**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

A picture containing graphics, graphic design, logo, text

Description automatically generated

**BÁO CÁO ĐỒ ÁN**

**MÔ HÌNH TRUY XUẤT THÔNG TIN**

**VECTOR SPACE MODEL VÀ BOOLEAN INDEPENDENCE MODEL**

**Môn học:** Truy Xuất Thông Tin

**Lớp:** CS419.N21.KHCL

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

**Nhóm thực hiện:** Nguyễn Thành Lâm - 20521517

Lê Nguyễn Tiến Đạt - 20521167

Lê Văn Khoa - 20521467

**TP. HỒ CHÍ MINH, 2023**

**MỤC LỤC**

[**I. GIỚI THIỆU 3**](#_Toc140078344)

[**1.1. Truy xuất thông tin là gì? 3**](#_Toc140078345)

[**1.2. Mô hình truy xuất thông tin 4**](#_Toc140078346)

[**1.3. Mục tiêu của bài toán 5**](#_Toc140078347)

[**II. CƠ SỞ LÝ THUYẾT 5**](#_Toc140078348)

[**2.1. Term là gì ? 5**](#_Toc140078349)

[**2.2. Tiền xử lý dữ liệu 6**](#_Toc140078350)

[**2.3. Lập chỉ mục 9**](#_Toc140078351)

[**2.4. Ngữ liệu 10**](#_Toc140078352)

[**III. VECTOR SPACE MODEL (VSM) 10**](#_Toc140078353)

[**3.1. Giới thiệu 10**](#_Toc140078354)

[**3.2. Lập chỉ mục 13**](#_Toc140078355)

[**3.3. Quá trình truy vấn và tính độ tương đồng 16**](#_Toc140078356)

[**IV. BOOLEAN INDEPENDENCE MODEL (BIM) 18**](#_Toc140078357)

[**4.1. Giới thiệu 18**](#_Toc140078358)

[**4.2. Lập Chỉ Mục 20**](#_Toc140078359)

[**4.3. Quá trình truy vấn và xếp hạng tài liệu 23**](#_Toc140078360)

[**V. ĐÁNH GIÁ 25**](#_Toc140078361)

[**5.1. Các độ đo đánh giá 25**](#_Toc140078362)

[**5.1.1. Recall và precision 25**](#_Toc140078363)

[**5.1.2. Độ chính xác trung bình không nội suy (Average Pecision) 26**](#_Toc140078364)

[**5.2. Kết quả thực nghiệm 29**](#_Toc140078365)

[**5.3. Cài đặt thực nghiệm 31**](#_Toc140078366)

[**5.3.1. Thiết kế chương trình thực nghiệm 31**](#_Toc140078367)

[**5.3.2. Môi trường thực nghiệm 33**](#_Toc140078368)

[**VI. KẾT LUẬN 35**](#_Toc140078369)

[**6.1. Nhận xét các mô hình 35**](#_Toc140078370)

[**6.2. Kinh nghiệm rút ra sau khi hoàn thành đồ án 36**](#_Toc140078371)

[**VII. TÀI LIỆU THAM KHẢO 37**](#_Toc140078372)

# GIỚI THIỆU

* 1. **Truy xuất thông tin là gì?**
* Truy xuất thông tin là quá trình tìm kiếm, lấy ra và hiển thị dữ liệu từ một nguồn thông tin hoặc cơ sở dữ liệu. Quá trình này thường được thực hiện bằng cách sử dụng một ngôn ngữ truy vấn hoặc giao diện người dùng đồ họa.
* Cụ thể hơn Truy xuất thông tin là tìm kiếm:
  + Vật liệu chứa thông tin (document)
  + Phi cấu trúc hoặc bán cấu trúc: Free Text, XML
  + Từ các bộ lưu trữ lớn (Collections)
  + Thỏa yêu cầu

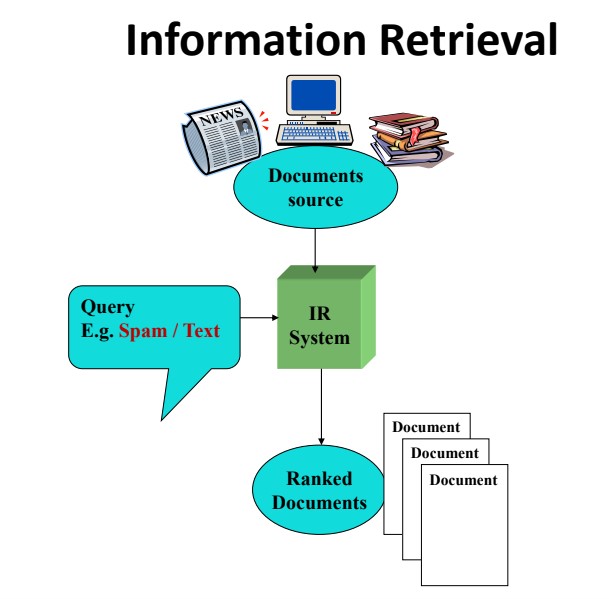


Figure 1: Quá trình truy xuất thông tin

* Một số ứng dụng của Information Retrieval có thể kể đến là Web Search: Google, Bing, Youtube… Ngoài ra còn bao gồm: Q&A, Text mining (Khai thác dữ liệu), Online Advertising (quảng cáo trực tuyến),…

A screenshot of a computer

Description automatically generated

Figure 2: Microsoft Bing Search

A screenshot of a computer

Description automatically generated

Figure 3: Google Search

* Tùy thuộc vào ng dụng mà các đối tượng dữ liệu (data object) có thể là tài liệu văn bản, hình ảnh, âm thanh, mindmap hoặc video. Thông thường, bản thân các tài liệu sẽ không được lưu trữ trực tiếp trong hệ thống IR mà thay vào đó sẽ được thể hiện trong hệ thống bằng các tài liệu thay thế hoặc metadata.
  1. **Mô hình truy xuất thông tin**

Nhóm em sẽ sử dụng 2 mô hình Vector Space Model (VSM) và Boolean Independence Model (BIM).

A diagram of a diagram

Description automatically generated

Figure 4: Mô hình truy vấn tổng quát

* 1. **Mục tiêu của bài toán**

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 đảm 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 accurancy, precision và recall.

# CƠ SỞ LÝ THUYẾT

* 1. **Term là gì ?**
* Một term có thể là một từ, 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, ‘studies” và ‘computing’ sau khi stemming (kĩ thuật loại bỏ một phần của từ thành một từ gốc) sẽ trở thành ‘studi’ và ‘comput’.
* Trong đồ án này, sử dụng tập Cranfield là một tập chứa các từ tiếng anh. Vì tiếng anh không phân biệt giữa từ đơn và từ ghép nên ta xác định 1 term là 1 từ.
  1. **Tiền xử lý dữ liệu**

Việc tiền xử lý rất là quan trọng và ảnh hưởng rất nhiều tới kết quả truy vấn vì cơ bản văn bản được liệt kê mà không có cấu trúc, để nguyên để xử lý rất là khó khăn. Đặc biệt là loại văn bản trên web có lẫn các thẻ HTML, gây khó khăn trong việc truy vấn. Để tiền xử lý dữ liệu một cách tối ưu nhất thì trãi qua 4 bước sau:

**Bước 1: Tách từ (Tokenize)**

* Tokenization là quá trình chia câu hoặc đoạn văn thành các phần tử nhỏ hơn được gọi là "token". Token có thể là các từ, các ký tự, hoặc các đơn vị ngữ nghĩa khác như từng câu, từng cụm từ, hoặc từng biểu thức. Mức độ tokenization có thể thay đổi tùy thuộc vào mục đích và yêu cầu của ứng dụng xử lý ngôn ngữ tự nhiên.

Ví dụ: 'some structural and aerelastic considerations of high speed flight .'

Kết quả: ['some', 'structural', ‘and’, 'aerelastic', 'considerations', 'of', 'high', 'speed', 'flight', '.']

**Bước 2: Loại bỏ Stopword**

* Việc loại bỏ Stopword để tiết kiệm dung lượng lưu trữ, tiết kiệm thời gian xử lý. 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. Nhóm sử dụng bộ Stopword của **NLTK** bao gồm 179 từ bao gồm: 'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'no', 'such', 'some', 'only',...

Ví dụ: ‘He is a good football player.’

Kết quả: ‘He good football player’ (‘is’ và ‘a’ là Stopword)

**Bước 3: Re.Sub và Lowercase**

* Việc sử dụng *re.sub* và *lowercase* trong tiền xử lý dữ liệu giúp chuẩn hóa và loại bỏ các yếu tố không cần thiết trong văn bản, đồng thời tạo ra sự đồng nhất và giảm sự phức tạp trong quá trình xử lý dữ liệu.
* ***Re.sub*** là phương pháp sẽ thay thế các kết quả khớp với pattern trong chuỗi bằng một nội dung khác được truyền vào và trả về chuỗi đã được sửa đổi. Mục đích là để loại bỏ các ký tự đặc biệt, dấu câu hoặc ký tự không mong muốn. Các ký tự được thay thế bao gồm : !"#$%&'()\*+,-./:;<=>?@[\]^\_`{|}~
* ***Lowercase*** là chuyển tất cả các ký tự in hoa trong câu về in thường. Điều này giúp chuẩn hóa dữ liệu bằng cách đưa tất cả các ký tự về dạng thường, loại bỏ sự phân biệt chữ hoa-chữ thường trong quá trình xử lý.

Ví dụ: ‘Hypersonic Speeds (speeds faster than about mach 5), Newtonian Flow’

Kết quả: ‘hypersonic speeds speeds faster than about mach 5 newtonian flow’’

**Bước 4: 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: walks, walking, walked có gốc là walk.
* Mục đích của việc lấy gốc từ là để giảm kích thước chỉ mục và

cho phép truy xuất các tài liệu với nhiều dạng biến tố của cùng một từ.

* Có 2 phương pháp lấy gốc từ phổ biến là Stemming và Lemmatization.
* **Stemming** là quá trình loại bỏ một phần (cuối hoặc đầu) của từ, hoặc giảm một từ thành từ gốc (root word) của nó. Có nhiều loại thuật toán Stemming phổ biến như là *Porter stemmer, Snowball stemmer* và *Lancaster stemmer*...

***Porter stemmer*** là thuật toán được sử dụng rộng rãi nhất và nó dựa trên một tập hợp các phương pháp phỏng đoán được sử dụng để loại bỏ các hậu tố phổ biến khỏi các từ.

***Snowball stemmer*** là một thuật toán tiên tiến hơn dựa trên Porter stemmer, nhưng nó cũng hỗ trợ một số ngôn ngữ khác ngoài tiếng Anh.

***Lancaster stemmer*** kém chính xác hơn so với Porter stemmer và Snowball stemmer.

Ví dụ: ‘retrieval’, ‘retrieved’, ‘retrieves’ rút gọn thành từ gốc ‘retrieve’.

* **Lemmatization** là phương pháp **x**em xét phân tích hình thái của các từ để chuyển về những từ tương tự từ gốc. Giúp đồng nhất các biến thể từ ngữ và giảm sự phức tạp trong việc xử lý và phân tích dữ liệu.

Ví dụ: ‘improving’, ‘improvements’, ‘improved’, ‘improver’ rút gọn thành từ gốc ‘improve’.

* Quá trình tiền xử lý dữ liệu:

Ví dụ: ‘For a plate (with finite bending rigidity) the results confirm those given by the Galerkin method .’

Bước 1: Tách từ

A blue rectangles with white text

Description automatically generated

Bước 2: Loại bỏ Stopword

A blue squares with white text

Description automatically generated

* Stopword: for, a, with, the, those, by, the

Bước 3: Re.sub và Lowercase

A blue rectangle with white text

Description automatically generated

* Loại bỏ các kí tự đặc biệt: ‘(‘, ‘)’, ‘.’ và Lowercase từ ‘Galerkin’

Bước 4: Lấy gốc từ bằng Porter Stemming

A blue rectangles with white text

Description automatically generated

* ‘finite’ → ‘finit’, ‘bending’ → ‘bend’, ‘rigidity’ → ‘rigid’, ‘results’ → ‘result’.

Kết quả cuối cùng sau khi trải qua các bước tiền xử lý dữ liệu: ['finit', 'bend', 'rigid', 'result', 'confirm', 'given', 'galerkin', 'mood']

* 1. **Lập chỉ mục**
* Lập chỉ mục tài liệu là quá trình liên kết thông tin với một tệp hoặc thẻ cụ thể cho phép dễ dàng tìm thấy và truy xuất sau này. Chẳng hạn, các tài liệu sẽ được biểu diễn thành các vector trong Vector Space Model. Thông tin được lập chỉ mục sau đó sẽ được lập trình thành một hệ thống quản lý tài liệu.
* Mục đích của việc lập chỉ mục là tăng cường hiệu suất và tốc độ truy vấn dữ liệu đồng thời giúp người dùng dễ dàng truy cập vào dữ liệu mà họ yêu cầu. Nếu không lập chỉ mục tài liệu hiệu quả, việc truy xuất thông tin có thể rất tốn thời gian và chi phí.
* 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 – 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).
  1. **Ngữ liệu**

Tập tài liệu Cranfield dùng để tính trên hai mô hình Vector Space Model (VSM) và Boolean Independence Model (BIM).

# VECTOR SPACE MODEL (VSM)

* 1. **Giới thiệu**
* Là phương pháp tính toán sự tương đồng giữa truy vấn và tài liệu.
* Vector Space Model là mô hinh truy vấn thông tin trong đó văn bản được biểu diễn như một vector với đặc điểm:
  + Tài liệu được biểu diễn dưới dạng “Bag of Word”
  + Tài liệu là một mảng số thực giống như một vector nhiều chiều.
  + Mỗi term là một chiều trong không gian. (Mỗi chiều ứng với một term trong tập hợp các từ)
  + Tính toán dựa trên hướng và độ lớn.

***Mô hình Bag of Word:*** *là xây dựng một "túi" (bag) chứa tất cả các từ trong văn bản và ghi nhận số lần xuất hiện của từng từ.*

*Sau quá trình xây dựng túi từ, mỗi văn bản sẽ được biểu diễn dưới dạng một vector số học, trong đó mỗi phần tử của vector tương ứng với số lần xuất hiện của một từ trong túi từ.*

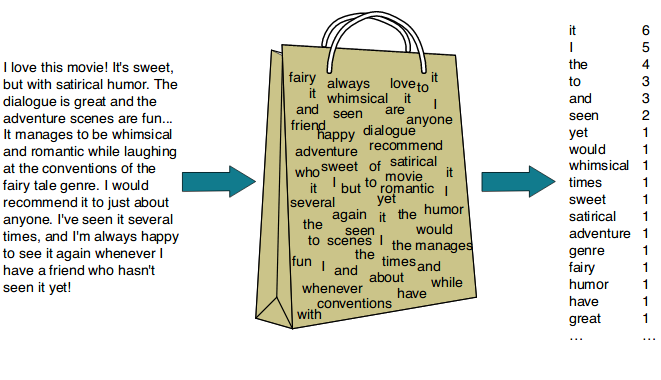


Figure 5: Bag of Word

Ví dụ:

: this is a good boy.

: this is a pretty girl.

Từ điển: [this, is, a, good, boy, pretty, girl]

Sau khi áp dụng mô hình Bag of Word ta có , tương ứng với ,

* Phương pháp so khớp:
  + Truy vấn được biểu diễn bằng một vector cùng không gian với tài liệu. Tính toán giữa truy vấn và tài liệu dựa trên chiều dài và hướng của vector tương ứng của chúng.
  + Khoảng cách giữa hai vector tài liệu và truy vấn được xem là độ tương đồng giữa tài liệu và truy vấn. Khoảng cách này được dùng để xếp hạng tài liệu.
  + Khoảng cách này được tính bằng ***Cosine Similarity*** hoặc ***Euclidean Distance***

A black and blue math equation

Description automatically generated

Figure 6: Cosine Similarity

* Kết quả cosine similarity nằm trong khoảng [-1, 1], trong đó 1 cho thấy hai vector hoàn toàn giống nhau, 0 cho thấy hai vector không có sự tương đồng và -1 cho thấy hai vector hoàn toàn đối nghịch nhau.

A math equation with a square and square and square with a square and square with a square and a square with a square and a square with a square and a square with a square and a square

Description automatically generated

Figure 7: Eucildean Distance

* Đặc điểm của Vector tài liệu
  + Vị trí các term tạo thành không gian không quan trọng
  + Giá trị mỗi chiều của vector tài liệu hay truy vấn là trọng số của term trong tài liệu hay truy vấn đó

Ví dụ ma trận tài liệu (Doc-term):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Doc Term | network | comput | neural | predict |
|  | 10 | 2 | 10 | 1 |
|  | 0 | 20 | 0 | 8 |
|  | 30 | 18 | 1 | 3 |
|  | 1 | 32 | 0 | 6 |
|  | 23 | 16 | 22 | 9 |
|  | 0 | 6 | 4 | 7 |

* Với cách tính trọng số là tần số xuất hiện của từ khóa trong tài liệu thì vector cho biết:

+ Từ “network” xuất hiện 10 lần.

+ Từ “comput” xuất hiện 2 lần.

+ Từ “neural” xuất hiện 10 lần.

+ Từ “predict” xuất hiện 1 lần.

* 1. **Lập chỉ mục**
* **Trọng số Term:** Là giá trị của chiều tương ứng trong vector tài liệu và vector truy vấn. Ảnh hưởng lớn đến độ chính xác và sự tương đồng của những tài liệu.
* Việc các từ có trọng số khác nhau là do không phải các từ đều có sự quan trọng giống nhau, nên sử dụng số lần xuất hiện của các từ làm vector không phải là một cách tối ưu.
* Công thức sự dụng để tính trọng số: TF-IDF (Term Frequency-Inverse Document Frequency). Công thức này được sử dụng để đo lường tầm quan trọng của một từ trong một tài liệu hoặc tập tài liệu.
  + - Trong đó:

TF: là số lần xuất hiện của term trong tài liệu

IDF: được tính theo công thức

* Trọng số TF-IDF sẽ cho ta thông tin về tầm quan trọng của term trong tài liệu cụ thể so với tập tài liệu toàn bộ. Các term có trọng số cao sẽ có sự xuất hiện quan trọng trong tài liệu và ít xuất hiện trong các tài liệu khác, đồng thời bỏ qua các term phổ biến xuất hiện nhiều trong nhiều tài liệu.
* Chuẩn hóa trọng số theo SMART
  + Nhằm khắc phục sự khác biệt giữa tài liệu dài và tài liệu ngắn.
  + Đưa trọng số về miền giá trị [0, 1]
  + Chuyển vector tài liệu thành vector cùng phương có độ dài bằng 1



Figure 8: Công thức chuẩn hóa trọng số term theo SMART

* Có rất nhiều cách tính , và khác nhau, nhưng trong bài báo cáo này nhóm em sử dụng công thức:

  + (Chuẩn Cosine)
* Ví dụ về cách lập chỉ mục và chuẩn hóa trọng số term

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Doc Term | bank | import | price | transport |
|  | 3 | 0 | 4 | 0 |
|  | 1 | 6 | 1 | 0 |
|  | 2 | 5 | 0 | 4 |
|  | 0 | 0 | 3 | 0 |

**B1: Từ ma trận Doc-Term, ta xác định được từ điển và danh sách Posting như sau:**

A diagram of a number

Description automatically generated

**B2: Tính giá trị IDF và w:**

A table with numbers and lines

Description automatically generated

**B3: Chuẩn hóa w theo SMART và kết quả lập chỉ mục**

A diagram of a data flow

Description automatically generated

* 1. **Quá trình truy vấn và tính độ tương đồng**

**Xác định tài liệu liên quan**

Đối với mỗi term truy vấn ta cần phải:

* Xác định tần số của nó
* Xác định vị trí của nó trong từ điển để xác định:
  + : số tài liệu chứa term này.
  + Tính trọng số w cho term
  + Xác định vị trí trong danh sách posting.
  + Xác định danh sách tài liệu từ danh sách posting.
* Ví dụ: Dựa vào kết quả lập chỉ mục ở trên, quá trình xử lý truy vấn câu q: *transportation pricing*

**B1: Tiền xử lý truy vấn**

Lọc stopword và stemming: transport price

**B2: Chuẩn hóa**

+ Truy xuất từ điển để lấy IDF để tính trọng số:

+ Chuẩn hóa

+ Tập tài liệu liên quan ban đầu R = {}. Truy xuất chỉ mục:

**Xử lý term ‘transport’**

|  |  |
| --- | --- |
| **DocID** | **w** |
| 3 | 0.97\*0.87 = 0.84 |

* R = {(3,0.84)}

**Xử lý term ‘price’**

|  |  |
| --- | --- |
| **DocID** | **w** |
| 1 | 0.13\*0.47 = 0.06 |
| 2 | 0.13\*0.15 = 0.02 |
| 4 | 0.13\*1 = 0.13 |

* R = {(3,0.84); (1,0.06); (2,0.02); (4,0.13)}

Vậy kết quả xử lý truy vấn q là tập tài liệu được xếp thứ tự:

**Tính toán độ tương đồng**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Doc Term | bank | import | price | transport |
|  | 0.66 | 0 | 0.47 | 0 |
|  | 0.15 | 2.8 | 0.15 | 0 |
|  | 0.13 | 2.6 | 0 | 0.87 |
|  | 0 | 0 | 1 | 0 |
| q | 0 | 0 | 0.97 | 0.13 |

Áp dụng tính độ tương đồng bằng công thức Cosine Similarity

# BOOLEAN INDEPENDENCE MODEL (BIM)

* 1. **Giới thiệu**

Mô hình Boolean Independence Model (BIM) là một mô hình được sử dụng trong lĩnh vực truy vấn thông tin và tìm kiếm thông tin để đo lường độ độc lập giữa các thuộc tính trong tập dữ liệu và xác định sự phụ thuộc giữa chúng trong quá trình tìm kiếm. Mô hình này dựa trên giả định về độc lập hoàn toàn giữa các thuộc tính.

Tài liệu và truy vấn được biểu diễn dưới dạng các phân phối xác suất của các từ trong từ điển. Mỗi từ được đại diện bởi một giá trị xác suất trong vector. Cuối cùng, điểm xếp hạng của tài liệu sẽ được tính bằng tổng các xác suất của các từ trong truy vấn, nhân với xác suất của từng từ trong tài liệu.

Tần số của mỗi term là 0 (không xuất hiện) và 1 (có xuất hiện)

***Công thức:***

Ta có 2 giả thuyết sau:

* Giả thuyết độc lập: Giả thiết này đơn giản hóa việc tính toán và khá hiệu quả.
* Các term của câu truy vấn là yếu tố duy nhất xác định sự liên quan giữa tài liệu và truy vấn:

Với term tdi∉q thì p(tdi |q,r) không phụ thuộc vào r:

p(tdi|q,r) = p(tdi|q,-r)

* Chỉ cần tính xác suất các term trong truy vấn q

Dựa trên hai giả thiết trên, sự liên quan được thể hiện qua giá trị:

Việc ước lượng giá trị p(td|r,q) và p(td|¬r,q) được thực hiện theo hai trường hợp:

* Trường hợp không có ngữ liệu mẫu:

Độ liên quan giữa tài liệu và truy vấn:

=

Sử dụng độ đo log-odds:

* Trọng số của mỗi term là wtd=log(0.5\*N/Ntd)
* Trường hợp có ngữ liệu mẫu

rtd là số tài liệu liên quan chứa term td

NR là tổng số tài liệu liên quan   
Ước lượng các xác suất:

p(td|r,q) =

p(td|-r,q) =

Để tránh trường hợp rtd=0 và rtd=Ntd, thực hiện smoothing:

p(td|r,q) =

p(td|-r,q) =

Độ liên quan chúng ta sẽ tính theo công thức :

Rel(d,q) =

Trong đó:

* 1. **Lập Chỉ Mục**

**Bước 1:** Xác định các term

* Là các từ cách nhau bằng khoảng trắng.
* Không phải là liên từ, danh sách các liên từ và các từ nối ( and, or,…)

**Bước 2:** Tiền xử lý

* Tiền xử lý các doc bằng các tách từ, loại bỏ stopword và bỏ các liên từ.

**Bước 3:** Tạo danh sách các term và tạo từ điển và danh sách posting

**Bước 4: Tính trọng số**  qua danh sách postings vừa lập được bằng công thức:

Trong đó:

N là tổng số tài liệu trong tập dữ liệu

Ntd là tổng số tài liệu chưa term đó.

* Ví dụ về cách lập chỉ mục và tính trọng số term:

d1: football cricket

d2: cricket termite grasshopper

d3: football football team hockey

d4: termite team goal

d5: obama football and hockey team

**Bước 1: Xác định các term**:

Các term ở bài toán này là : {football, cricket, termite, grasshoper, hockey, goal, obama, team}

**Bước 2: Tiền xử lý bằng cách tách từ + loại bỏ các liên từ**.

d1: football cricket

d2: cricket termite grasshopper

d3: football football team hockey

d4: termite team goal

d5: obama football hockey team

**Bước 3: Tạo danh sách các term đã được xác đinh ở bước 1 và ghi DocID mà nó xuất hiện.**

|  |  |
| --- | --- |
| Term | docID |
| football | 1 |
| cricket | 1 |
| cricket | 2 |
| termite | 2 |
| grasshoper | 2 |
| football | 3 |
| football | 3 |
| team | 3 |
| hockey | 3 |
| termite | 4 |
| team | 4 |
| goal | 4 |
| obama | 5 |
| football | 5 |
| hockey | 5 |
| team | 5 |

**Bước 4: Tạo từ điển và danh sách posting list**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Term* | *ndoc* | *Posting list* | | | |
| cricket | 2 | 1 | | 2 | |
| football | 3 | 1 | 3 | | 5 |
| goal | 1 | 4 | | | |
| grasshoper | 1 | 2 | | | |
| hockey | 2 | 3 | | 5 | |
| obama | 1 | 5 | | | |
| team | 3 | 3 | 4 | | 5 |
| termite | 2 | 2 | | 4 | |

**Bước 5: Tính trọng số cho từng term bằng công thức:**

Ta có kết quả như sau:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Term* | *ndoc* | *Posting list* | | | |  |
| cricket | 2 | 1 | | 2 | | 0,322 |
| football | 3 | 1 | 3 | | 5 | -0,263 |
| goal | 1 | 4 | | | | 1,32 |
| grasshoper | 1 | 2 | | | | 0,322 |
| hockey | 2 | 3 | | 5 | | -0,263 |
| obama | 1 | 5 | | | | 1,32 |
| team | 3 | 3 | 4 | | 5 | -0,263 |
| termite | 2 | 2 | | 4 | | 0.322 |

* 1. **Quá trình truy vấn và xếp hạng tài liệu**

**Xác định tài liệu liên quan**

Đối với mỗi term truy vấn ta cần phải:

* Xác định posting lists của nó
* Xác định vị trí của nó trong từ điển để xác định:
  + : số tài liệu chứa term này.
  + Tính trọng số cho term
  + Xác định vị trí trong danh sách posting.
  + Xác định danh sách tài liệu từ danh sách posting.
* Ví dụ: Dựa vào kết quả lập chỉ mục ở trên, quá trình xử lý truy vấn câu q: football cricket hockey termite

**Bước 1: Tiền xử lý truy vấn**

Lọc stopword, bỏ các liên từ và stemming câu truy vấn, ta được

q : football cricket hockey termite.

**Bước 2: Xử lý các term**

**Xử lý term ‘football”**

**R = { }**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DocID** | **Posting list** | | |  |
| 3 | 1 | 3 | 5 | -0,263 |

* R = {(1, -0,263), (3, -0,263), (5, -0,263)}

**Xử lý term ‘cricket”**

**R = {(1, -0,263), (3, -0,263), (5, -0,263)}**

|  |  |  |  |
| --- | --- | --- | --- |
| **DocID** | **Posting list** | |  |
| 2 | 1 | 2 | -0,263 |

* R = {(1, 0,059), (2, 0,322), (3, -0,263), (5, -0,263)}

**Xử lý term ‘hockey”**

**R = {(1, 0,059), (2, 0,322), (3, -0,263), (5, -0,263)}**

|  |  |  |  |
| --- | --- | --- | --- |
| **DocID** | **Posting list** | |  |
| 2 | 3 | 5 | 0,322 |

* R = {(1, 0,059), (2, 0,322), (3, 0,059), (5, 0,059)}

**Xử lý term ‘termite”**

**R = {(1, 0,059), (2, 0,322), (3, 0,059), (5, 0,059)}**

|  |  |  |  |
| --- | --- | --- | --- |
| **DocID** | **Posting list** | |  |
| 2 | 2 | 4 | 0,322 |

* R = {(1, 0,059), (2, 0,644), (3, 0,059), (4, 0,322), (5, 0,059)}

Sắp xếp các doc theo thứ tự giảm dần, ta có:

R = {(2, 0,644), (4, 0,322), (1, 0,059), (3, 0,059), (5, 0,059)}

Vậy kết quả xử lý truy vấn q là tập tài liệu được xếp thứ tự:

**Tính toán độ liên quan:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | cricket | football | goal | grasshoper | hockey | obama | team | termite |
|  | 0.322 | -0,263 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | 0.322 | 0 | 0 | 1,32 | 0 | 0 | 0 | 0,322 |
|  | 0 | -0,263 | 0 | 0 | 0,322 | 0 | -0,263 | 0 |
|  | 0 | 0 | 1,32 | 0 | 0 | 0 | -0,263 | 0,322 |
|  | 0 | -0,263 | 0 | 0 | 0,322 | 1,32 | -0,263 | 0 |
| q | 0,322 | -0,263 | 0 | 0 | 0,322 | 0 | 0 | 0,322 |

Ta sẽ tính toán độ liên quan bằng công thức:

Rel(d,q) =

Rel(d1,q) = 0,762

Rel(d2,q) = 1,288

Rel(d3,q) = -0,934

Rel(d4,q) = 0,644

Rel(d5,q) = -0,408

# ĐÁNH GIÁ

* 1. **Các độ đo đánh giá**
     1. **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:

* + 1. **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:

* + 1. **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 1:** 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

**Ví dụ 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

* 1. **Kết quả thực nghiệm**

Sau khi thực hiện 2 mô hình VSM và Bim thì nhóm có làm thêm một mô hình nữa đó chính là mô hình WHOOSH

Nhóm em rút ra được kết quả thực nghiệm saukhi sử dụng tập Cranfield trong các trường hợp sau:

* Trường hợp các tài liệu và truy vấn được xử lý bằng Stemming:

|  |  |  |  |
| --- | --- | --- | --- |
|  | VSM | BIM | WHOOSH |
| Precision | 0.008 | 0.005 | 0.007 |
| Recall | 0.947 | 0.99 | 0.937 |
| MAPr | 0.352 | 0.308 | 0.257 |

Từ bảng trên nhóm vẽ được biểu đồ như sau:

A graph of different colored bars

Description automatically generated

* Trường hợp các tài liệu và truy vấn được xử lý bằng Lemmatization

|  |  |  |  |
| --- | --- | --- | --- |
|  | VSM | BIM | WHOOSH |
| Precision | 0.009 | 0.005 | 0.007 |
| Recall | 0.937 | 0.988 | 0.937 |
| MAPr | 0.355 | 0.298 | 0.257 |

Từ bảng trên nhóm vẽ được biểu đồ như sau:

A graph of different colored bars

Description automatically generated

Sau khi tiến hành thực nghiệm, nhóm em rút ra được mô hình VSM có Precision, Recall và MAPr cao nhất sau đó đến mô hình BIM và cuối cùng là Whoosh.

* 1. **Cài đặt thực nghiệm**
     1. **Thiết kế chương trình thực nghiệm**

**Ngôn ngữ lập trình Python:**

Python là một ngôn ngữ lập trình thông dịch, đa mục đích và dễ học. Nó đã được phát triển vào những năm 1980 bởi Guido van Rossum và trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới. Nhóm chọn cài đặt thuật toán trên ngôn ngữ lập trình Python vì:

* Dễ học và đọc: Python có cú pháp đơn giản và rõ ràng, giúp việc đọc và hiểu mã nguồn dễ dàng hơn so với nhiều ngôn ngữ khác. Nó sử dụng các từ khóa và cấu trúc ngữ pháp tự nhiên, giúp tạo ra mã nguồn gần như giống với câu nói tiếng Anh thông thường.
* Đa năng: Python là một ngôn ngữ đa năng, có thể được sử dụng để phát triển nhiều loại ứng dụng khác nhau. Nó hỗ trợ phát triển web, phân tích dữ liệu, trí tuệ nhân tạo, đồ họa máy tính, trò chơi và nhiều lĩnh vực khác. Python cũng được sử dụng làm một ngôn ngữ kịch bản (scripting language) cho các tác vụ tự động hóa.
* Cú pháp động: Python là một ngôn ngữ cú pháp động, có nghĩa là bạn không cần khai báo kiểu dữ liệu cho biến. Kiểu dữ liệu của biến được xác định tự động dựa trên giá trị mà nó đang lưu trữ. Điều này giúp đơn giản hóa việc viết mã và tăng tính linh hoạt.
* Hỗ trợ thư viện phong phú: Python có một cộng đồng lớn và năng động, cung cấp rất nhiều thư viện và module cho nhiều mục đích khác nhau. Ví dụ, thư viện NumPy hỗ trợ tính toán số học và khoa học, thư viện Pandas hỗ trợ xử lý dữ liệu, và thư viện TensorFlow hỗ trợ trí tuệ nhân tạo. Sự phong phú của các thư viện này giúp việc phát triển ứng dụng Python nhanh chóng và hiệu quả.
* Tương thích và giao tiếp tốt: Python có khả năng tương thích với hầu hết các hệ điều hành phổ biến như Windows, macOS và Linux. Ngoài ra, Python cũng có khả năng giao tiếp với các ngôn ngữ khác như C/C++, Java và .NET. Điều này cho phép bạn sử dụng các thư viện và module có sẵn trong các ngôn ngữ khác và tích hợp chúng vào ứng dụng Python của mình.
* Hỗ trợ cộng đồng mạnh mẽ: Python có một cộng đồng phát triển lớn và nhiệt tình. Cộng đồng này cung cấp tài liệu phong phú, hỗ trợ trực tuyến, diễn đàn thảo luận và các dự án mã nguồn mở. Điều này giúp bạn giải quyết các vấn đề kỹ thuật, tìm kiếm thông tin và học hỏi từ những người khác.

Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới do tính đa năng, dễ học và sử dụng. Nó được sử dụng rộng rãi trong các công ty công nghệ hàng đầu, các tổ chức nghiên cứu, trường đại học và cộng đồng phát triển phần mềm trên toàn cầu.

* + 1. **Môi trường thực nghiệm**

Jupyter Notebook là một ứng dụng mã nguồn mở được sử dụng rộng rãi trong việc phân tích dữ liệu, phát triển và chia sẻ mã Python (cũng như các ngôn ngữ lập trình khác). Nó cung cấp một môi trường tương tác trực quan cho việc viết và thực thi mã, hiển thị kết quả, tạo ra văn bản, hình ảnh, công thức toán học và biểu đồ.

Nhóm chọn sử dụng Jupyter Notebook thay vì các IDE thông thường vì một số lý do sau:

* Giao diện web tương tác: Jupyter Notebook được truy cập thông qua trình duyệt web, cho phép bạn làm việc với mã nguồn, kết quả và nội dung tài liệu trong một giao diện trực quan và dễ sử dụng. Bạn có thể viết mã, chạy nó và xem kết quả ngay trên trình duyệt mà không cần sử dụng môi trường lập trình truyền thống.
* Tích hợp mã và văn bản: Jupyter Notebook cho phép bạn viết mã Python và văn bản giải thích, mô tả hoặc hướng dẫn trong cùng một tài liệu. Bạn có thể sử dụng định dạng Markdown để tạo tiêu đề, định dạng văn bản, thêm hình ảnh, tạo công thức toán học và tạo liên kết. Điều này rất hữu ích khi bạn muốn tạo ra một tài liệu phân tích dữ liệu hoặc báo cáo kỹ thuật.
* Tương tác trực tiếp: Một trong những đặc điểm quan trọng của Jupyter Notebook là khả năng tương tác trực tiếp với mã. Bạn có thể chạy từng dòng mã một hoặc chạy toàn bộ tệp Notebook, đồng thời xem kết quả trực tiếp trong cùng một tài liệu. Điều này giúp bạn kiểm tra và hiểu kết quả từng bước của phân tích dữ liệu hoặc quy trình lập trình.
* Hỗ trợ nhiều ngôn ngữ lập trình: Mặc dù được phát triển ban đầu cho Python, Jupyter Notebook cũng hỗ trợ nhiều ngôn ngữ lập trình khác như R, Julia, Scala và nhiều ngôn ngữ khác thông qua các kernel mở rộng. Điều này cho phép bạn sử dụng các ngôn ngữ khác và tận dụng được tính tương tác và tạo tài liệu của Jupyter Notebook.
* Tích hợp các thư viện phổ biến: Jupyter Notebook được tích hợp với nhiều thư viện phổ biến cho phân tích dữ liệu và khoa học máy tính như NumPy, Pandas, Matplotlib, Scikit-learn và nhiều thư viện khác. Điều này giúp bạn tiếp cận và sử dụng các công cụ phân tích mạnh mẽ một cách thuận tiện và hiệu quả.
* Chia sẻ và tái sử dụng: Jupyter Notebook cho phép bạn chia sẻ tài liệu của mình với người khác dễ dàng. Bạn có thể xuất tài liệu của mình thành các định dạng như HTML, PDF, Markdown hoặc tệp Notebook thô (.ipynb) để chia sẻ hoặc xuất bản trực tuyến. Điều này giúp bạn trình bày kết quả phân tích và quá trình làm việc của mình một cách trực quan và tương tác cho người đọc.

Jupyter Notebook đã trở thành một công cụ quan trọng trong cộng đồng khoa học dữ liệu, phân tích dữ liệu và lập trình Python. Nó kết hợp tính tương tác, tài liệu và tạo ra một môi trường làm việc mạnh mẽ cho việc thực hiện nhiều nhiệm vụ khác nhau như phân tích dữ liệu, học máy, trực quan hóa dữ liệu và nghiên cứu khoa học.

# KẾT LUẬN

* 1. **Nhận xét các mô hình**

Mô hình Boolean Independence là một mô hình truy vấn chỉ dựa vào việc term có xuất hiện hay không xuất hiện trong tập tài liệu truy vấn và trong câu truy vấn chứ hoàn toàn không phụ thuộc vào tần số của term trong tài liệu cũng như trong câu truy vấn. Điều này có thể dẫn đến việc một tài liệu liên quan hơn lại bị sắp xếp sau một tài liệu ít liên quan hơn do có chỉ số relevant thấp hơn. Mặc dù mô hình đạt hiệu quả cũng khá cao nhưng thời gian chạy chương trình còn khá là chậm.

Mô hình Vector Space là một mô hình dựa vào tần số xuất hiện của term có trong tập tài liệu và trong câu truy vấn. Điều này của mô hình Vector Space đã giải quyết được một phần nào đó về vấn đề độ liên quan của tài liệu ở mô hình Boolean Independence. Mô hình cho kết quả MAP và MAP nội suy cũng khá tốt đồng thời thời gian thực hiện chương trình cũng khá nhanh. Nhưng mô hình Vector Space chỉ dựa vào term có sẵn trong chỉ mục để truy vấn nên có thể không truy vấn được theo nhu cầu thông tin. Ngoài ra VSM cũng tồn tại những nhược điểm như có rất nhiều các công thức tính trọng số, cách chuẩn hóa (normalize) vector. Vì vậy khá khó khăn trong việc chọn ra công thức phù hợp nhất để xây dựng chương trình truy xuất thông tin.

* 1. **Kinh nghiệm rút ra sau khi hoàn thành đồ án**

Sau khi hoàn thành xong đề tài này, các thành viên trong nhóm chúng em đã đều có khả năng trả lời các câu hỏi liên quan tới mô hình Boolean Independence và mô hình Vector Space. Bài báo cáo đi sát với toàn bộ nội dung thành viên đã thực hiện cũng như sưu tầm được những kiến thức cần phải phân tích rõ ràng. Trong quá trình nghiên cứu và tìm hiểu, chúng em đã thu được những kết quả thực nghiệm hữu ích thông qua việc áp dụng các phương pháp do nhóm đề xuất. Qua việc đánh giá các phương pháp, nhóm cũng đã thu được một số kinh nghiệm quý báu từ mô hình, tùy thuộc vào đặc điểm và yêu cầu của từng bài toán cụ thể mà lựa chọn phương pháp sao cho hợp lý. Mục đích của nhóm khi thực hiện thử nghiệm các trường hợp khác nhau khi cài đặt mô hình là để tìm ra phương pháp/mô hình tối ưu nhất với độ chính xác cao nhất có thể và đáp ứng được các yêu cầu đề ra. Ngoài ra, để có thể củng cố cho việc nghiên cứu, phát triển, chúng em sẽ cố gắng thu tập những thông tin cần thiết, tạo ra nhiều tính năng mới, đồng thời để khắc phục những hạn chế còn tồn tại trong đề tài này và nghiên cứu thêm những cái mới, hay để có thể hoàn thiện đề tài nghiên cứu về sau.

Qua đó cũng có thể đánh giá rằng, đồ án môn học do nhóm thực hiện không chỉ là một đồ án dừng lại ở mức độ chuyên ngành dành cho các thành viên trong nhóm, mà còn là cơ hội giúp các thành viên trong nhóm chúng em trau dồi thêm nhiều kiến thức và kỹ năng bổ ích, giúp mỗi thành viên trong nhóm nâng cao khả năng làm việc tập thể, nâng cao được tính tương tác giữa các thành viên, từ đó giúp bản thân mỗi thành viên trong nhóm qua quá trình học lập, làm việc nhóm rèn luyện thêm cho mình nhưng kĩ năng cần thiết cho công việc sau này.

# TÀI LIỆU THAM KHẢO

1. An Introduction to Information Retrieval: <https://www.academia.edu/27076940/An_Introduction_to_Information_Retrieval>
2. Information retrieval: <https://en.wikipedia.org/wiki/Information_retrieval#:~:text=Information%20retrieval%20is%20the%20science,of%20texts%2C%20images%20or%20sounds.>
3. Introduction to Information Retrieval:

<https://nlp.stanford.edu/IR-book/information-retrieval-book.html>

1. The Binary Independence Model

<https://nlp.stanford.edu/IR-book/html/htmledition/the-binary-independence-model-1.html>

1. Vector Space Model

<https://medium.com/m/global-identity-3?redirectUrl=https%3A%2F%2Ftowardsdatascience.com%2Fvector-space-models-48b42a15d86d>

<https://www.geeksforgeeks.org/web-information-retrieval-vector-space-model>

**Bảng phân công công việc**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **MSSV** | **Họ và tên** | **Mô hình VSM** | **Mô hình BIM** | **Mô hình Whoosh** | **Viết báo cáo** | **Làm slide** |
| 20521167 | Lê Nguyễn Tiến Đạt |  | x |  | x | x |
| 20521517 | Nguyễn Thành Lâm | x | x | x |  | x |
| 20521467 | Lê Văn Khoa | x |  |  | x | x |