiều tài liệu thì khả năng phân biệt của nó đối với càng tài liệu càng kém. 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 vă bản.

### **III. Các bước tiền xử lý dữ liệu**

- Phương pháp đơn giản nhất trong tách token mà nhóm em muốn đem đến với giảng viên đó là tách term theo từ. điển hình ta sẽ xem mỗi token là một từ, tách nhau bởi một khoảng trắng. Tuy nhiên với mỗi kí tự đặc biệt không nằm trong bảng anphabet thì ta vẫn xem nó là một token riêng.

Ví dụ:

Graphical user interface, diagram

Description automatically generated

- Sau khi thực hiện tách token ra thì với mỗi token là một kí tự đặc biệt ta sẽ xoá nó, ngược lại số ta sẽ giữ nguyên phòng cho trường hợp trong query có tồn tại số, đồng thời ta thực hiện lowercase chúng phòng cho trường hợp phân biệt viết hoa và viết thường trong lúc truy vấn.

Ví dụ:



- Sau đó ta thực hiện bỏ stopwork ( là những từ không có nghĩa quan trọng )

Ví dụ:

Graphical user interface

Description automatically generated

- Cuối cùng ta thực hiện lemmatize ( phương pháp loại bỏ đuôi biến tố và đưa về từ cơ bản nhất dựa trên 1 từ điển chung ) để loại bỏ đuôi biến tố và đưa về dạng cơ bản.

Ví dụ:

Graphical user interface, application

Description automatically generated

**III. Ngữ liệu**

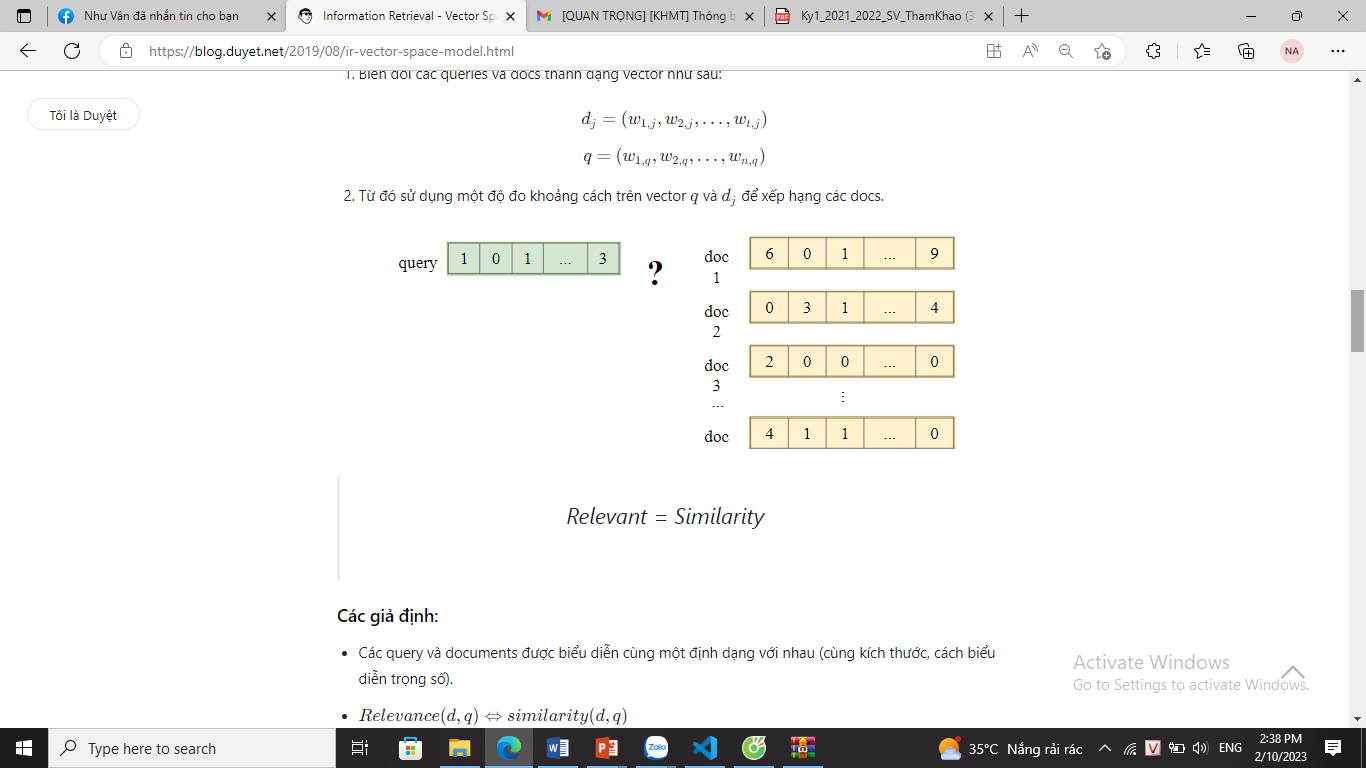
1. Tập tài liệu Cranfield dùng để tính trên 2 mô hình VSM và BIM
2. Tài liệu test\_small dùng để tính trên mo hình GVSM
3. Tài liệu TIME được thêm vào để so sánh với bộ tài liệu Cranfield

# **Chương 3. VectorSpace Model (VSM)**

# **Giới thiệu:**

* VectorSpace Model là một mô hình truy vấn thông tin trong đó văn bản được biểu diễn như một vector.
* VSM ra quyết định cho ta về những tài liệu nào truy vấn giống nhau .
* VSM cho phép kết quả tính toán được xếp hạng theo độ đo tương tự.
* Biểu diễn văn bản như những điểm trong không gian Euclid n-chiều,
* Mỗi chiều tương ứng với một term trong tập hợp các từ.
* Giá trị mỗi chiều của vector (trọng số của term) thể hiện số lần xuất hiện của term đó trong document.

1. Biến đổi các queries và docs thành dạng vector như sau
2. Từ đó sử dụng một độ đo khoảng cách trên vector *q* và *dj*​ để xếp hạng các docs.



## **1.1 Tìm trọng số (weights) cho vector**

## Cách xác định và tính weights cho vector là hết sức quan trọng, ảnh hưởng đến độ chính xác của các thuật toán xếp hạng. Việc các từ có trọng số khác nhau là do không phải các từ đều có sự quan trọng giống nhau, sử dụng số lần xuất hiện của các từ làm vector không phải là một cách tối ưu. Ở phương diện các documents, một vài từ có thể mang nhiều thông tin hơn các từ còn lại.

Có nhiều kỹ thuật tính trọng số: TF, IDF, TF-IDF, …

### a. TF: Term frequency

Từ nào xuất hiện nhiều trong câu thì quan trọng, công thức này sẽ đếm tuần suất xuất hiện các từ trong câu.

**TF normalization**: Do tùy độ dài ngắn khác nhau của từng câu, mà việc đếm tần suất các từ có thể không công bằng, một số kỹ thuật chuẩn hóa sau có thể giảm tránh được điều này,

### b. IDF: Inverse document frequency

Từ nào xuất hiện nhiều trong mọi câu thì không mang nhiều ý nghĩa (ví dụ như a, the, are, thì, là, …). Vì vậy trọng số IDF là nghịch đảo của tuần suất xuất hiện của các từ trong các documents.

### c. TF-IDF: Term frequency - Inverse document frequency

Phép nhân giữa TD và IDF cho phép ta kết hợp cả 2 độ đo trên, từ vừa xuất hiện nhiều lần trong câu, vừa không phải là từ phổ biến xuất hiện trong mọi câu.

## **1.2. Công thức xếp hạng:**

**1.2.1. Euclidean Distance**

Chúng ta có thể xác định khoảng cách giữa hai vectơ bằng phép đo này.

p,q là 2 điểm trong không gian Euclidean,

Nhược điểm: Docs dài hơn sẽ bị giảm hạn do chứa nhiều từ khác gây nhiễu, khiến khoảng cách xa hơn.

**1.2.2 Cosine Similarity**

Về cơ bản, nó sử dụng cosin của góc giữa hai vectơ. Dựa vào đó cho ta biết hai vectơ có gần nhau hay không

Cosine similarity =

**II. Lập chỉ mục**

Cho các tài liệu sau

d1) ship ocean wood d2) boat ocean

d3) ship d4) wood tree

d5) wood d6) tree

Yêu cầu:

* Lập chỉ mục cho tập tài liệu trên
* Xử lý truy vấn theo mô hình vector với trọng số tf.idf cho các truy vấn sau: Q = ship wood

|  |  |  |
| --- | --- | --- |
| Word | IDDoc & Frequency | Quantity doc |
| boat | (2,1) | 1 |
| ocean | (1,1); (2,1) | 2 |
| ship | (1,1);(3,1) | 2 |
| tree | (4,1);(6,1) | 2 |
| wood | (1,1);(4,1);(5,1) | 3 |

Chuẩn hóa theo công thức

|  |  |
| --- | --- |
| Word | Chuẩn hóa TF cho từng IDDoc |
| boat | (2,1) |
| ocean | (1,1); (2,1) |
| ship | (1,1);(3,1) |
| tree | (4,1);(6,1) |
| wood | (1,1);(4,1);(5,1) |

Chuẩn hóa IDF theo công thức

|  |  |  |
| --- | --- | --- |
| Word | Chuẩn hóa IDF | Quantity doc |
| boat | 0.7781 | 1 |
| ocean | 0.4771 | 2 |
| ship | 0.4771 | 2 |
| tree | 0.4771 | 2 |
| wood | 0.301 | 3 |

1. **Quá trình truy vấn**

Tính trọng số từng term

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Q | d1 | d2 | d3 | d4 | d5 | d6 |
| boat | 0 | 0 | 0.7781 | 0 | 0 | 0 | 0 |
| ocean | 0 | 0.4771 | 0.4771 | 0 | 0 | 0 | 0 |
| ship | 0.4771 | 0.4771 | 0 | 0.4771 | 0 | 0 | 0 |
| tree | 0 | 0 | 0 | 0 | 0.4771 | 0 | 0.4771 |
| wood | 0.4771 | 0.301 | 0 | 0 | 0.4771 | 0.4771 | 0 |

= (0, 0.4771, 0.4771,0, 0.301) =(0.7781,0.4771,0,0,0)

=(0,0, 0.4771,0,0) =(0,0,0,0.4771, 0.4771)

=(0,0,0,0, 0.4771) =(0,0,0, 0.4771,0)

Áp dụng chuẩn hóa cho vector tài liệu , :

Áp dụng tính độ tương đồng bằng công thức cosine similarity:

Cos(, Cos(, Cos(, Cos( Cos(,

Cos(

1. **Generalized vector spaces model**

**4.1 . Giới thiệu**

* Giới thiệu vào năm 1985, nhóm tác giả Wong.
* GVSM giới thiệu các mối tương quan giữa các term điều này phản đối giả định trực giao theo cặp, sử dụng co-occurrences và mô hình trong WordNet, một cơ sở dữ liệu từ vựng lớn gồm các từ được nhóm thành các tập hợp từ đồng nghĩa.
* Trong đó mỗi vectơ term được biểu thị dưới dạng tổ hợp tuyến tính của 2^n vector minterm(r) ( r=1,2,...2^t), với các term là k1,k2,...kt

m1 = (1,0,...0) … m2^t = (0,0,...,1)

minterm này thể hiện sự xuất hiện của từ trong document nào, ví dụ m1 sẽ có document 1 chưa term k1.

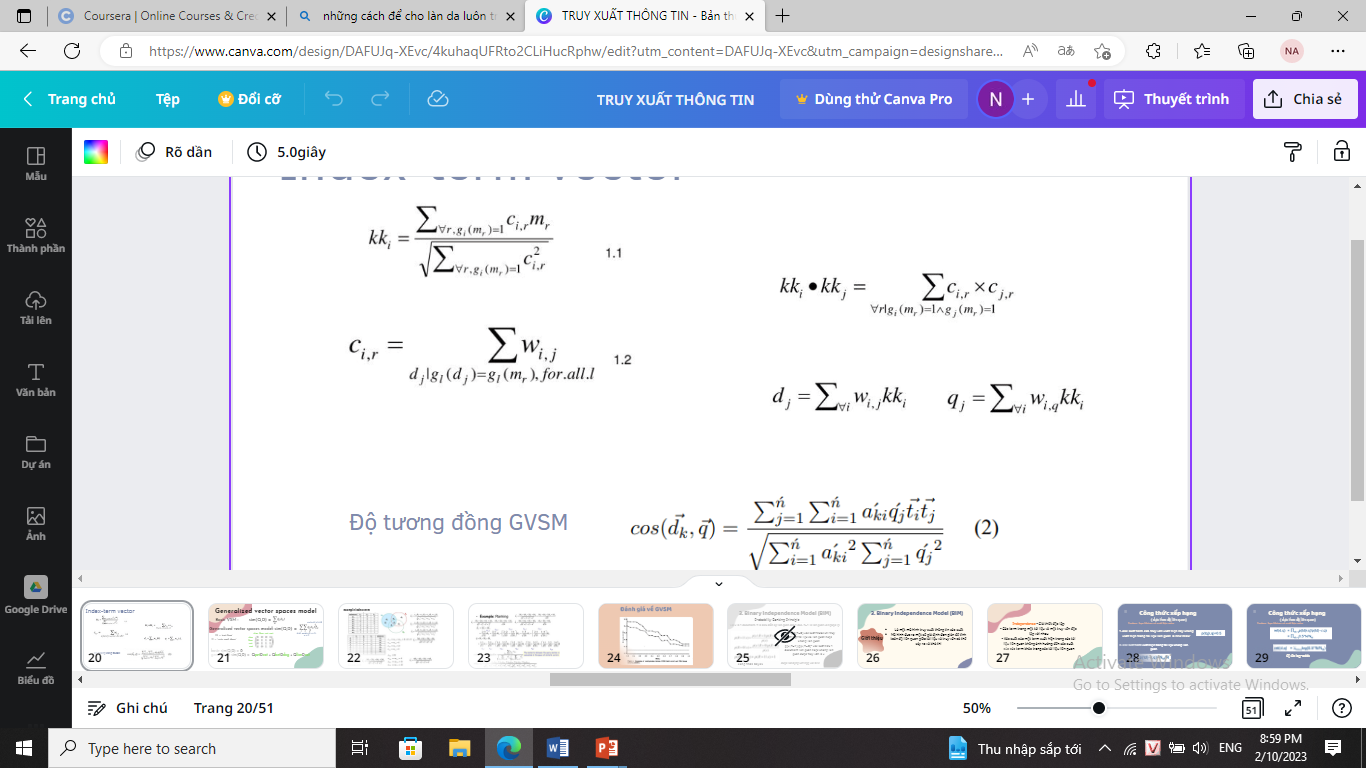
So sánh : Basic VSM : sim(Q,D) =

Generalized vector spaces model: sim(Q,D) =

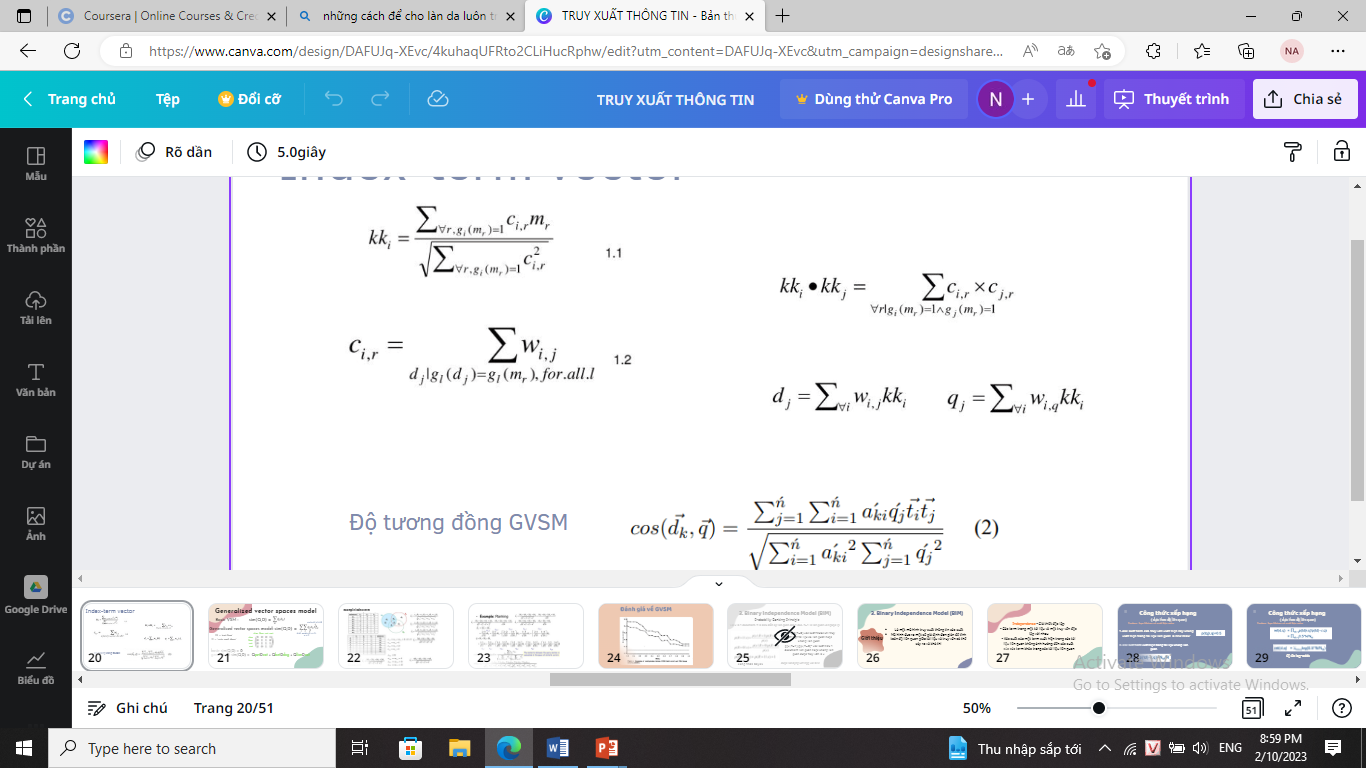
\* Với Q,D là truy vấn và tài liệu v,w là terrm trong Q,D, là độ tương đồng giữa Q,D. Generalized VSM sẽ có thêm trong công thức tính độ tương đồng

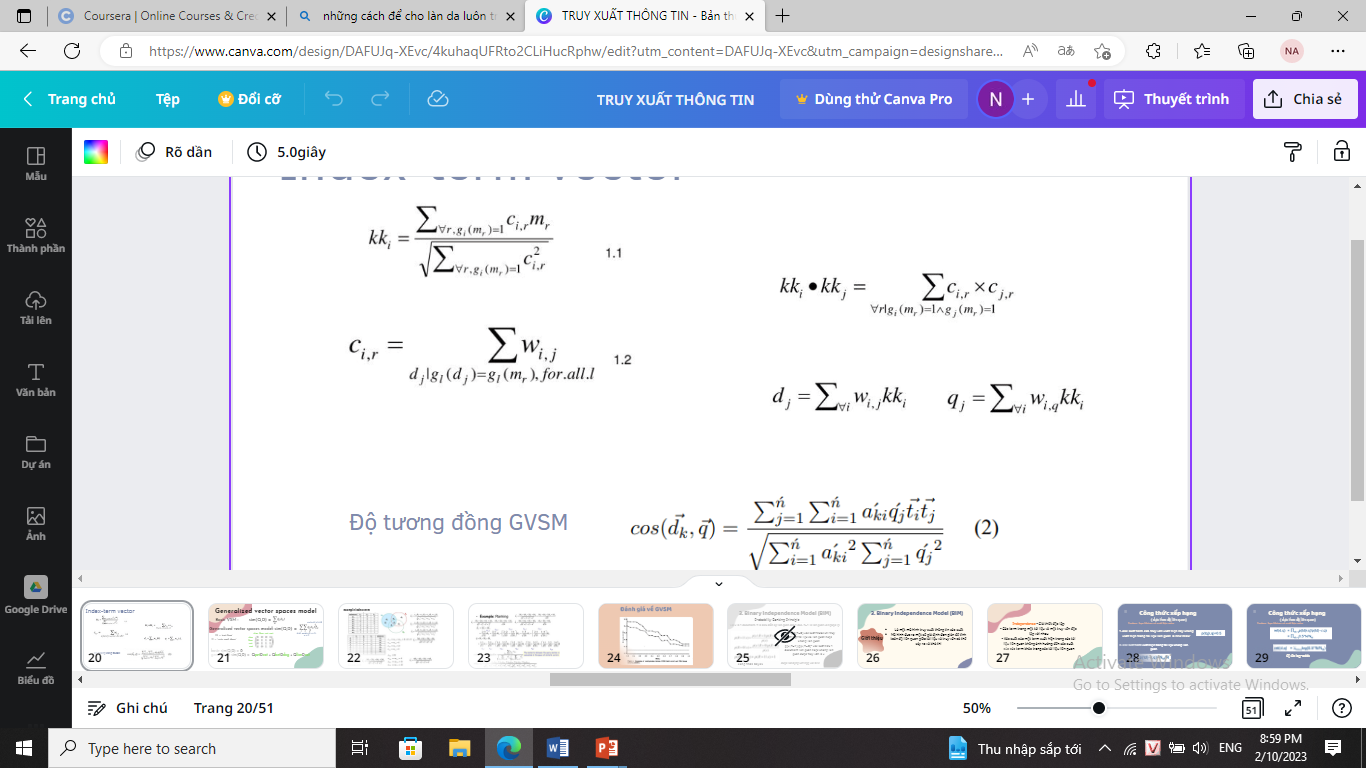
4.2 Các bước tiến hành

* Có 2 thành phần mới là và với được tính theo và min term . là tổng trọng số của term i trong tài liệu j ứng với mintern

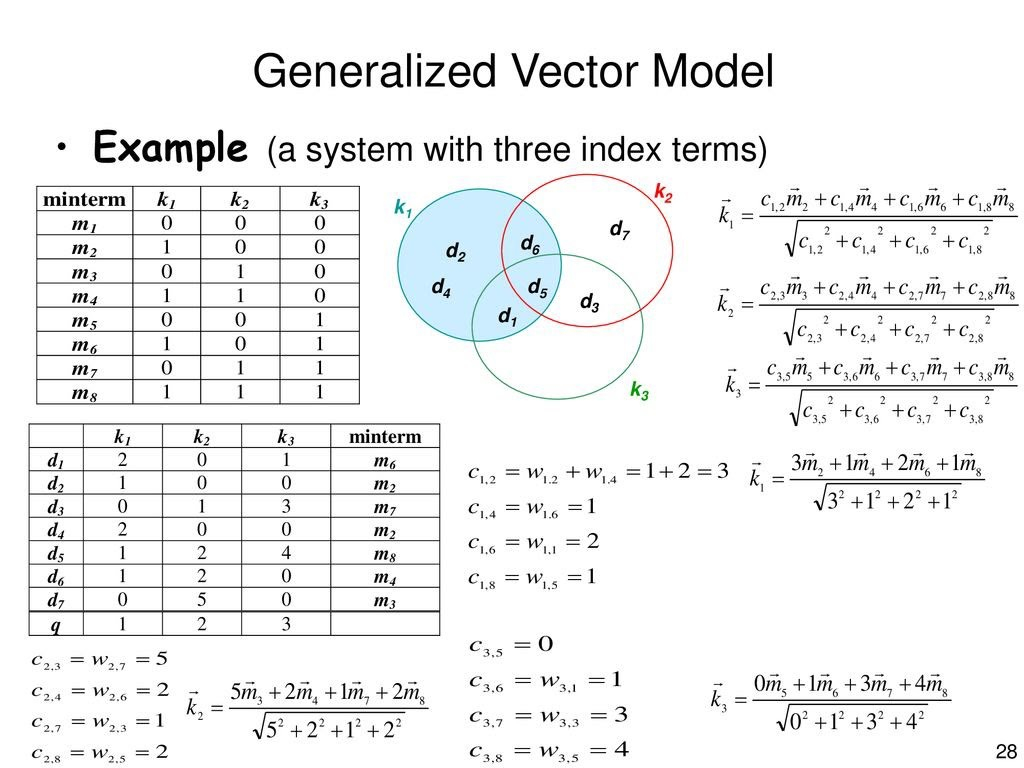


* Sau đó ta có công thức của dj và qj và có công thức tính độ tương đồng mới

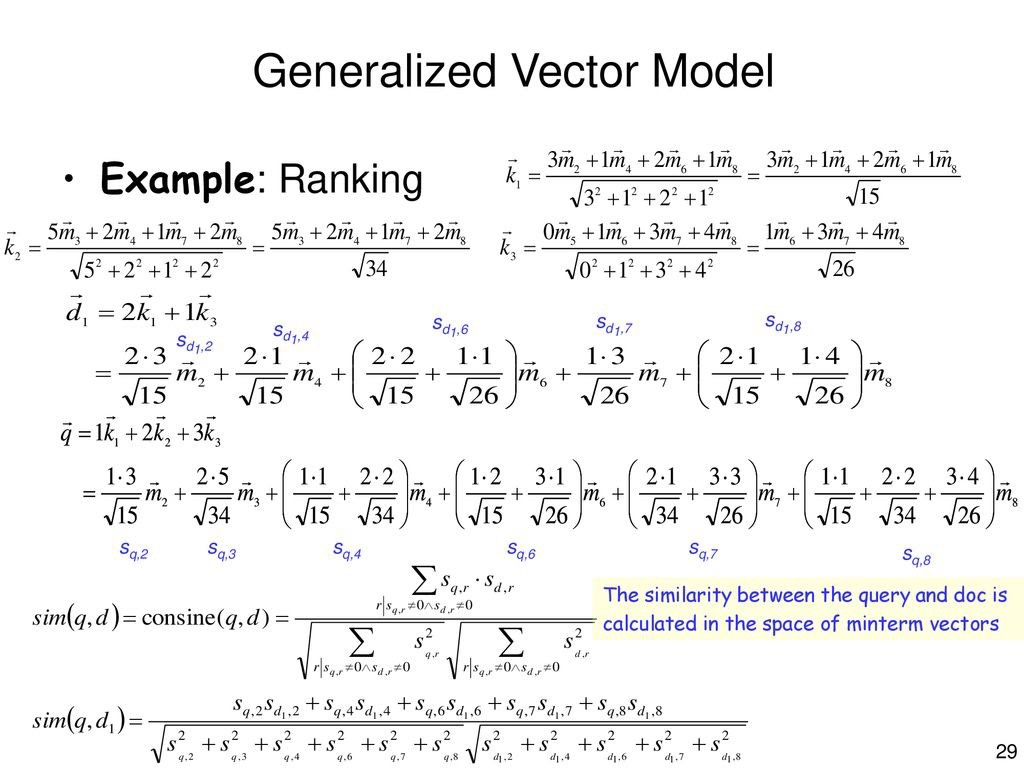




* Một ví dụ khi cho 6 bộ tài liệu d1,d2,..d6 và câu truy vấn q, ta có trọng số của term được tính theo tần số

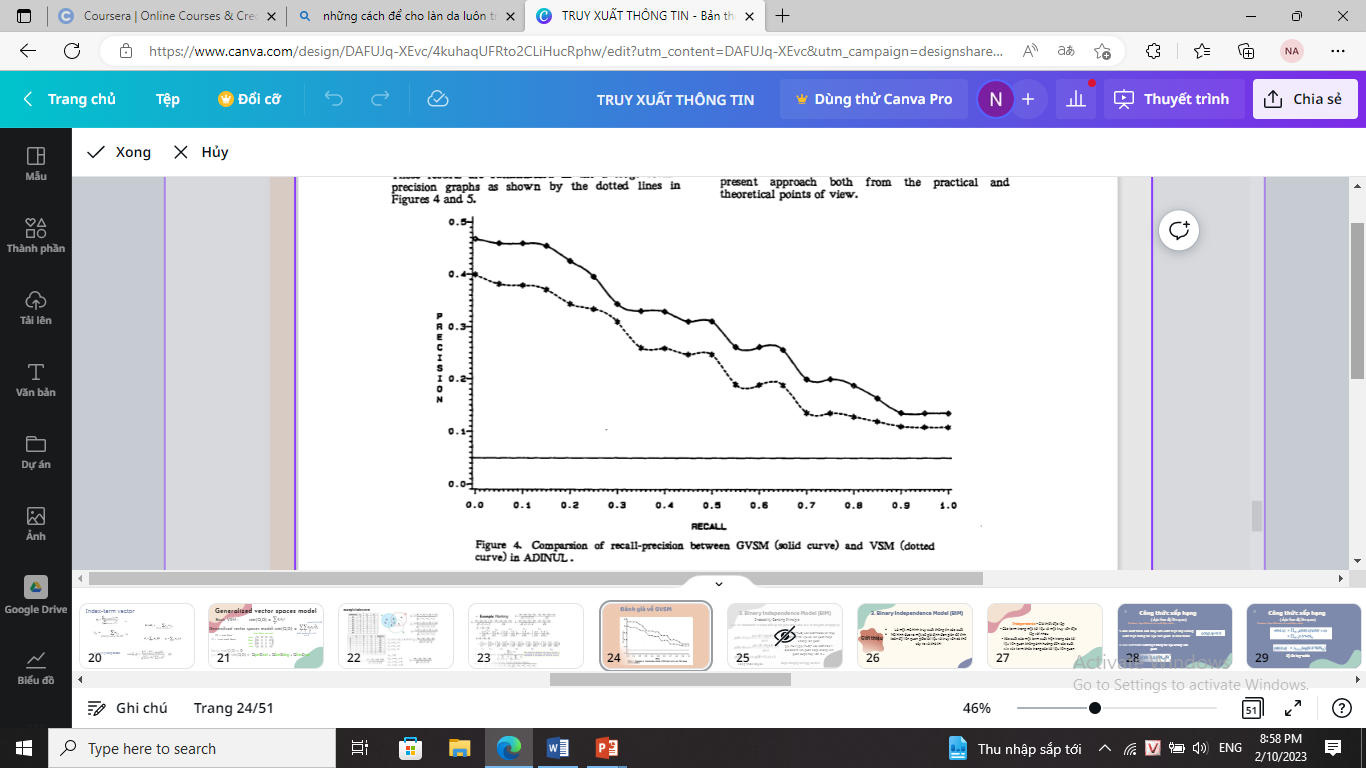


Sau đó được xếp hạng như sau:



**Đánh giá GVSM :**

* Recall-precision cao hơn so với VSM
* Chi phí tốn kém hơn. Mô hình chỉ áp dụng được với bộ data nhỏ.



##### Chương 4. BINARY INDEPENDENT model

1. **Giới thiệu**
2. Giới thiệu và công thức

* Là một mô hình truy xuất thông tin xác xuất
* Mô hình đưa ra một số giả định đơn giản để tính toán độ liên quan giữa tài liệu và truy vấn có thể xảy ra và khả thi

**Independence**

- Giả thiết độc lập:

- Các term trong một tài liệu và một truy vấn độc lập với nhau

- Xác suất của một term xuất hiện trong các tài liệu liên quan không ảnh hưởng đến xác suất của các term khác trong các tài liệu liên quan

Truy vấn: vectơ tỷ lệ thuật ngữ nhị phân

Đưa ra truy vấn q, đối với mỗi tài liệu d cần tính p(R|q,d). thay thế bằng phép tính p(R|q,x) trong đó x là véc tơ tần suất số hạng nhị phân đại diện cho d Chỉ quan tâm đến xếp hạng

Sẽ sử dụng odds và Quy tắc Bayes:



Suy ra 

Với  là hằng số cho truy vấn, còn  là thứ cần phải ước lượng

Ta phân tích thành phần cần ước lượng



Vì vậy 



Vì *xi*  có giá trị là 0 hoặc 1:





Viết lại:



Ta sử dụng  để xếp hạng





Nếu các tài liệu không liên quan được xấp xỉ bằng toàn bộ bộ sưu tập, thì ri (xác suất xuất hiện trong các tài liệu không liên quan cho truy vấn) là n/N và

log (1– ri)/ri = log (N– n)/n ≈ log N/n = IDF!

1. Xác suất term của truy vấn xuất hiện hay không xuất hiện trong tài liệu liên quan là như nhau:
2. Relevant Feedback
   1. Giới thiệu

* User đánh dấu những document là liên quan hoặc không liên quan, truy xuất thông tin sẽ đánh giá lại để xay dựng model tốt hơn
* reweigting thông qua các công thức xác suất
* Pi=|Vi|/|V|
* ri= (ni-|Vi|)/(N-|V|)
* pi là xác suất những tài liệu liên quan và ri là không liên quan. V là những tài liệu liên quan, N-V là những tài liệu không liên quan.

3.2 Ta gọi R liên quan, NR không liên quan. Các bước RF như sau:

* Đoán một mô tả xác suất sơ bộ của R và sử dụng nó để truy xuất bộ tài liệu đầu tiên V.
* Tương tác với người dùng để tinh chỉnh mô tả: tìm hiểu một số thành viên xác định của R và NR
* Ước tính lại pi và ri trên cơ sở

Có thể kết hợp thông tin mới với dự đoán ban đầu (dùng Bayesian trước):

, k là prior weight, trong chương trình em cài đặt là k=1

* Lặp lại, do đó tạo ra một loạt các xấp xỉ cho R.

1. **Lập chỉ mục**

Với N: tổng số tài liệu

Ntd: số lượng tài liệu chứa term

Cho các tài liệu sau

d1) ship ocean wood d2) boat ocean

d3) ship d4) tree

d5) wood d6) tree

Yêu cầu:

* Lập chỉ mục cho tập tài liệu trên
* Xử lý truy vấn theo mô hình vector với trọng số tf.idf cho các truy vấn sau: Q = ship wood

## **Quá trình truy vấn**

|  |  |  |
| --- | --- | --- |
| Word | IDDoc & Frequency | Quantity doc |
| boat | (2,1) | 1 |
| ocean | (1,1); (2,1) | 2 |
| ship | (1,1);(3,1) | 2 |
| tree | (4,1);(6,1) | 2 |
| wood | (1,1);(5,1) | 2 |

Chuẩn hóa trọng số theo công thức

|  |  |  |
| --- | --- | --- |
| Word | Chuẩn hóa trọng số từng term | Quantity doc |
| boat | 1.585 | 1 |
| ocean | 0.585 | 2 |
| ship | 0.585 | 2 |
| tree | 0.585 | 2 |
| wood | 0.585 | 2 |

Tính trọng số từng term

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Q | d1 | d2 | d3 | d4 | d5 | d6 |
| boat | 0 | 0 | 1.585 | 0 | 0 | 0 | 0 |
| ocean | 0 | 0.585 | 0.585 | 0 | 0 | 0 | 0 |
| ship | 0.585 | 0.585 | 0 | 0.585 | 0 | 0 | 0 |
| tree | 0 | 0 | 0 | 0 | 0.585 | 0 | 0.585 |
| wood | 0.585 | 0.585 | 0 | 0 | 0 | 0.585 | 0 |

Áp dụng tính độ tương đồng bằng công thức RVS :

rel(, rel(,

rel(, rel(,

rel( rel(

Xếp hạng d3=d4=d5=d6>d1>d2

ƯU NHƯỢC ĐIỂM 2 mô hình

1. VECTOR SPACE MODEL:

a, Ưu điểm:

* Dễ hiểu và dễ cài đặt
* Đã được nghiên cứu từ lâu và thực nghiệm cho kết quả tốt và khả thi
* Hiện tại là phương pháp được sử dụng rộng rãi
* Có nhiều công thức tính TF.IDF khác nhau.

b, Nhược điểm:

* Các chiều trong không gian là mỗi từ khóa, không đảm bảo độc lập về ngữ nghĩa.
* Các vector chỉ nằm trong phần dương của không gian.
* Số lượng chiều rất lớn, trong đó có những từ khóa có thể không cần thiết phải biểu diễn (những từ nhiễu)

1. BIM:

Ưu điểm:

* Ta có thể phân cấp tài liệu một cách dễ dàng.
* Có cấu trúc tương đối giống với mô hình Vector Space. Dễ xây dựng.

Nhược điểm:

* Kết quả trả về có thể có thể là những tài liệu trùng lặp hoặc gần trùng lặp.

# **Chương 5. CÁC PHƯƠNG PHÁP ĐÁNH GIÁ**

I. Recall và precision:

Trong bài toán phân lớp, để tìm được recall và precision do thì trước hết cần phải có các tham số đánh giá là: True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN):

* True positive (TP): Kết quả dự đoán đúng và kết quả làm chuẩn đúng.
* False positive (FP): Kết quả dự đoán đúng và kết quả làm chuẩn sai.
* True negative (TN): Kết quả dự đoán sai và kết quả làm chuẩn sai.
* False negative (FN): Kết quả dự đoán sai và kết quả làm chuẩn đúng.

Từ các tham số trên, ta có định nghĩa:

* Precision: Xác suất trường hợp positive trong tổng số trường hợp positive được dự đoán. Ở đây Mẫu số là tổng các trường hợp positive được dự đoán trong mô hình từ toàn bộ tập dữ liệu đã cho. Precision có thể được xác định nôm na là "mô hình đúng bao nhiêu khi nó là đúng". Precision được tính như sau:

Precision =

* Recall: Xác suất trường hợp positive trong tổng số trường hợp thực sự là positive. Do đó, mẫu số (TP + FN) là số lượng thực tế các trường hợp positive có trong tập dữ liệu. Có thể hiểu recall như là "mô hình đã bỏ lỡ bao nhiêu cái đúng khi nó hiển thị cái đúng".

Recall =

Precision và Recall trung bình sẽ được tính như sau:

II. Độ chính xác trung bình không nội suy (Average Pecision)

Tính theo từng truy vấn

Bằng độ chính xác trung bình tại độ phủ khác nhau của truy vấn:

III. Độ chính xác kỳ vọng (Mean Average Precision)

Tính trên tất cả truy vấn

Bằng giá trị trung bình của độ chính xác trung bình từng truy vấn:

Ví dụ cho bài toán đánh giá theo precision, recall, Map, và Map nội suy.

Cho kết quả của 2 truy vấn đã xếp hạng trên dữ liệu thử nghiệm gồm 10000 tài liệu trong đó truy vấn 1 và 2 lần lượt có 20 và 12 tài liệu liên quan (R: liên quan, N: không liên quan):

1: R R N N N N N N R N R N N N R N N N N R

2: R N R R N R R N N N R N N R R R R N N N

Tính MAP

|  |  |  |  |
| --- | --- | --- | --- |
| query1 | | | |
| n | re | pre | AP |
| 1 | (1/20)=0.05 | (1/1)=1.00 | (1+1+0.33+0.36+0.33+0.30)/6=0.56 |
| 2 | (2/20)=0.10 | (2/2)=1.00 |
| 3 | (3/20)=0.15 | (3/9)=0.33 |
| 4 | (4/20)=0.20 | (4/11)=0.36 |
| 5 | (5/20)=0.25 | (5/15)=0.33 |
| 6 | (6/20)0.30 | (6/20)=0.30 |

|  |  |  |  |
| --- | --- | --- | --- |
| query2 | | | |
| n | re | pre | AP |
| 1 | (1/12)=0.08 | (1/1)=1.00 | 0.65 |
| 2 | (2/12)=0.17 | (2/3)=0.67 |
| 3 | (3/12)=0.25 | (3/4)=0.75 |
| 4 | (4/12)=0.33 | (4/6)=0.67 |
| 5 | (5/12)=0.42 | (5/7)=0.71 |
| 6 | (6/12)=0.50 | (6/11)=0.55 |
| 7 | (7/12)=0.58 | (7/14)=0.50 |
| 8 | (8/12)=0.67 | (8/15)=0.53 |
| 9 | (9/12)=0.75 | (9/16)=0.56 |
| 10 | (10/12)=0.83 | (10/17)=0.59 |

Độ đo MAP= (0.56+0.65)/2=0.6

Bài 2) tính P tại từng mức k có thay đổi độ phủ, sau đó nội suy ra P tại 11 điểm theo độ phủ và tính trung bình cộng của các P này. Giá trị này là AP. Nếu có nhiều truy vấn, các bạn tính AP trung bình từ kết quả của các truy vấn sẽ được MAP.

R R N N R N R N R N R N N R R N R R N R

Tính P tại các điểm độ phủ:

R = 1/15, P = 1/1

R = 2/15, P = 2/2

R = 3/15, P = 3/5

R = 4/15, P = 4/6

R = 5/15, P = 5/9

R = 6/15, P = 6/11

R = 7/15, P = 7/14

R = 8/15, P = 8/15

R = 9/15, P = 9/17

R = 10/15, P = 10/18

R = 11/15, P = 11/20

Nội suy:

|  |  |  |
| --- | --- | --- |
| 0 |  | 1 |
| 0.1 |  | 1 |
| 0.2 |  | 0.67 |
| 0.3 |  | 0.56 |
| 0.4 |  | 0.56 |
| 0.5 |  | 0.56 |
| 0.6 |  | 0.56 |
| 0.7 |  | 0.55 |
| 0.8 |  | 0 |
| 0.9 |  | 0 |
| 1.0 |  | 0 |

Độ chính xác trung bình nội suy:

AP = 0.4964

******Kết quả thực nghiệm**

### Sau khi thực hiện 2 mô hình VSM và BIM thì nhóm có mang đến một mô hình nữa là mô hình WHOOSH:

* Nhóm em có làm thêm 2 phương pháp của Stemming nhưng do google colab không chạy được nên không thể ghi kết quả được
* Từ đó có được bảng đánh giá khách quan:

Trước hết là bảng thông tin về bộ dữ liệu Cranfield:

|  |  |  |  |
| --- | --- | --- | --- |
|  | VSM | BIM | WHOOSH |
| Tổng số term sau khi rút gọn | 6925 | 6925 | 6925 |
| Tổng số tài liệu trả về | 182492 | 193222 | 193222 |
| Precision | 0.0093 | 0.0088 | 0.0077 |
| Recall | 0.922 | 0.922 | 0.934 |
| Map | 0.357 | 0.337 | 0.258 |
| Mapr | 0.333 | 0.319 | 0.248 |

Đánh giá 3 mô hình trên bộ dữ liệu đó

|  |  |  |  |
| --- | --- | --- | --- |
|  | VSM | BIM | WHOOSH |
| Precision | 0.041 | 0.039 | 0.030 |
| Recall | 0.941 | 0.941 | 0.95 |
| Map | 0.53 | 0.47 | 0.37 |
| Mapr | 0.49 | 0.43 | 0.37 |

# Từ bảng trên ta có được biểu đồ:

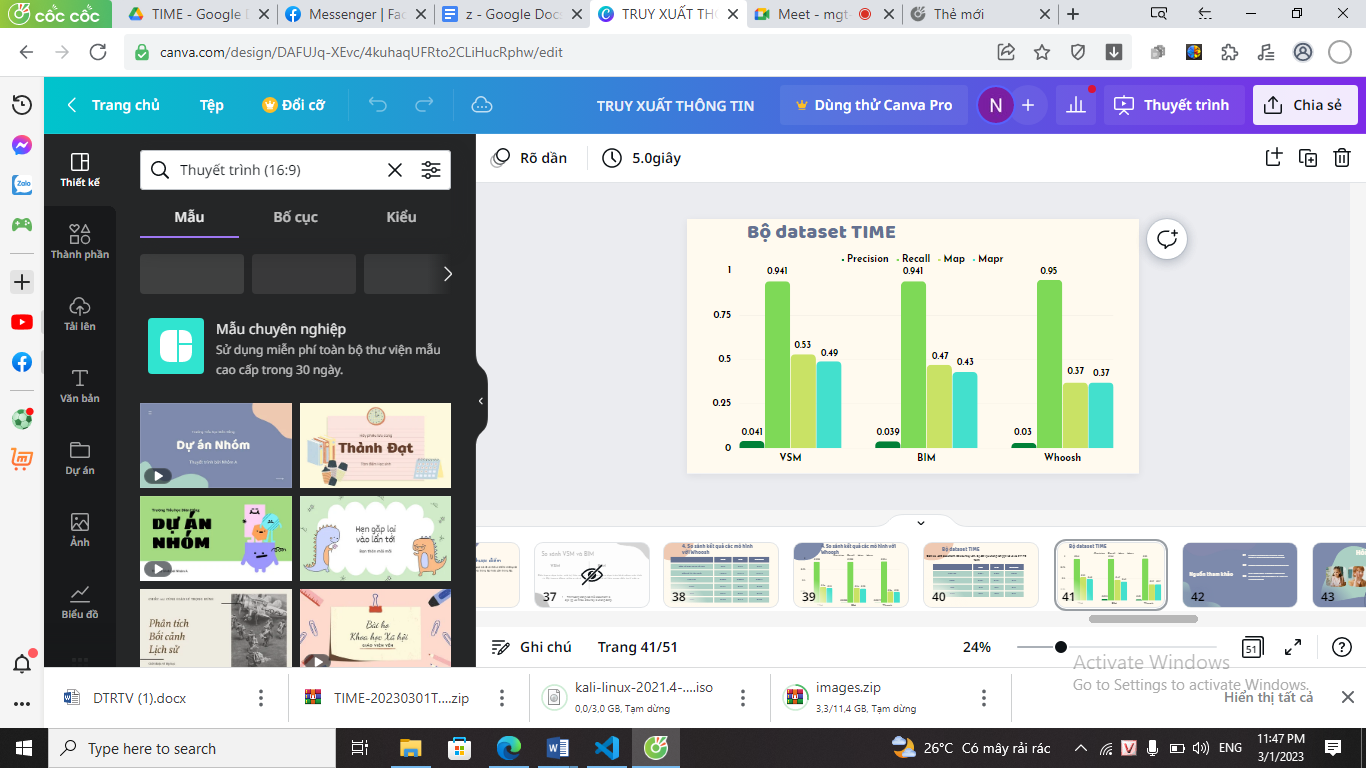
Bộ dữ liệu TIME

|  |  |  |  |
| --- | --- | --- | --- |
|  | VSM | BIM | WHOOSH |
| Precision | 0.041 | 0.039 | 0.030 |
| Recall | 0.941 | 0.941 | 0.95 |
| Map | 0.53 | 0.47 | 0.37 |
| Mapr | 0.49 | 0.43 | 0.37 |

Đánh giá

Bộ dữ liệu TIME có nhiều thông tin về thời gian

Đánh giá về các độ đo MAP,precision cũng cao hơn



**Đánh giá và nhận xét**

Mô hìnhVSM có độ đo cao nhất với độ đo precison, recall, MAP và MAP nội suy sau đó đến mô hình BIM và cuối cùng là Whoosh

# **Chương 6. CÀI ĐẶT THỬ NGHIỆM**

### **1. Thiết kế chương trình cài đặt**

**Ngôn ngữ lập trình**

Python ra đời năm 1991, và là một ngôn ngữ thông dịch. Trải qua gần 30 năm phát triển, Python là một trong những ngôn ngữ được sử dụng nhiều nhất trong dạy lập trình và nghiên cứu khoa học. Rất nhiều trường đại học sử dụng Python để dạy về lập trình cho các sinh viên ngành Khoa Học Máy Tính. Rất nhiều công ty lớn sử dụng Python để xây dựng hệ thống như Google, Youtube, Instagram, Dropbox, Atlassian... Python là một ngữ sử dụng được cho nhiều mô hình lập trình, đơn giản khi học và sử dụng. Và đặc biệt là nó cung cấp cho các lập trình viên rất nhiều thư viện tuyệt vời về máy học, thị giác máy tính, xử lý ngôn ngữ tự nhiên,… như Scikitlearn, Tensorflow, OpenCV và đặc biệt là nltk (The Natural Language Toolkit) – một package của Python về Xử lý ngôn ngữ tự nhiên. Vì thế, nhóm đã chọn Python làm ngôn ngữ để cài đặt chương trình thực nghiệm.

**Môi trường lập trình**

Jupyter Notebook là một ứng dụng web mã nguồn mở cho phép chạy Interactive Python (hay IPython), bạn có thể đưa cả code Python và các thành phần văn bản phức tạp như hình ảnh, công thức, video, biểu thức... vào trong cùng một file giúp cho việc trình bày trở lên dễ hiểu, giống như một file trình chiếu nhưng lại có thể thực hiện chạy code tương tác trên đó, cốt lõi của việc này chính là Markdown. Các file "notebook" này có thể được chia sẻ với mọi người và có thể thực hiện lại các công đoạn một cách nhanh chóng và chính xác như những gì bạn đã làm trong quá trình tạo ra file.

Google colab hay Colaboratory là một sản phẩm miễn phí do Google nghiên cứu và dựa trên Jupyter. Colab là một công cụ tuyệt vời cho cả người mới bắt đầu và người dùng nâng cao. Hầu hết tất cả các thư viện quan trọng đều được cài đặt sẵn, vì vậy ta không cần phải cài đặt từng cái một. Các file của Colab được lưu trữ trong google drive của người dùng, vì vậy ai cũng có thể truy cập chúng từ bất kỳ đâu. Google colab cũng cho phép chia sẻ file với người dùng khác mà không cần tải xuống, đây được xem như là tính năng tốt nhất đối với nhiều người. Ngoài ra, sản phẩm này còn cung cấp GPU và TPU miễn phí cho công việc và điều đó làm cho Google colab trở nên lý tưởng cho các dự án học sâu và máy học. Để sử dụng Google Colab, ta không phải cài đặt bất cứ thứ gì mà có thể truy cập trực tiếp vào trang web và bắt đầu sử dụng.

Nhóm chọn Jupyter notebook để có thể thử nghiệm code dễ dàng và Google colab để thuận tiện cho việc chia sẻ file. Cả hai thực chất khá giống nhau, đều trình bày kết quả một cách trực quan hơn là dùng các IDE.

# **Chương 7. KẾT LUẬN**

### **1. Nhận xét về mô hình**

- Hai mô hình VSM và BIM tương đối giống nhau về cách cài đặt nhưng nếu ta nhìn vào bảng đánh giá trên ta có thể thấy đươc sự tương đối về cả mặt đáp án.

- Tuy nhiên về sự chặt chẽ thì VSM tương đối tốt hơn so với BIM về cả phần precision cũng như là recall.

**2. Kết luận và hướng phát triển**

2.1 Kết luận

Trong báo cáo, chúng tôi đã trình bày các bước cơ bản cần thực hiện để xây dựng hệ thống truy xuất thông tin; nghiên cứu và so sánh các kỹ thuật lập chỉ mục và các mô hình truy xuất thông tin, trình bày cơ chế lập chỉ mục Inverted Files và mô hình truy xuất thông tin dựa trên mô hình không gian véctơ của thư viện.

Chúng tôi đã trình bày các bước của quá trình truy xuất thông tin, bao gồm hai giai đoạn chính là tiền xử lý tài liệu và thu thập thông tin. Chúng tôi cũng trình bày các phương pháp đánh giá hiệu quả truy xuất theo hai hướng chính là tiếp cận hướng mô hình và tiếp cận đánh giá lấy người dùng làm trung tâm.

Chúng tôi đã trình bày cấu trúc chỉ mục Inverted Files, đã so sánh các hướng tiếp cận truy xuất thông tin là các mô hình logic, các mô hình đại số và các mô hình xác suất.

Cuối cùng, chúng tôi xây dựng chương trình thử nghiệm để kiểm nghiệm và đánh giá hiệu quả của cấu trúc chỉ mục Inverted Files và mô hình không gian véctơ. Chúng tôi đã thực hiện thử nghiệm chương trình nhiều lần với nhiều truy vấn khác nhau.

**4.2 Hướng phát triển**

Hoàn thiện chương trình tìm kiếm thông tin để có thể sử dụng trong thực tiễn như tìm kiếm thông tin, tìm kiếm thông tin trên mạng internet, tìm kiếm thông tin trên web…

Hiện tại, nhóm mới dừng lại ở việc tìm kiếm thông tin trong tập tài liệu mang nội dung là tiếng Anh. Bởi vậy, một hướng phát triển tiếp theo của đề tài là nghiên cứu kỹ thuật và mở rộng chương trình để có thể tìm kiếm tài liệu mang nội dung tiếng Việt.

Tiếng Việt có những đặc điểm riêng về cấu tạo của từ, cấu trúc ngữ pháp. Vì vậy cần có cải tiến trong việc phân tích từ vựng. Đề tài này là một bước khởi đầu để nhóm làm quen và nghiên cứu về bài toán truy xuất thông

tin.

# **TÀI LIỆU THAM KHẢO**

1. <https://machinelearningmastery.com/a-gentle-introduction-to-vector-space-models/>

[2] <https://aclanthology.org/E09-3009.pdf>

[3] <https://www.slideserve.com/adanne/information-retrieval-models>

Bảng phân công công việc

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MSSV | Họ và tên | Mô hình VSM | Mô hình BIM | Mô hình Language model | Viết báo cáo | Làm slide |
| 20520461 | Huỳnh Thái Dương |  |  |  | x | x |
| 20521072 | Nguyễn Thị Kim Anh | x |  | x | x | x |
| 205201268 | Đặng Công Hà |  | x |  |  | x |