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

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

BÁO CÁO ĐỒ ÁN CUỐI KỲ

CS419.N21.KHCL

VECTOR SPACE MODEL & LANGUAGE MODEL

Sinh viên thực hiện:

20521938 Trần Phương Thảo

20522176 Đặng Thị Tường Vy

20521568 Nguyễn Hoàng Long

Hồ Chí Minh, 06 – 2023

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

MỤC LỤC

NHẬN XÉT CỦA GIẢNG VIÊN 2

BẢNG PHÂN CÔNG CÔNG VIỆC.....7

I. GIỚI THIỆU TỔNG QUAN 8

1. ĐẶT VẤN ĐỀ.....8

2. KHÁI NIỆM "TRUY XUẤT THÔNG TIN" 9

2.1. SO KHỚP TÀI LIỆU 11

2.2. INPUT VÀ OUTPUT 11

3. QUY TRÌNH TRUY XUẤT THÔNG TIN.....12

4. CÁC BƯỚC TIỀN XỬ LÝ 13

5. LẬP CHỈ MỤC 15

6. XỬ LÝ TRUY VẤN 16

7. ỨNG DỤNG CỦA TRUY XUẤT THÔNG TIN.....18

II. VECTOR SPACE MODEL.....20

1. GIỚI THIỆU.....20

2. KÝ HIỆU 20

3. Ý TƯỞNG.....21

4. PHÂN TÍCH TÀI LIỆU VÀ LẬP CHỈ MỤC.....23

4.1. PHÂN TÍCH TÀI LIỆU 23

4.1.1. TOKENIZATION 23
4.1.2. LOWERCASE 24
4.1.3. LOẠI BỎ PUNCTUATIONS 24
4.1.4. LOẠI BỎ STOPWORD VÀ SỐ NGUYÊN 25
4.1.5. STEMMING 25
4.2. LẬP CHỈ MỤC 27
4.2.1. CHỈ MỤC 27
4.2.2. CHỈ MỤC NGƯỢC (INVERTED INDEX) 28
5. TRUY XUẤT CHỈ MỤC 28
5.1. CONCEPT VECTORS 28
5.2. TRỌNG SỐ (WEIGHT) CHO CÁC VECTOR 30
5.2.1. TF: TERM FREQUENCY 31
5.2.2. IDF: INVERSE DOCUMENT FREQUENCY 31
5.2.3. TF – IDF: TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY 32
5.2.4. BIẾN ĐỔI TÀI LIỆU VÀ CÂU TRUY VẤN THÀNH CÁC VECTOR CÓ TRỌNG SỐ 33
5.2.5. CHUẨN HÓA TRỌNG SỐ 35
6. TÍNH ĐỘ TƯƠNG ĐỒNG VÀ XẾP HẠNG 36
7. VÍ DỤ 39
7.1. LẬP CHỈ MỤC CHO TÀI LIỆU 40
7.2. TRUY VẤN 40
7.2.1. TÍNH TOÁN TRỌNG SỐ 40
7.2.2. TÍNH ĐỘ TƯƠNG ĐỒNG VÀ XẾP HẠNG 42
8. NHẬN XÉT 43

III. LANGUAGE MODEL.....44
1. GIỚI THIỆU.....44
2. KÝ HIỆU.....45
3. TÓM TẮT QUÁ TRÌNH XỬ LÝ.....46
4. LẬP CHỈ MỤC.....47
5. XÂY DỰNG MÔ HÌNH NGÔN NGỮ CHO TÀI LIỆU 50
6. TÍNH $Rel(q|d)$.....53
7. SMOOTHING.....54
7.1. LAPLACE SMOOTHING.....55
7.2. LINEAR INTERPOLATION SMOOTHING.....56
8. NHẬN XÉT.....60
8.1. ƯU ĐIỂM.....60
8.2. NHƯỢC ĐIỂM.....60
IV. CÀI ĐẶT VÀ ĐÁNH GIÁ.....61
1. THIẾT KẾ VÀ CÀI ĐẶT.....61
1.1. VECTOR SPACE MODEL.....61
1.2. LANGUAGE MODEL.....62
2. BỘ DỮ LIỆU CRANFIELD.....62
3. PHƯƠNG PHÁP ĐÁNH GIÁ MÔ HÌNH.....64
3.1. MEAN AVERAGE PRECISION (mAP).....64

3.2. mAP NỘI SUY 65
3.3. THỜI GIAN TRUY XUẤT 66
4. KẾT QUẢ THỰC NGHIỆM CỦA MÔ HÌNH 67
4.1. VECTOR SPACE MODEL 67
4.1.1. KHÔNG SỬ DỤNG SMART 67
4.1.2. SỬ DỤNG SMART 68
4.1.3. KẾT LUẬN 69
4.2. LANGUAGE MODEL 70
4.2.1. UNIGRAM 71
4.2.2. BIGRAM 75
4.2.3. KẾT LUẬN 79
5. SO SÁNH KẾT QUẢ 79
6. SO SÁNH VỚI THƯ VIỆN WHOOSH 81
6.1. GIỚI THIỆU VỀ THƯ VIỆN WHOOSH 81
6.2. SO SÁNH KẾT QUẢ VỚI THƯ VIỆN WHOOSH 82
V. TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN 85
1. TỔNG KẾT 85
2. HƯỚNG PHÁT TRIỂN 86
TÀI LIỆU THAM KHẢO 87

BẢNG PHÂN CÔNG CÔNG VIỆC

Member / **Work**	**Trần Phương Thảo**	**Đặng Thị Tường Vy**	**Nguyễn Hoàng Long**
Xử lý tài liệu và câu truy vấn		✓	
Tìm hiểu và cài đặt Vector Space Model			✓
Tìm hiểu và cài đặt Language Model	✓		
Đánh giá Vector Space Model			✓
Đánh giá Language Model	✓		
So sánh kết quả hai mô hình		✓	
So sánh với thư viện Whoosh		✓	
Mức độ hoành thành (%)	100%	100%	100%

I. GIỚI THIỆU TỔNG QUAN

1. ĐẶT VẤN ĐỀ

Trong thời đại hiện nay, công nghệ thông tin phát triển với tốc độ chóng mặt và Internet đã trở thành một phần không thể thiếu trong cuộc sống của mọi người. Công nghệ đang đóng vai trò ngày càng quan trọng đối với con người. Các lĩnh vực khác nhau trong cuộc sống đều được ảnh hưởng bởi công nghệ. Tuy nhiên, việc đáp ứng yêu cầu người dùng và quản lý khối lượng dữ liệu khổng lồ trên máy tính là một trong những thách thức lớn nhất. Trong những bộ dữ liệu đó, có rất nhiều thông tin quý giá. Tuy nhiên, việc khai thác và sử dụng thông tin đó để hỗ trợ người dùng vẫn là một vấn đề khó khăn. Cần phải tìm cách để xử lý khối lượng dữ liệu lớn và tìm ra những thông tin hữu ích và có ích cho người dùng.

Hình 1: Vấn đề lưu trữ và truy xuất thông tin

Công nghệ xử lý thông tin đã có những tiến bộ và thành tựu đáng kể trong những năm gần đây, giúp giải quyết phần nào các nhu cầu về truy xuất thông tin của con người. Mô hình truy xuất thông tin (IR) đã trở thành một trong những mô hình đang nhận được sự quan tâm rất lớn và được sử dụng trong các hệ thống tìm kiếm phổ biến như Google, AOL, Bing, Yahoo, và nhiều hệ

thống tương tự khác. IR là quá trình tìm kiếm và truy xuất các tài liệu phù hợp nhất với từ khóa hoặc truy vấn được cung cấp bởi người dùng. Tuy nhiên, Mô hình IR đang đối mặt với nhiều thách thức và hạn chế, bao gồm việc không hiệu quả trong tìm kiếm các tài liệu bằng ngôn ngữ tự nhiên do từ ngữ và cách sử dụng từ khác nhau giữa các tài liệu, cũng như khả năng thích ứng với các dạng tài liệu khác nhau như hình ảnh, âm thanh, video và văn bản.

Ngày nay, các chuyên gia đang nỗ lực nghiên cứu và phát triển các giải pháp mới nhằm tối ưu hóa mô hình IR và đáp ứng các nhu cầu của người dùng. Các công nghệ mới như học sâu và học máy đã được áp dụng vào mô hình IR, giúp nâng cao độ chính xác và hiệu quả của quá trình tìm kiếm thông tin. Bên cạnh đó, các phương pháp và công nghệ khác nhau cũng đang được nghiên cứu và áp dụng để cải thiện mô hình IR, như phân tích ngữ nghĩa, xử lý ngôn ngữ tự nhiên, và các phương pháp khai thác dữ liệu khác. Tóm lại, mô hình truy xuất thông tin đang phát triển và được nghiên cứu một cách tích cực để đáp ứng nhu cầu của người sử dụng. Những tiến bộ và công nghệ mới sẽ giúp nâng cao hiệu quả và độ chính xác của mô hình truy xuất thông tin, từ đó mang lại trải nghiệm tốt nhất cho người dùng.

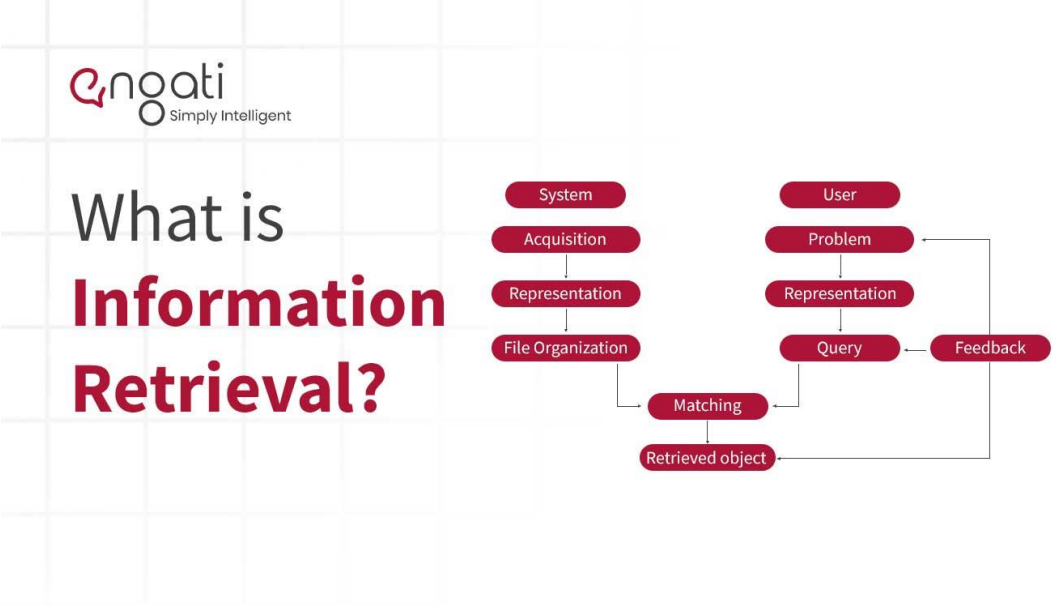
2. KHÁI NIỆM “TRUY XUẤT THÔNG TIN”

Truy xuất thông tin (IR) là một chương trình phần mềm được sử dụng để tổ chức, lưu trữ, truy xuất và đánh giá thông tin từ các kho tài liệu, đặc biệt là thông tin văn bản. Mục đích của IR là hỗ trợ người dùng tìm kiếm thông tin mà họ yêu cầu, thông báo về sự tồn tại và vị trí của các tài liệu có thể bao gồm các thông tin cần thiết. Các tài liệu thỏa mãn yêu cầu của người sử dụng được gọi là các tài liệu liên quan. Truy xuất thông tin là hoạt động thu thập tài nguyên hệ thống thông tin có liên quan đến nhu cầu thông tin từ tập hợp các

nguồn thông tin. Các tìm kiếm có thể dựa trên tìm kiếm toàn văn bản hoặc các chỉ mục.

Mô hình truy xuất thông tin (IR) là một quá trình phức tạp trong đó các trang web hoặc tài liệu được chọn và xếp hạng dựa trên truy vấn của người dùng. Các hệ thống IR thường sử dụng các hàm so khớp để so sánh giữa truy vấn và các tài liệu trong bộ sưu tập và trả về giá trị trạng thái truy xuất (RSV) cho mỗi tài liệu. RSV được sử dụng để xác định thứ tự sắp xếp các tài liệu liên quan trong kết quả tìm kiếm.

Để mô tả nội dung tài liệu, phần lớn các hệ thống IR sử dụng một tập hợp các bộ mô tả được gọi là từ trong từ vựng V. Từ vựng V là một tập hợp các từ khóa mà được sử dụng để mô tả và đại diện cho nội dung của các tài liệu trong bộ sưu tập. Mỗi từ trong V có thể được gán một trọng số để biểu thị tầm quan trọng của từ đó trong bộ sưu tập. Trọng số này có thể được xác định bằng nhiều cách khác nhau, ví dụ như sử dụng tần suất xuất hiện của từ trong bộ sưu tập hoặc sử dụng các phương pháp học máy để đánh giá tầm quan trọng của từ.



Hình 2: Khái niệm truy xuất thông tin

Tóm lại, mô hình truy xuất thông tin sử dụng các hàm so khớp để xếp hạng các tài liệu liên quan dựa trên truy vấn của người dùng. Từ vựng V được sử dụng để mô tả nội dung của các tài liệu và có thể được gán trọng số để đánh giá tầm quan trọng của từ trong bộ sưu tập. Các phương pháp khác nhau có thể được sử dụng để xác định trọng số của từ trong V.

2.1. SO KHỚP TÀI LIỆU

Trong mô hình truy xuất thông tin, việc so khớp tài liệu là quá trình so sánh truy vấn của người dùng với các tài liệu trong bộ sưu tập để tìm ra các tài liệu liên quan nhất. Các hệ thống truy xuất thông tin sử dụng các hàm so khớp để đánh giá độ tương đồng giữa truy vấn và các tài liệu trong bộ sưu tập.

Chức năng so khớp tài liệu truy vấn trong mô hình IR được xác định theo cách sau:

- Ước tính khả năng liên quan của người dùng đối với từng trang và truy vấn liên quan đến bộ sưu tập tài liệu đào tạo query.
- Trong không gian vectơ, hàm tương tự giữa các truy vấn và tài liệu được tính toán.

2.2. INPUT VÀ OUTPUT

Input và output là hai thành phần quan trọng của hệ thống truy xuất thông tin (IR). Input là truy vấn của người dùng, trong khi output là danh sách các tài liệu liên quan được trả về cho người dùng. Cả input và output đều có ảnh hưởng đáng kể đến hiệu suất và độ chính xác của hệ thống truy xuất thông tin.

- Input: Một bộ ngữ liệu (corpus) các tài liệu văn bản, một câu truy vấn (query) của người dùng dưới dạng văn bản.

- Output: Một tập xếp hạng (ranked list) các văn bản mà được cho là phù hợp (relevant) với câu truy vấn (query)

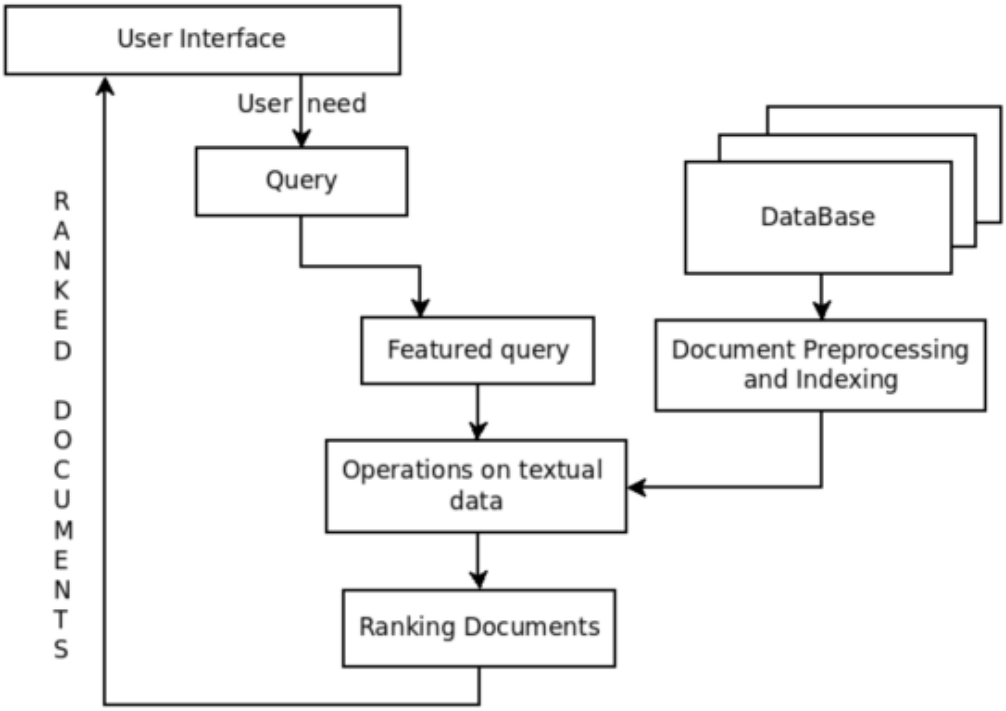
Input và output đều ảnh hưởng đến hiệu suất và độ chính xác của hệ thống truy xuất thông tin. Input không rõ ràng hoặc không chính xác sẽ làm giảm khả năng truy xuất thông tin, trong khi output không liên quan đến truy vấn sẽ giảm độ chính xác và độ tin cậy của hệ thống. Do đó, cần cải thiện khả năng nhập truy vấn và tối ưu hóa kết quả truy xuất thông tin để đáp ứng nhu cầu của người dùng.

3. QUY TRÌNH TRUY XUẤT THÔNG TIN

Quy trình truy xuất thông tin (Information Retrieval - IR) là quá trình tìm kiếm và thu thập thông tin từ các nguồn khác nhau để đáp ứng nhu cầu thông tin của người dùng. Quy trình truy xuất thông tin bao gồm các hoạt động chính như sau:

- Quá trình truy xuất thông tin bắt đầu khi người dùng tạo bất kỳ truy vấn nào vào hệ thống thông qua một số giao diện đồ họa được cung cấp.
- Các truy vấn do người dùng tạo ra thể hiện cho nhu cầu về thông tin của người dùng (information need).
- Trong Truy xuất thông tin, một truy vấn có thể phù hợp với nhiều tài liệu. Trong trường hợp này tài liệu có độ liên quan cao sẽ xem xét để thực hiện các bước tiếp theo.
- Sự khác biệt chính giữa truy xuất thông tin và tìm kiếm cơ sở dữ liệu là quá trình xếp hạng các tài liệu liên quan được thực hiện để tìm ra tài liệu liên quan nhất với câu truy vấn đã cho.
- Sau khi truy vấn được xử lý bởi hệ thống và trả về một kết quả R, thì kết quả đó sẽ được trả lại cho người dùng.

- Quá trình lặp lại và kết quả được sửa đổi cho đến khi người dùng hài lòng với những họ thực sự đang tìm kiếm.



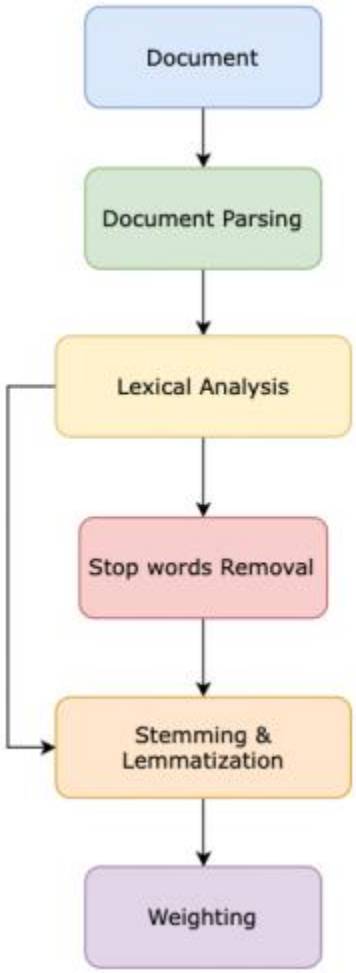
Hình 3: Quá trình xử lý truy vấn

4. CÁC BƯỚC TIỀN XỬ LÝ

Truy xuất thông tin là quá trình thu thập thông tin liên quan từ một tập hợp lớn các cơ sở dữ liệu. Tiền xử lý văn bản đóng một vai trò quan trọng trong việc truy xuất thông tin bằng cách trích xuất thông tin liên quan. Để đạt được điều này, phương pháp tiền xử lý văn bản được đề xuất. Trong giai đoạn tiền xử lý, hệ thống truy xuất thông tin sẽ tạo ra biểu diễn bên trong từng tài liệu thông qua quá trình đánh chỉ mục. Đầu tiên, tập tài liệu văn bản được tiền xử lý bằng một số phương pháp tự động như phân tích từ vựng, loại bỏ stopword và lấy gốc từ (stemming) để chuyển đổi tài liệu thành dạng văn bản đơn giản.

Sau khi qua giai đoạn tiền xử lý, ta thu được một bộ dữ liệu mới chứa các từ đã được chuyển đổi về dạng gốc (stemming) hoặc từ điển (lemmatization).

Bộ dữ liệu này giúp đơn giản hóa việc so sánh các từ với nhau trong quá trình tìm kiếm và tăng tính nhất quán của kết quả truy xuất thông tin.



Hình 4: Lưu đồ thực hiện các bước tiền xử lý

Tiền xử lý dữ liệu là một bước quan trọng trong quá trình truy xuất thông tin, với nhiều lợi ích như sau:

- *Tối ưu hóa tốc độ truy xuất:* Tiền xử lý giúp làm giảm kích thước dữ liệu và đơn giản hóa cấu trúc văn bản, từ đó tăng tốc độ truy xuất thông tin.
- *Cải thiện độ chính xác:* Tiền xử lý giúp loại bỏ các từ không có ý nghĩa (stopwords) và chuyển đổi từ các dạng khác nhau của từ về dạng gốc, giúp cải thiện độ chính xác của kết quả truy xuất thông tin.

- *Tăng tính nhất quán:* Tiền xử lý giúp đồng bộ hóa các từ và thuật ngữ trong cùng một lĩnh vực, giúp tăng tính nhất quán của kết quả truy xuất thông tin.
- *Giảm thiểu nhiễu:* Tiền xử lý giúp loại bỏ các từ không cần thiết, đồng thời giảm thiểu nhiễu và tăng tính chính xác của kết quả truy xuất thông tin.
- *Tăng khả năng tìm kiếm:* Tiền xử lý giúp chuyển đổi các từ và thuật ngữ như đồng nghĩa, từ viết tắt hoặc từ viết sai chính tả về dạng chính xác, từ đó tăng khả năng tìm kiếm và thu thập thông tin.

Do đó, tiền xử lý thông tin là một bước quan trọng để cải thiện hiệu suất và độ chính xác của hệ thống truy xuất thông tin.

5. LẬP CHỈ MỤC

Lập chỉ mục (Indexing) là quá trình xây dựng một bộ chỉ mục cho tập văn bản trong hệ thống truy xuất thông tin (IR). Bộ chỉ mục này được sử dụng để tìm kiếm và truy xuất các văn bản phù hợp với các yêu cầu của người dùng. Quá trình lập chỉ mục bao gồm các bước sau đây:

- *Tiền xử lý dữ liệu:* Trước khi tạo chỉ mục, ta thực hiện tiền xử lý dữ liệu để chuẩn hóa các từ và loại bỏ những từ không cần thiết. Những từ này có thể là các từ dừng (stopwords), các từ trùng lặp hoặc các từ không mang nhiều ý nghĩa.
- *Phân tích văn bản:* Sau khi đã được tiền xử lý, các văn bản sẽ được phân tích để tìm ra các thuật ngữ (term) phù hợp. Các thuật ngữ này sẽ được thêm vào bộ chỉ mục.
- *Lập chỉ mục:* Từ các term được tìm ra, chúng ta sẽ lập được một bộ chỉ mục. Bộ chỉ mục này sẽ gồm các thông tin như tên tài liệu, vị trí của term trong tài liệu, tần suất xuất hiện của các term đó trong văn bản, …

- *Lưu trữ chỉ mục:* Bộ chỉ mục sẽ được lưu trữ để sử dụng trong quá trình truy xuất thông tin. Hệ thống truy xuất thông tin sẽ sử dụng bộ chỉ mục này để tìm kiếm các tài liệu phù hợp với yêu cầu của người dùng.

Quá trình lập chỉ mục là một trong những bước quan trọng trong hệ thống truy xuất thông tin và đóng vai trò quan trọng trong việc cải thiện chất lượng kết quả truy xuất. Bộ chỉ mục được tạo ra trong quá trình lập chỉ mục chứa các thuật ngữ và thông tin liên quan đến các văn bản trong tập dữ liệu. Bộ chỉ mục này sẽ được sử dụng để tìm kiếm và truy xuất các văn bản phù hợp với các yêu cầu của người dùng.

Quá trình lập chỉ mục đóng vai trò quan trọng trong việc đạt được hiệu suất và hiệu quả cao trong hệ thống truy xuất thông tin. Một bộ chỉ mục tốt cải thiện tính chính xác và tốc độ của quá trình truy xuất thông tin, đồng thời tăng hiệu quả và độ phủ của hệ thống.

6. XỬ LÝ TRUY VẤN

Xử lý truy vấn (query processing): là một quá trình quan trọng trong hệ thống truy xuất thông tin, có nhiệm vụ xử lý câu truy vấn của người dùng và trả về các kết quả phù hợp nhất. Quá trình này bao gồm nhiều bước, bao gồm phân tích, tìm kiếm và sắp xếp kết quả trả về.

- *Phân tích truy vấn (query parsing):* là một bước quan trọng trong quá trình truy xuất thông tin, trong đó truy vấn của người dùng sẽ được phân tích để tìm ra các thuật ngữ (term)liên quan đến yêu cầu truy vấn của họ. Các thuật ngữ này sau đó sẽ được sử dụng để tìm kiếm các tài liệu phù hợp trong bộ chỉ mục.
- *Tạo câu truy vấn (query formulation):* các term đã được tìm ra sẽ được sử dụng để tạo ra câu truy vấn (query) để tìm kiếm các tài liệu phù hợp trong bộ chỉ mục. Quá trình tạo câu truy vấn có thể được thực hiện bằng

nhiều cách khác nhau, tùy thuộc vào mục đích của truy vấn và cách mà người dùng muốn kết hợp các thuật ngữ. Một cách phổ biến để tạo câu truy vấn là sử dụng các toán tử logic (AND, OR, NOT) để kết hợp các thuật ngữ với nhau.

- *Tìm kiếm trong bộ chỉ mục (search in the index):* Sau khi tạo câu truy vấn, hệ thống sử dụng nó để tìm kiếm các tài liệu phù hợp trong bộ chỉ mục và trả về kết quả cho người dùng. Quá trình tìm kiếm và trả về kết quả là bước quan trọng trong quá trình truy xuất thông tin.
- *Xếp hạng kết quả (ranking results):* Sau khi tìm kiếm và tìm ra các tài liệu phù hợp, hệ thống sẽ tiến hành xếp hạng chúng theo độ phù hợp với truy vấn. Việc xếp hạng được thực hiện bằng cách tính điểm cho các tài liệu phù hợp sử dụng các thuật toán sử dụng TF-IDF hoặc BM25. Quá trình xếp giúp đảm bảo rằng kết quả trả về cho người dùng là các tài liệu phù hợp nhất và được sắp xếp theo độ ưu tiên phù hợp với truy vấn của họ.
- *Trả về kết quả (returning results):* Cuối cùng, sau khi đã xếp hạng các kết quả phù hợp, hệ thống sẽ trả về cho người dùng các kết quả phù hợp nhất, thường là các tài liệu được xếp hạng cao nhất. Các kết quả này có thể được hiển thị dưới dạng danh sách, tùy thuộc vào cách thiết lập của hệ thống.

Các bước xử lý truy vấn trên đóng vai trò rất quan trọng trong việc cải thiện hiệu quả tìm kiếm thông tin trên các nền tảng trực tuyến. Khi người dùng nhập vào một câu truy vấn, hệ thống sẽ tiến hành các bước xử lý như tách từ, phân tích cú pháp, phân tích ngữ nghĩa, và tìm kiếm thông tin trong cơ sở dữ liệu.

Các bước xử lý này giúp hệ thống hiểu được ý định của người dùng, phân tích và xác định những từ khóa quan trọng, và tìm kiếm thông tin phù hợp với

câu truy vấn. Nhờ đó, hệ thống có thể cung cấp cho người dùng những kết quả truy vấn chính xác và phù hợp nhất với nhu cầu của họ.

7. ỨNG DỤNG CỦA TRUY XUẤT THÔNG TIN

Trong thời đại kỹ thuật số hiện nay, Công nghệ thông tin đang có ảnh hưởng đến nhiều khía cạnh của xã hội trên toàn cầu. Nó thúc đẩy sự phát triển của thương mại, kinh doanh, cải thiện giáo dục và chăm sóc sức khỏe, đồng thời tạo ra một môi trường giao tiếp thuận tiện giữa các bên liên quan.

Một trong những điểm nhấn của công nghệ thông tin là hệ thống truy xuất thông tin. Nó cho phép xã hội truy xuất dữ liệu liên quan đến một tìm kiếm cụ thể thông qua một hệ thống hiệu quả. Từ đó, các tổ chức và cá nhân có thể nhanh chóng tìm kiếm và sử dụng thông tin cần thiết để đưa ra các quyết định quan trọng và tiến hành các hoạt động kinh doanh, giáo dục hay chăm sóc sức khỏe một cách hiệu quả hơn. Với hệ thống truy xuất thông tin, xã hội có thể truy xuất dữ liệu liên quan đến một tìm kiếm cụ thể thông qua một hệ thống hiệu quả.

Thu thập, phân loại và tích lũy thông tin để phân phối và sử dụng theo yêu cầu của người dùng. Phạm vi áp dụng bao gồm nhiều lĩnh vực khác nhau như thư mục, tài liệu kỹ thuật, tiền lệ tư pháp, kiểm tra bằng sáng chế, điều kiện thị trường và thông tin giao thông. Hệ thống thường được chia thành bộ phận thu thập, phân loại, tích lũy thông tin và bộ phận tìm kiếm thông tin theo yêu cầu. Truy xuất thông tin đang là lĩnh vực ứng dụng hiệu quả nhất của sức mạnh máy tính, có rất nhiều ứng dụng trong các lĩnh vực khác nhau như công cụ tìm kiếm trên web, nghiên cứu khoa học, quản lý dữ liệu và thông tin khách hàng…

- *Thông tin trong y tế:* Công nghệ IR giúp các chuyên gia y tế tiếp cận nhanh chóng với các thông tin liên quan đến bệnh lý, thuốc và các

phương pháp điều trị hiệu quả nhất. Kết quả truy xuất được cung cấp từ IR giúp cho các chuyên gia y tế có thể cập nhật kiến thức mới nhất và áp dụng chúng vào công tác khám và điều trị bệnh nhân.

- *Thư viện số:* IR được áp dụng trong thư viện để hỗ trợ người dùng trong việc tìm kiếm các tài liệu phù hợp như sách, bài báo và các nguồn tài nguyên khác.
- *Tìm kiếm trong thương mại điện tử:* Trong các trang web mua hàng trực tuyến, Information retrieval giúp người dùng tìm kiếm những sản phẩm phù hợp với nhu cầu của họ.
- *Các công cụ tìm kiếm:* Một số công cụ tìm kiếm nổi tiếng phát triển như Google, Bing, Yahoo sử dụng IR để truy xuất thông tin phù hợp từ web.
- *Tìm kiếm tài liệu pháp lý:* Các kỹ thuật truy xuất thông tin có thể được áp dụng để tìm kiếm các tài liệu pháp lý phù hợp như hồ sơ vụ kiện, điều lệ và các quy định liên quan.
- *Tìm kiếm bằng sáng chế:* IR được sử dụng trong các cơ sở dữ liệu sáng chế để tìm kiếm các sáng chế phù hợp cho nghiên cứu và sáng tạo
- *Lọc thông tin:* Các phương pháp truy xuất thông tin có thể được áp dụng để lọc và phân loại thông tin dựa trên sở thích và quan tâm của người dùng.
- *Phân tích hành vi người dùng:* Sử dụng các kỹ thuật truy xuất thông tin, phân tích mạng xã hội có thể dùng để phân tích nội dung trên các mạng xã hội, giúp xác định xu hướng, tâm trạng và hành vi của người dùng.

Như vậy, IR đóng một vai trò quan trọng và có nhiều ứng dụng đa dạng trong nhiều ngành công nghiệp và lĩnh vực nghiên cứu. Nó được sử dụng trong các lĩnh vực như tìm kiếm web, phân tích dữ liệu, giải trí số và các ứng dụng

y tế. Ngoài ra, IR còn được dùng để tìm kiếm thông tin trong giáo dục, quản lý tri thức và trong lĩnh vực kinh doanh. Các ứng dụng của IR đang phát triển rất nhanh và có tiềm năng trong nhiều lĩnh vực khác nhau, đóng vai trò quan trọng trong việc cung cấp thông tin đa dạng và hiệu quả cho người dùng.

II. VECTOR SPACE MODEL

1. GIỚI THIỆU

Vector space model là một trong những mô hình phổ biến trong hệ thống truy vấn thông tin (Information Retrieval). Đây là một mô hình đại số để biểu diễn các tài liệu văn bản và bất kỳ đối tượng khác nói chung dưới dạng các vector chỉ mục. Nó được sử dụng trong chắt lọc thông tin, truy xuất thông tin, lập chỉ mục và xếp hạng mức độ phù hợp. Lần đầu tiên nó được sử dụng trong Hệ thống Truy xuất Thông tin SMART.

Mô hình Vector Space (Mô hình không gian vector) được sử dụng để truy vấn thông tin biểu thị các tài liệu và truy vấn dưới dạng các vectơ trọng số. Mỗi trọng số là thước đo tầm quan trọng của một thuật ngữ chỉ mục trong các tài liệu hoặc câu truy vấn. Trọng số của từ khóa chỉ mục được tính toán dựa trên tần suất xuất hiện của các thuật ngữ chỉ mục trong tài liệu, câu truy vấn hoặc tập tài liệu. Sau khi biểu diễn các tài liệu và truy vấn trong không gian vector, ta có thể tính toán độ tương đồng giữa chúng bằng cách sử dụng các phép tính đại số tuyến tính như khoảng cách cosine giữa 2 vector. Độ tương đồng này có thể được sử dụng để xếp hạng theo thứ tự giảm dần của các tài liệu và trả về kết quả tìm kiếm cho người dùng.

2. KÝ HIỆU

Ta có một số ký hiệu thường gặp trong Vector space model như sau:

- *Vocabulary:* là tập hợp các từ khóa (term) riêng biệt trong tất cả các document ($V = \{t_1,\ t_2,\ t_3, \dots, t_N\}$) trong đó N là tổng số lượng các term
- *Document*: tài liệu $d_i = \{d_{i1},\ d_{i2},\ d_{i3}, \dots, d_{iN}\}$ trong đó N là số lượng term trong document
- *Query (câu truy vấn)*: $q = t_1, t_2, \dots, t_N$ trong đó N là tổng số lượng term trong q
- *Collection* là tập hợp các document $C = \{d_1,\ d_2,\ d_3, \dots, d_N\}$ trong đó N là số lượng document
- *Rel(q, d)* độ liên quan giữa document d và query q
- *Rep(d)* hàm biểu diễn document d
- *Rep(q)* hàm biểu diễn query q
- *Dictionary* đây là một cấu trúc dữ liệu ta tạo ra để hỗ trợ việc lập chỉ mục giúp cải thiện tốc độ truy xuất và hiệu xuất truy xuất

3. Ý TƯỞNG

Vector space model hoạt động dựa trên việc vector hóa các document và query trong không gian vector đa chiều với mỗi chiều tương ứng với một term và sử dụng độ đo tương đồng (similarity) như là Euclid hoặc cosine để xếp hạng các kết quả và trả kết quả về cho người dung theo xếp hạng.

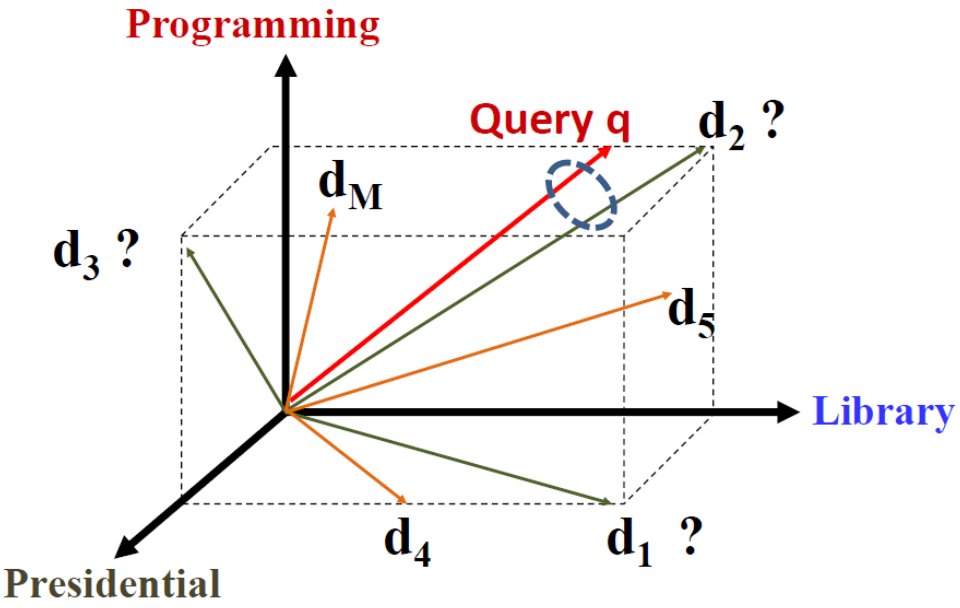
Cụ thể, ý tưởng của mô hình Vector Space là biểu diễn văn bản và các câu truy vấn dưới dạng Vector có trọng số $Rep(d)$ của docs và $Rep(q)$ của query sẽ cho kết quả là các vector

Mỗi trọng số là thước đo tầm quan trọng của một thuật ngữ chỉ mục trong một tài liệu hoặc một truy vấn tương ứng.

Tại thời điểm truy xuất, các tài liệu được xếp hạng theo độ tương đồng của query với từng documents theo công thức để tìm ra docs vào phù hợp nhất với query:

$$Similarity(rep(d), rep(q))$$

Chúng ta có thể dễ dàng hình dung qua hình vẽ sau:



Hình 5: Minh họa mô hình vector space

Xét ở ví dụ trên ta có:

Term: Programming, Presidential, Library (các term chưa được xử lý)

C = {d_1, d_2, d_3, d_4}

Query: q

Ta có thể sử dụng khoảng cách Euclid hoặc độ đo cosine để đo độ tương đồng (trong ví dụ trên là giữa truy vấn q và tài liệu d2)

4. PHÂN TÍCH TÀI LIỆU VÀ LẬP CHỈ MỤC

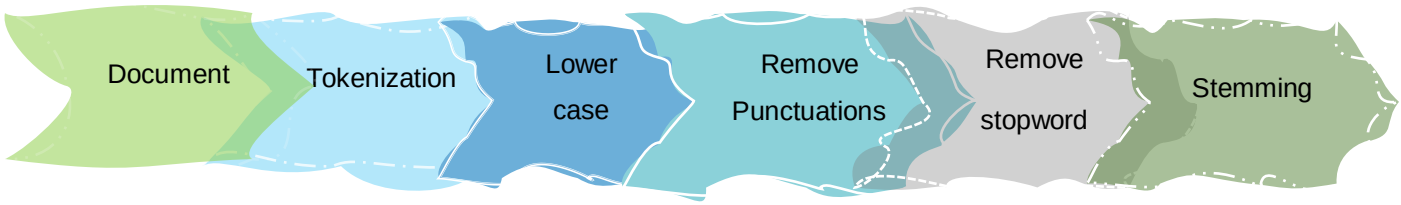
4.1. PHÂN TÍCH TÀI LIỆU

Phân tích tài liệu là bước đầu tiên, cần thiết cho việc thực hiện truy vấn, hiểu đơn giản thì nó là bước tiền xử lý văn bản. Điều này giúp cho văn bản, câu truy vấn được tối ưu giúp cho việc tính toán nhanh hơn và chính xác hơn.

Vì văn bản vốn dĩ là không có cấu trúc và nó tồn tại rất nhiều từ stopword và các từ gây nhiễu khác, không có nhiều giá trị ngữ nghĩa nên việc xử lý văn bản là rất cần thiết.

Ví dụ: trong V chỉ có "USA" nhưng query lại là "U.S.A", nếu không xử lý để chuyển "U.S.A" về thành "USA" thì sẽ không truy vấn được document mong muốn.

Flow chart:



Hình 6: Các bước xử lý truy vấn – VSM.

4.1.1. TOKENIZATION

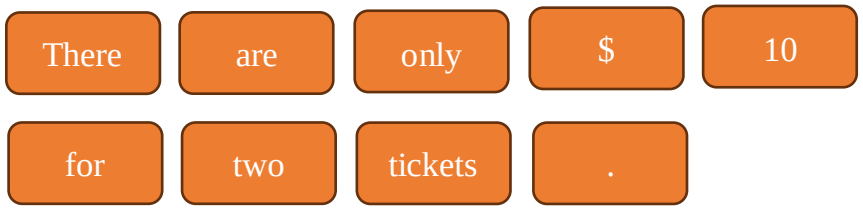
Tokenization (tách từ) là việc ta tách các từ, văn bản thành các đoạn ngắn hơn hay còn gọi là token, trong tiếng anh thì ta tách và phân chúng dựa trên khoảng trắng.

Một token là một thể hiện của một chuỗi các kí tự, được gom nhóm lại thành một đơn vị ngữ nghĩa hữu ích cho việc xử lý. Một type là một lớp tất cả các token có chuỗi kí tự giống nhau. Một term là một type đã được thêm vào dictionary của hệ thống truy vấn thông tin.

Chẳng hạn nếu document được lập chỉ mục là “to sleep perchance to dream”, thì sẽ có 5 token, nhưng chỉ có 4 type (vì có 2 thể hiện của “to”) nên cũng chỉ có 4 term, và nếu “to” bị loại bỏ như là stopword thì cuối cũng chỉ còn lại 3 term: “sleep”, “perchance” và “dream”.

Ví dụ: cho một tài liệu như sau: “There are only $10 for two tickets.”

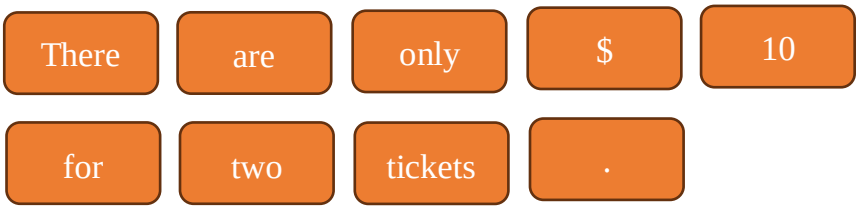
Lúc này tách theo khoảng trắng ta sẽ được:



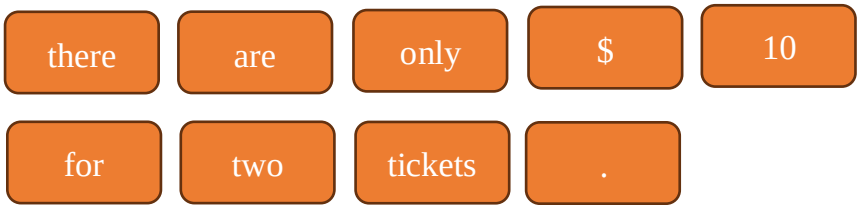
4.1.2. LOWERCASE

Dùng để tránh trường hợp case-sensitive (phân biệt chữ hoa và chữ thường). Ví dụ: “Vietnam” và “vietnam” là hai term hoàn toàn khác nhau.

Tài liệu khi chưa lowercase:



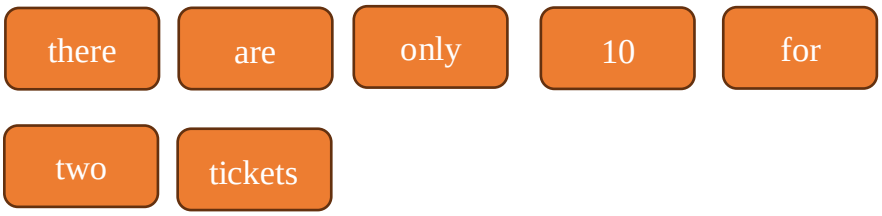
Tài liệu sau khi lowercase



4.1.3. LOẠI BỎ PUNCTUATIONS

Punctuations là tập hợp các ký tự sau: !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~ Punctuations thường không đóng góp gì nhiều vào việc truy vấn, thậm chí là đôi khi chúng còn ảnh hưởng nặng tới việc truy vấn, ví dụ như nếu không loại bỏ punctuations thì "math." và "math" là hai term hoàn toàn khác nhau.

Trong ví dụ trên khi ta áp dụng thao tác punctuations thì ta sẽ thu được kết quả như sau:



4.1.4. LOẠI BỎ STOPWORD VÀ SỐ NGUYÊN

Stopword là những từ xuất hiện nhiều trong văn bản nhưng nó lại không mang nhiều ý nghĩa, ví dụ như các từ "a, an, is, …" những từ này đôi khi tồn tại chỉ đem đến cho mô hình của chúng ta những tính toán không cần thiết vì chúng làm tang kích thước vector và ma trận nên việc loại bỏ stopword là điều cũng rất quan trọng.

Cùng còn tùy vào ngữ cảnh, hoàn cảnh mà ta sẽ lựa chọn những tập stopword cho phù hợp.

Trong ví dụ trên ta khi ta thực hiện thao tác remove stopword thì ta sẽ thu được kế quả như sau:

4.1.5. STEMMING

Stemming (lấy gốc từ) là thao tác biến đổi từ về dạng ngữ pháp gốc của nó. Thao tác này giúp giảm đi số lượng từ vựng mang ngữ pháp gốc giống nhau nhưng bị biến đổi sang dạng ngữ pháp khác. Đồng thời

thao tác này giúp nhóm các từ có chung ngữ nghĩa để người dùng không phải xác định quá cụ thể trong truy vấn.

Trong lý thuyết ta có hai sự lựa chọn trong việc chọn ra phương pháp để lấy gốc từ

- *Stemming:*

  Đối với stemming, phương pháp này cho rằng biến thể của từ được sinh ra từ việc thêm vào một vài ký tự vào cuối từ đó. Ví dụ như chúng ta thấy các từ như walked, walking, walks chỉ khác nhau là ở những ký tự cuối cùng, bằng cách bỏ đi các hậu tố -ed, -ing hoặc -s, chúng ta sẽ được từ nguyên gốc là walk. Người ta gọi các bộ xử lý stemming là Stemmer.

- *Lematizing:*

  Lemmatization, nó cải tiến hơn stemming trong việc biến đổi các từ về dạng ngữ pháp gốc của nó một cách thông minh, và nó xử lý được các trường hợp đã nêu ở trên mà stemming không thực hiện được. Cụ thể, lemmatization sẽ xử lý thông minh hơn 24 bằng một bộ từ điển hoặc một bộ ontology nào đó. Điều này sẽ đảm bảo rằng các từ như “goes”, “went” và “go” sẽ chắc chắn có kết quả trả về là như nhau. Kể các từ danh từ như mouse, mice cũng đều được đưa về cùng một dạng như nhau. Người ta gọi bộ xử lý lemmatization là Lemmatizer.

  Nhược điểm của lemmatization là tốc độ xử lý khá chậm vì phải thực hiện tra cứu từ trong cơ sở dữ liệu. Trong các ứng dụng xử lý NLP mà cần độ chính xác cao hơn và thời gian không quan trọng, người ta có thể sử dụng Lemmatization. Qua đó ta có thể thấy, việc phân tích tài liệu là bước cần thiết

phải có để giảm bớt đi số lượng từ trong tập tài liệu. Giảm bớt đi các từ không quan trọng, và giữ lại các từ có độ quan trọng cao. Việc giảm bớt đi số lượng từ này sẽ rất hữu dụng trong việc lập chỉ mục vì thời gian lập chỉ mục cũng sẽ giảm đi đáng kể sau khi phân tích tài liệu.

Trong ví dụ trên ta thấy sau khi hoàn thành thao tác stemming thì ta thu được tài liệu gồm:

4.2. LẬP CHỈ MỤC

4.2.1. CHỈ MỤC

Là phương pháp thực hiện quét một lần trên các file văn bản và lưu lại danh sách các thuật ngữ (từ, cụm từ) có trong file kèm thông tin đi kèm với thuật ngữ (term) (vị trí, tần suất, độ quan trọng, ...). Khi đó các thông tin này sẽ được tổ chức theo một cấu trúc dữ liệu riêng và được gọi là chỉ mục.

Lập chỉ mục hiểu đơn giản ở đây là ta đang tạo ra một cấu trúc dữ liệu mà nó lưu trữ những thông tin cần thiết phục vụ cho việc truy vấn, giúp cho việc thuận tiện cho việc truy vấn, các thao tác truy vấn sau này sẽ được thực hiện trên chỉ mục thay vì thực hiện trực tiếp truy vấn trên tập tài liệu.

	t_1	t_2	…	t_m
d_1	1	0	…	1
…	…	…	…	1
d_n	1	1	…	1

Bảng 1: Ví dụ về chỉ mục.

Việc lập chỉ mục diễn ra nhanh chóng vì các từ khóa được thêm vào liên tục khi tìm thấy. Nhưng bù lại, việc tìm kiếm diễn ra khá chậm, tốn nhiều thời gian xử lý vì kích thước chỉ mục lớn.

4.2.2. CHỈ MỤC NGƯỢC (INVERTED INDEX)

Chỉ mục ngược (Inverted Index), mỗi thuật ngữ sẽ tương ứng với danh sách các tài liệu chứa nó. Nó là một cấu trúc dữ liệu lưu trữ ánh xạ từ từ khóa sang tài liệu hoặc tập hợp tài liệu, tức là chỉ cho người tìm kiếm biết là từ đó xuất hiện trong tập tài liệu nào.

Quy trình xây dựng chỉ mục ngược

- Quét tài liệu, chuẩn bị danh sách các từ duy nhất (Tức là từ nào đã được duyệt qua và đưa lên chỉ mục rồi thì không xét tới nữa).
- Chuẩn bị một danh sách các chỉ mục của tất cả các từ duy nhất và ánh xạ chúng để tìm kiếm tài liệu.
- Lặp lại các bước trên đến khi hoàn thành tất cả các tài liệu

t_1	d_1	d_5	d_{50}	d_{68}
t_2	d_1	d_3	d_{98}	
…				
t_n	d_1	d_{45}		

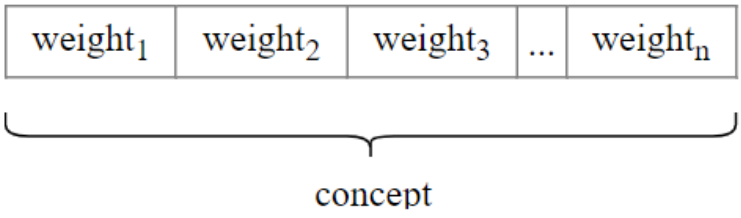
Bảng 2: Ví dụ về chỉ mục ngược.

5. TRUY XUẤT CHỈ MỤC

Trong quá trình truy xuất chỉ mục, những thứ ta cần quan tâm sẽ bao gồm: Concept vectors và trọng số (weight) cho các vectors đó.

5.1. CONCEPT VECTORS

Là hình thức mà ta dùng để biểu diễn tài liệu (document) và truy vấn (query), việc truy xuất chỉ mục cũng sẽ được thực thi dựa trên các concept vectors này.



Hình 7: Mô tả concept vector.

- Mỗi concept vector biểu diễn một chiều.
- Cần định nghĩa trọng số (weights) cho từng chiều của không gian vector.

Ví dụ concept vector cho document sau: Hàng đầu tiên là danh sách các cuốn sách giáo khoa môn toán của Bộ giáo dục và đào tạo Việt Nam. Cột đầu tiên là các từ, và các ô thể hiện sự xuất hiện hay không xuất hiện của các từ đó trong các cuốn sách giáo khoa.

	Toán 12	Toán 11	Toán 10	Toán 9	Toán 8	Toán 7
Hình học không gian	1	1	1	0	0	0
Đệ quy	1	1	0	0	0	0
Đạo hàm	1	0	0	0	0	0
Ma trận	1	0	1	1	0	0
Lượng giác	0	1	1	0	1	1

Và mỗi tài liệu được biểu diễn dưới dạng vector ví dụ toán 11

$$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

	Toán 12	Toán 11	Toán 10	Toán 9	Toán 8	Toán 7
Hình học không gian	7	1	10	0	0	0
Đệ quy	14	15	0	0	0	0
Đạo hàm	3	0	0	0	0	0
Ma trận	1	0	4	2	0	0
Lượng giác	0	5	5	0	1	1

Với mỗi tài liệu được biểu diễn dưới dạng vector, ví dụ toán 11

$$\begin{bmatrix} 1 \\ 15 \\ 0 \\ 0 \\ 5 \end{bmatrix}$$

5.2. TRỌNG SỐ (WEIGHT) CHO CÁC 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. Từ hiếm thì quan trọng hơn những từ có tần suất xuất hiện cao.

Đối với term frequency (TF), thì những từ càng xuất hiện nhiều thì có điểm càng cao, còn những từ hiếm thì điểm xếp hạng lại thấp hơn. Do đó chúng ta cần một cách đánh giá khác với các từ hiếm, vì nó sẽ mang nhiều thông tin hơn là những từ phổ biến trong văn bản. Ví dụ trong một tập hợp các tài liệu về nghành công nghiệp ô-tô, thì từ khóa “ô-tô” sẽ có khả năng có mặt hầu hết trong tất cả các tài liệu. Để hạn chế nhược điểm này, người ta giới thiệu cơ chế để giảm thiểu sự ảnh hưởng của việc này

và tăng tính chính xác khi quyết định sự phù hợp của tài document và query. Ý tưởng là giảm trọng số của từ nào có document frequency cao. Có nhiều kỹ thuật tính trọng số: TF, IDF, TF-IDF, ...

5.2.1. TF: TERM FREQUENCY

Term frequency (tần số thuật ngữ) là một khái niệm trong lĩnh vực xử lý ngôn ngữ tự nhiên và khai thác dữ liệu văn bản. Nó được sử dụng để đo tần suất xuất hiện của một từ hoặc cụm từ trong một tài liệu hoặc tập các tài liệu.

Cụ thể, term frequency là số lần xuất hiện của một từ hoặc cụm từ trong một tài liệu, chia cho tổng số từ trong tài liệu đó. Kết quả thu được cho biết độ quan trọng của từ hoặc cụm từ đó trong tài liệu đó. Công thức tính term frequency (TF) của một từ tiếng Anh là:

$$TF = \frac{\text{số lần xuất hiện của từ đó trong văn bản}}{\text{tổng số từ trong văn bản}}$$

$$tf(t,d) = \begin{cases} 1 + \log\big(f(t,d)\big), nếu\ f(t,d) > 0 \\ 0, nếu\ f(t,d) \leq 0 \end{cases}$$

Với $1 + \log(f(t,d))$ để tránh trường hợp $tf(t,d)$ sẽ cho cùng một kết quả với $f(t,d) = 1$ và $f(t,d) = 0$ nếu chỉ dung mỗi $\log(f(t,d))$. Với công thức trên thì giá trị $f(t,d) \in [0, +\infty)$.

Ví dụ, trong câu "I love to eat pizza", số lần xuất hiện của từ "pizza" là 1, và tổng số từ trong câu là 5, do đó term frequency của từ "pizza" trong câu đó là $\frac{1}{5} = 0.2$.

5.2.2. IDF: INVERSE DOCUMENT FREQUENCY

Inverse document frequency idf là một khái niệm trong xử lý ngôn ngữ tự nhiên và khai thác dữ liệu văn bản. idf được sử dụng để đo độ quan trọng của một từ hoặc cụm từ trong một tập các tài liệu.

Cụ thể, idf tính toán độ hiếm của một từ hoặc cụm từ trong toàn bộ tập các tài liệu. Nó được tính bằng cách lấy tổng số tài liệu trong tập dữ liệu, chia cho số tài liệu trong đó từ hoặc cụm từ đó xuất hiện ít nhất một lần, rồi lấy logarithm cơ số 2 của kết quả đó. Công thức tính IDF của một từ tiếng Anh là:

$$idf = \log\left(\frac{N}{n}\right)$$

Trong đó, N là số tài liệu trong tập dữ liệu và n là số tài liệu trong đó từ đó xuất hiện ít nhất một lần.

idf được sử dụng trong kỹ thuật $tf - idf$ (term frequency-inverse document frequency) để đánh giá độ quan trọng của một từ hoặc cụm từ trong một tài liệu.

Kết hợp tf và idf, $tf - idf$ tính toán giá trị trọng số cho mỗi từ trong một tài liệu, dựa trên sự xuất hiện của từ đó trong tài liệu và độ quan trọng của từ đó trong toàn bộ tập các tài liệu.

$df(t)$ là tần suất document của term t (số lượng document chứa term t).

$df(t)$ là độ đo nghịch đảo của mức độ thông tin của term t.

$df(t) \in [1, N]$, do vocabulary V được tạo thành từ các term trong collection C, nên tất nhiên các term trong V sẽ nằm trong một document $d \in C$, từ đây suy ra được miền giá trị của $df(t)$

Và công thức tính idf là:

$$idf = \log(\frac{N}{df(t,d)})$$

5.2.3. TF – IDF: TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY

$tf - idf$ là viết tắt của cụm từ Term Frequency-Inverse Document Frequency, là một kỹ thuật trong xử lý ngôn ngữ tự nhiên và khai thác dữ liệu văn bản để đánh giá độ quan trọng của một từ hoặc cụm từ trong một tài liệu.

$tf - idf$ kết hợp hai thành phần quan trọng:

- Term frequency (tf): đo tần suất xuất hiện của một từ hoặc cụm từ trong tài liệu.
- Inverse document frequency (idf): đo độ hiếm của một từ hoặc cụm từ trong toàn bộ tập các tài liệu.

Công thức tính toán $tf - idf$ cho một từ tiếng Anh là:

$$tf - idf = tf * idf$$

Trong đó, tf được tính như đã giải thích trong câu trả lời trước đó. idf được tính bằng cách lấy logarithm cơ số 2 của tổng số tài liệu trong tập dữ liệu, chia cho số tài liệu trong đó từ đó xuất hiện ít nhất một lần.

Kết quả $tf - idf$ cho biết độ quan trọng của một từ hoặc cụm từ trong một tài liệu. Nó thường được sử dụng trong các ứng dụng tìm kiếm, phân loại văn bản, và khai thác dữ liệu để giúp tìm kiếm và phân loại nội dung văn bản chính xác hơn.

5.2.4. BIẾN ĐỔI TÀI LIỆU VÀ CÂU TRUY VẤN THÀNH CÁC VECTOR CÓ TRỌNG SỐ

Nếu có để ý ta sẽ thấy trong các bước tiền xử lý, hay biến đổi thành dạng vector đều nhắc tới cả các document và query là vì chúng ta đều muốn các document và query đều được biểu diễn thành vector bằng cùng một cách duy nhất, vì nếu không làm vậy thì các term giữa các document có thể sẽ không giống các term của query, từ đó làm ảnh

hưởng tới kết quả truy vấn. Ví dụ với hai document, một query cụ thể và một danh sách stopwords đã được rút ngắn lại từ danh sách stopwords trong tiếng Anh của thư viện NLTK.

Giả sử 2 document không loại bỏ stopwords, còn query sẽ có 2 phiên bản, không loại bỏ stopwords và loại bỏ stopwords:

doc1: what have i done and why who s this

doc2: what have you done

query1 (không loại bỏ stopwords): what have i done

query2 (loại bỏ stopwords): done

Rõ ràng với phiên bản query1 thì sẽ xếp hạng doc1 cao hơn doc2 vì doc1 chứa tất cả term trong query1, còn với phiên bản query2 sau khi đã qua xử lí thì chỉ còn truy vấn với mỗi một term ‘done’, mà trong cả doc1 và doc2 lại chứa term ‘done’, nên kết quả truy vấn đã bị ảnh hưởng, không còn tốt như với phiên bản query1. Đó là lí do mà chúng ta sẽ muốn các document và query được xử lí giống nhau. Đây chỉ là ví dụ chứng minh việc document và query nên được xử lí giống nhau, chứ 35 không chứng minh loại bỏ stopwords là việc không nên làm, việc có nên loại bỏ stopwords hay không đã được trả lời ở phần Inverse document frequency idf.

Ví dụ: Biểu diễn câu truy vấn sau thành dạng vector có trọng số:

query: teacher student supervise

- Tiền xử lý: query: teacher student supervis Loại bỏ các token không có trong dictionary: vì không có token nào nằm ngoài dictionary nên các token vẫn được giữ nguyên.
- Lập chỉ mục: Vì mỗi lần truy vấn chỉ có một query nên không cần DocID, do đó chúng ta chỉ cần lập cột token và bỏ cột DocID

Token
teacher
student
supervis

Sau khi sắp xếp lại ta thu được bảng như sau

Token
student
supervis
teacher

Thống kê tf: Đối với các document thì bước này gọi là "Tạo dictionary", bây giờ để cho khỏi phải tạo ra thêm một cấu trúc dữ liệu dictionary cho query thì chúng ta sẽ gọi bước này là "Thống kê tf". Do đã bỏ cột docID, bây giờ chúng ta cũng bỏ luôn cột doc freq và thay đổi cột thành tf.

Term weighting (gán trọng số)

Tf: Bước này vẫn áp dụng công thức như bình thường

$$tf(t,d) = \begin{cases} 1 + \log\big(f(t,d)\big), nếu\ f(t,d) > 0 \\ 0, nếu\ f(t,d) \leq 0 \end{cases}$$

Cuối cùng ta thu được:

Token	idf	tf	tf-idf
student	0.12	1	0.12
supervis	0.3	1	0.3
teacher	0.6	1	0.6

5.2.5. CHUẨN HÓA TRỌNG SỐ

Chuẩn hóa trọng số hay còn gọi là normalize, đây là bước chuyển các trọng số về miền giá trị [0,1] để nhằm hạn chế sự xuất hiện của các term trong tài liệu dài và các term trong tài liệu ngắn và đồng thời giúp chuyển các vector của các document và query thành các vector cùng phương và có độ dài là 1

Công thức cơ bản:

$$w_{ik} = \frac{tf_{ik} * idf_{ik}}{\sqrt{\sum_{k=1}^{t}(tf_{ik})^2 * (idf_{ik})^2}}$$

Ngoài công thức chuẩn hóa trọng số trên thì còn một số công thức chuẩn hóa trọng số khác phổ biến như là chuẩn hóa trọng số theo hệ thống S.M.A.R.T.

$$w_{kd} = \frac{collect_k * fred_{kd}}{norm}$$

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times\mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0,\log\frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$ (Section 6.4.4)
b (boolean)	$\begin{cases}1 & \text{if } \mathrm{tf}_{t,d}>0\\0 & \text{otherwise}\end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha<1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$				

6. TÍNH ĐỘ TƯƠNG ĐỒNG VÀ XẾP HẠNG

Sau khi các document và query đã được biểu diễn thành các vector hay các point (điểm, cả vector và điểm đều nằm trong không gian |V| chiều, |V| là kích thước của vocabulary V hay số lượng term trong V). Vector được hình thành bởi hai yếu tố: chiều và độ lớn, chúng ta chỉ mới có một điểm, vậy thì điểm còn lại để tạo nên chiều và độ lớn cho vector là điểm gì, điểm đó nằm ở đâu? Câu trả lời là gốc tọa độ V (0, 0, ..., |V|) như chúng ta có thể thấy rõ ở hình phía trên. Sau khi đã gán trọng số cho các vector document và vector query

xong, bước tiếp theo là tính độ liên quan giữa các document với query. Vậy chúng ta phải xếp hạng các document như thế nào với query trong không gian |V| chiều này? Câu trả lời là, vì chúng ta đã biểu diễn các document và query trong không gian |V| chiều này, nên chúng ta có thể xếp hạng các document dựa theo với query trong không gian này, và proximity = similarity giữa những vector. Bây giờ chúng ta sẽ tiến hành định lượng proximity giữa các vector.

6.1. ĐỘ TƯƠNG ĐỒNG COSINE

Độ đo tương đồng cosine (cosine similarity) là một phương pháp đo sự tương đồng giữa hai vector trong không gian nhiều chiều. Nó được sử dụng rộng rãi trong xử lý ngôn ngữ tự nhiên để đánh giá mức độ tương đồng giữa hai văn bản.

Công thức tính toán độ đo tương đồng cosine giữa hai vector A và B là:

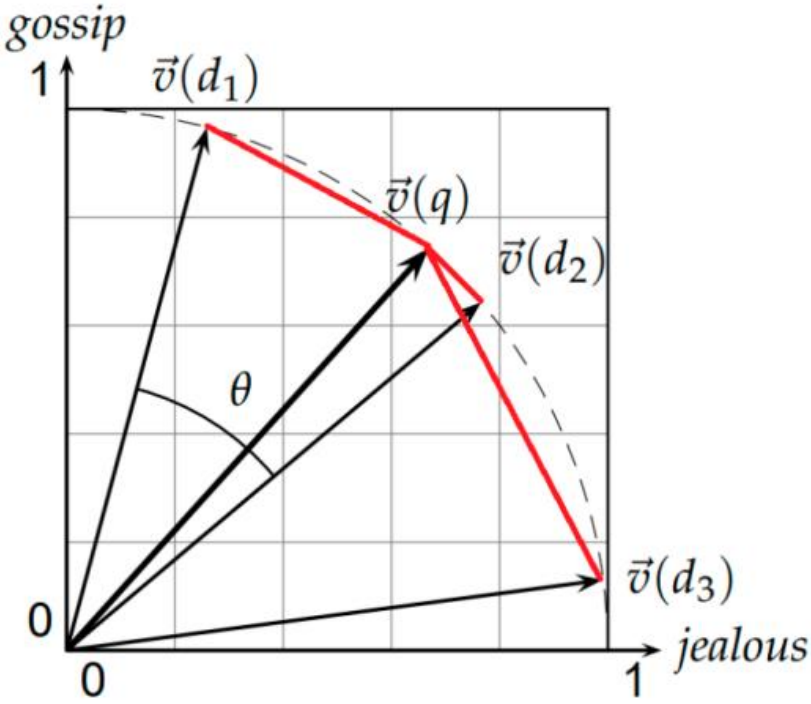
$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\cdot\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

Trong đó, A và B là hai vector đang được so sánh, "." là phép nhân vô hướng của hai vector, ||A|| và ||B|| là độ dài của vector A và B, tương ứng.

Kết quả độ đo tương đồng cosine nằm trong khoảng từ -1 đến 1, trong đó 1 đại diện cho sự tương đồng hoàn toàn giữa hai vector, 0 đại diện cho sự không tương đồng hoặc không có mối quan hệ giữa hai vector, và -1 đại diện cho sự tương phản hoàn toàn giữa hai vector.

Độ đo tương đồng cosine thường được sử dụng để đánh giá mức độ tương đồng giữa các văn bản, phân loại văn bản, tìm kiếm văn bản và các ứng dụng khác trong xử lý ngôn ngữ tự nhiên.

6.2. KHOẢNG CÁCH EUCLID



Hình 8: Minh họa kết quả giữa khoảng cách Euclid và Cosine

Độ đo Euclid (Euclidean distance) là một phương pháp đo khoảng cách giữa hai điểm trong không gian nhiều chiều. Nó được sử dụng rộng rãi trong các ứng dụng liên quan đến phân loại, gom nhóm, và xử lý hình ảnh.

Công thức tính toán độ đo Euclid giữa hai điểm P và Q trong không gian nhiều chiều có n chiều là:

Kết quả độ đo Euclid là một số dương, thể hiện khoảng cách giữa hai điểm trong không gian nhiều chiều. Nó thường được sử dụng để đánh giá mức độ tương đồng giữa các điểm trong không gian nhiều chiều, đo khoảng cách giữa các vector và phân loại dữ liệu.

Tuy nhiên, độ đo Euclid có một số hạn chế, đặc biệt là khi áp dụng cho các không gian có số chiều lớn, nó có thể dẫn đến hiện tượng "vấn đề

chiều" (curse of dimensionality), khiến cho các kết quả đo lường trở nên không hiệu quả.

Công thức tính khoảng cách Euclid là:

$$dis(q,d) = \sqrt{\sum_{i=1}^{|V|}(d_i - q_i)^2}$$

7. VÍ DỤ

Cho tập tài liệu như sau:

Tài liệu:

d1: Ship ocean of wood

d2: Boat in ocean

d3: Wood is trees

Yêu cầu: Sử dụng mô hình Vector Space:

1. Lập chỉ mục cho tài liệu trên.
2. Thực hiện truy vấn:

q: ocean ocean wood

Đầu tiên thì ta cần tiền xử lý cho tập tài liệu và truy vấn, tức là làm các thao tác:

- Tách từ.
- Chuyển về chữ thường.
- Loại bỏ punctuations.
- Loại bỏ từ dừng (Stopwords) và số thực.
- Lấy gốc từ (stemming).

Tập tài liệu và truy vấn sau khi tiền xử lý là:

d1: ship ocean wood

d2: boat ocean

d3: wood tree

d4: ocean ocean wood

7.1. LẬP CHỈ MỤC CHO TÀI LIỆU

Lấy tập tài liệu và truy vấn sau khi tiền xử lý để lập chỉ mục, ta thu được chỉ mục như sau:

Term	DocID
ship	1
ocean	1
wood	1
boat	2
ocean	2
wood	3
tree	3

7.2. TRUY VẤN

Để thực hiện truy vấn, như đã trình bày ở các phần trước, ta cần phải thực hiện các bước:

- Đối với tài liệu:
  - Chọn weighting: tính $tf, idf, tf - idf$
  - Chuẩn hóa các vector tài liệu
- Đối với truy vấn:
  - Chọn weighting: tính $tf, idf, tf - idf$
  - Chuẩn hóa các vector truy vấn

7.2.1. TÍNH TOÁN TRỌNG SỐ

Đầu tiên là tính toán trọng số của các từ. Trong ví dụ này, công thức tf, tf- idf được sử dụng là những công thức của các phần cùng tên đã trình bày trong mục 5.2, riêng đối với idf thì ta sử dụng công thức:

$$idf(t) = \log(\frac{N}{df(t)})$$

Với N: Tổng số lượng tài liệu, $df(t)$ số lượng tài liệu chứa từ t

Từ tập tài truy vấn đã được xử lý ta xây dựng được bảng sau

Terms	Q	D1	D2	D3
boat	0	0	1	0
ocean	2	1	1	0
ship	0	1	0	0
tree	0	0	0	1
wood	1	1	0	1

Các ô xuất hiện chữ số 0 tức là từ đó không xuất hiện trong tài liệu/ truy vấn đang xét, và ngược lại 1 tức là có xuất hiện.

Từ bảng trên hoặc từ chỉ mục ngược đã xây dựng ở yêu cầu 1, ta có thể đếm được số lượng tài liệu chứa các từ đã liệt kê như sau:

Terms	Df(t)
boat	1
ocean	2
ship	1
tree	1
wood	2

Vì số lượng tài liệu là 3, nên N = 3, khi đó ta tính được thành phần 1 23(5) và sau đó lấy logarit cho thành phần này, ta thu được kết quả idf cho từng từ:

Terms	D/df_i	Idf = $\log(D/df_i)$
boat	3/1=3	0.4771

ocean	3/2 = 1.5	0.1761
ship	3/1 = 3	0.4771
tree	3/1 = 3	0.4771
wood	3/2 = 1.5	0.1761

Khi mà đã có tf và idf, ta sẽ tìm được trọng số (weight) cho từng từ trong cả truy vấn và tài liệu:

Terms	Q	D1	D2	D3
boat	0	0	0.4771	0
ocean	0.3522	0.1761	0.1761	0
ship	0	0.4771	0	0
tree	0	0	0	0.4771
wood	0.1761	0.1761	0	0.1761

Lúc này ta sẽ thu được vector của từng document và query

$$\vec{Q} = (0, 0.3522, 0, 0, 0.1761)$$

$$\overrightarrow{D1} = (0, 0.1761, 0.4771, 0, 0.1761)$$

$$\overrightarrow{D2} = (0.4771, 0.1761, 0, 0, 0)$$

$$\overrightarrow{D3} = (0, 0, 0, 0.4771, 01761)$$

7.2.2. TÍNH ĐỘ TƯƠNG ĐỒNG VÀ XẾP HẠNG

Trong ví dụ này, ta sẽ tính độ tương đồng và xếp hạng theo Cosine Similarity theo công thức:

$$Cosine(\vec{Q}, \vec{D}) = \frac{\vec{Q} * \vec{D}}{|\vec{Q}| * |\vec{D}|}$$

Kết quả sau khi chuẩn hóa vector ta thu được các tham số sau:

$$|\overrightarrow{D1}| = \sqrt{0.1761^2 + 0.4471^2 + 0.1761^2} = 0.5382|$$

$$|\overrightarrow{D2}| = \sqrt{0.4471^2 + 0.1761^2} = 0.5085|$$

$$|\overrightarrow{D3}| = \sqrt{0.4471^2 + 0.1761^2} = 0.5085|$$

$$\overrightarrow{|Q|} = \sqrt{0.3522^2 + 0.1761^2} = 0.3937$$

Ta tính giá trị:

$$\vec{Q} * \overrightarrow{D1} = (0.3522 * 0.1761 + 0.1761 * 0.1761) = 0.093$$

$$\vec{q} * \overrightarrow{d2} = 0.3522*0.4771= 0.062$$

$$\vec{q} * \overrightarrow{d3} = 0.1761*0.1761= 0.031$$

Từ đó ta tính được độ đo cosine như sau:

$$\cos(\vec{q} * \overrightarrow{d1}) = \frac{0.093}{0.5381 + 0.3937} = 0.099$$

$$cos(\vec{q} * d2) = \frac{0.062}{0.5085 + 0.3937} = 0.068$$

$$cos(\vec{q} * \overrightarrow{d3}) = \frac{0.031}{0.5085 + 0.3937} = 0.034$$

Tại đây ta thấy d1 có độ tương đồng với q cao nhất nên ta sẽ trả kết quả về là d1. Thứ tự ranking là: d1, d2, d3.

8. NHẬN XÉT

Tổng kết lại về mô hình Vector Space, ta rút ra được một số khả năng mà mô hình làm được cho việc truy vấn:

- Có thể xếp hạng các kết quả trả về.
- Các document và query đều được biểu diễn dưới dạng vector.
- Mỗi chiều của vector tương ứng với một term trong vocabulary.
- Mỗi chiều của vector là một số thực, phản ánh độ quan trọng của term trong document, query.
- Tính độ liên quan dựa vào công thức tính Cosine giữa hai vector.
- Mỗi vector (document, query, một câu văn, ...) đều có thể được so sánh độ liên quan với nhau.

Thông qua nghiên cứu về mô hình Vector Space, một số ưu điểm và nhược điểm của mô hình được thể hiện như sau.

8.1. ƯU ĐIỂM

- Đã được nghiên cứu nhiều.
- Dễ hiểu và dễ cài đặt.
- Có thể xếp hạng kết quả trả về dựa trên độ liên quan.
- Có nhiều cách tính trọng số term.

8.2. NHƯỢC ĐIỂM

- Không quan tâm tới thứ tự xuất hiện của term.
- Các chiều trong không gian là mỗi từ khoá, không đảm bảo về mặt ngữ nghĩa.
- Các vector chỉ nằm trong phần dương của không gian.
- So khớp dựa vào từ khoá, nếu tài liệu và truy vấn không có từ khoá chung thì độ tương đồng bằng 0.

III. LANGUAGE MODEL

1. GIỚI THIỆU

Khi người dùng có nhu cầu tìm kiếm thông tin thường sẽ tạo các câu truy vấn có chứa các từ liên quan đến nhu cầu thông tin cần kiếm và tài liệu trả về thường sẽ chứa một số lượng lớn các từ nằm trong câu truy vấn của người dùng. Ví dụ khi tìm kiếm thông tin liên quan đến trường Đại học Công nghệ Thông tin thì các tài liệu trả về sẽ chứa một hoặc nhiều cụm từ “trường Đại học Công nghệ Thông tin”.

Language Model dựa vào ý tưởng trên để tìm kiếm tài liệu liên quan đến nhu cầu thông tin. Trong Language Model, thì câu truy vấn được tạo ra một cách ngẫu nhiên, và mục tiêu Language Model là biểu diễn câu truy vấn thông qua một mô hình xác suất. Sử dụng Language Model thì ta có thể tính được

xác suất của một câu truy vấn được sinh ra và các tài liệu liên quan có quan hệ với câu truy vấn đó như thế nào thông qua xác suất.

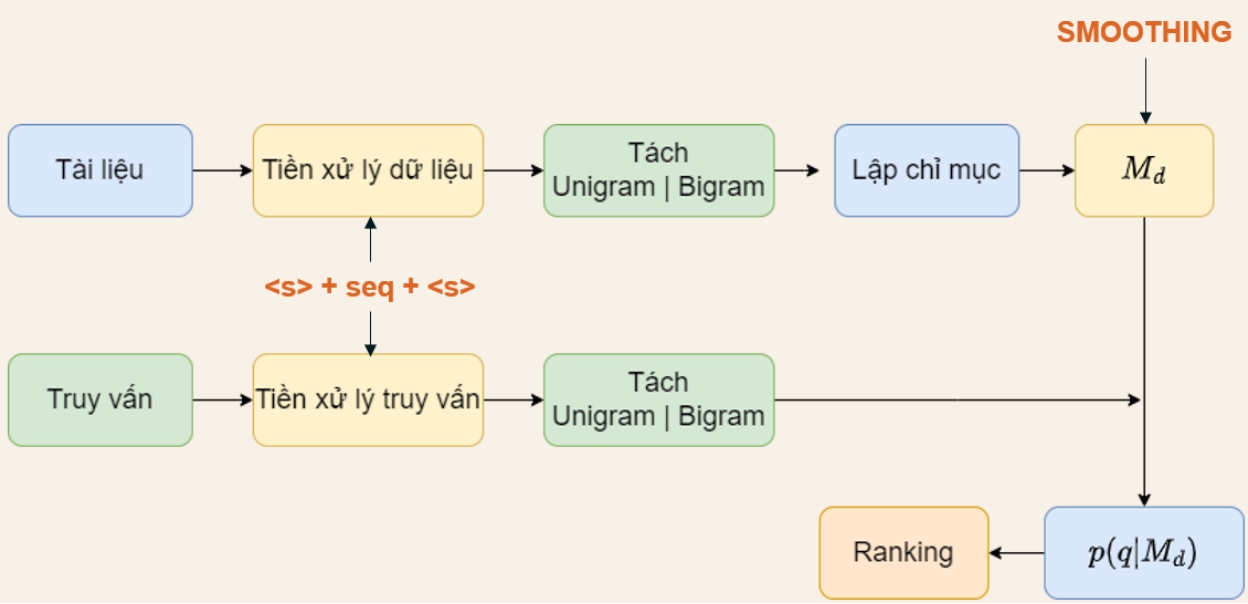
Hay nói cách khác Language Model tính toán xác suất mà một tài liệu có thể sinh ra câu truy vấn của người dùng. Nếu xác suất đó càng cao thì khả năng tài liệu đó có độ liên quan càng cao đến câu truy vấn, tương đương với đáp ứng được nhu cầu của thông tin cần tìm kiếm của người dùng. Language Model cũng dùng xác suất đó để xếp hạng các tài liệu trả về, xác suất càng cao thì xếp hạng độ liên quan của tài liệu trả về tương ứng với câu truy vấn càng cao.

2. KÝ HIỆU

Ta có một số ký hiệu thường gặp trong Vector space model như sau:

- *Vocabulary (từ vựng)*: $V = \{t_1,\ t_2,\ t_3, \dots, t_N\}$), N là số lượng các term riêng biệt trong tất cả document.
- *Document (tài liệu)*: $d_i = \{d_{i1},\ d_{i2},\ d_{i3}, \dots, d_{iN}\}$ với $d_{ij} \in V$ và N là số lượng term trong document
- *Query (câu truy vấn)*: $q = \{q_1, q_2, q_3, \dots, q_M\}$ với $q_i \in V$ và M là số lượng các term trong query.
- *Collection (tập tài liệu)*: $C = \{D_1,\ D_2,\ D_3, \dots, D_k\}$, k là số lượng document trong collection.
- M_d: mô hình ngôn ngữ của tài liệu d.
- $Rel\ (q, d)$: độ liên quan giữa document d và query q.
- $p(q|M_d)$: xác suất mô hình ngôn ngữ của tài liệu d sinh ra truy vấn q.
- *Dictionary (từ điển):* Cấu trúc dữ liệu lưu trữ các term, ứng với mỗi term t là số lượng document chứa term t và các cặp $docID$ - trọng số, trọng số ở đây chính là trọng số của term xuất hiện trong document được gán chỉ số $docID$ ($docID$ dùng để phân biệt giữa các document).

3. TÓM TẮT QUÁ TRÌNH XỬ LÝ

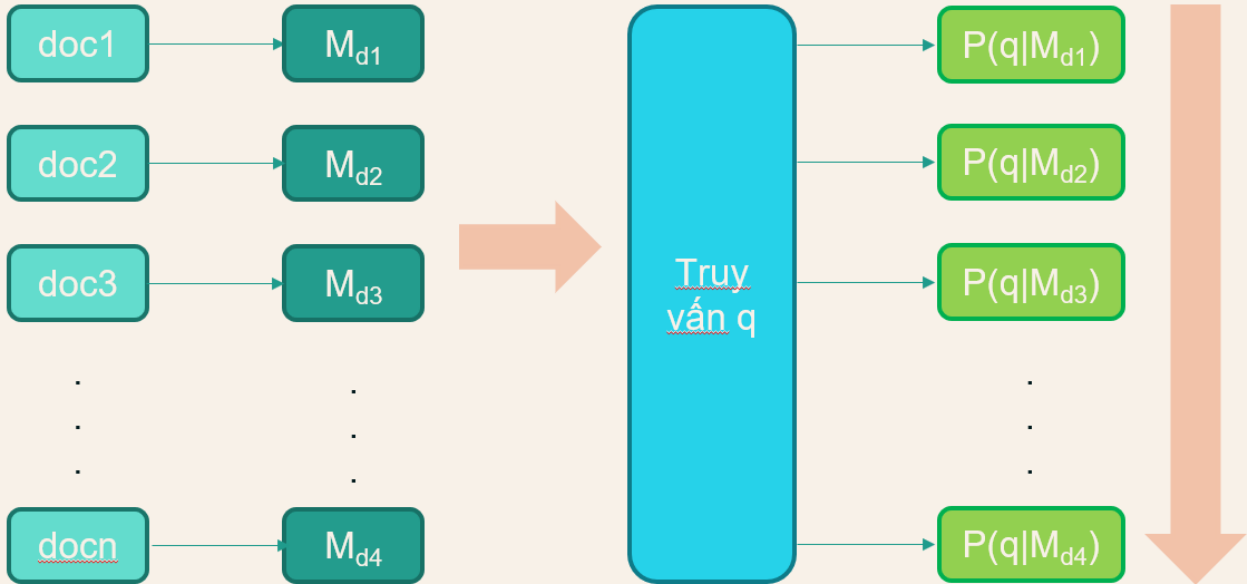


Hình 9: Các bước thực hiện trong Language Model

Theo bước tiền xử lý dữ liệu đã đề cập ở trên, tài liệu và câu truy vấn sẽ qua các bước tiền xử lý dữ liệu theo thứ tự sau:

Hình 10: Các bước xử lý tài liệu và câu truy vấn trong Language Model

Với mỗi tài liệu, Language Model sẽ tạo mô hình ngôn ngữ M_d tương ứng với tài liệu đó. Sau đó, ta tính $Rel(q, M_d)$ bằng cách tính $p(q|M_d)$ của mỗi tài liệu D với truy vấn q. Cuối cùng là xếp hạng $p(q|M_d)$ theo thứ tự cao đến thấp, $p(q|M_d)$ càng cao thì $Rel(q, M_d)$ càng cao.



Hình 11: Quá trình truy vấn trong Language Model.

4. LẬP CHỈ MỤC

Language Model lập chỉ mục bằng cách lập chỉ mục ngược (Inverted Index). Bảng chỉ mục ngược được thiết kế như sau:

- *Cột đầu:* các term (tổ hợp term) riêng biệt được lấy tách ra từ corpus.
- *Mỗi hàng:* các DocID của tài liệu chứa term đó và tần số xuất hiện của term đó trong tài liệu tương ứng.

Ví dụ:

- $t_1 : d_1 : 2$ là term t_1 có tần số xuất hiện trong tài liệu d_1 là 2.
- $t_2, t_1 : d_1 : 2$ là cặp term t_2, t_1 có tần số xuất hiện trong tài liệu d_1 là 2.

Quy trình xây dựng chỉ mục:

- Quét tài liệu, chuẩn bị danh sách các từ duy nhất (Tức là từ nào đã được duyệt qua và đưa lên chỉ mục rồi thì không xét tới nữa).
- Chuẩn bị một danh sách các chỉ mục của tất cả các từ duy nhất và ánh xạ chúng để tìm kiếm tài liệu.

- Thêm $< DocID: 1 >$ của tài liệu đang xét vào hàng có chỉ mục của term tương ứng nếu như term đó lần đầu xuất hiện trong tài liệu nếu không thì cộng 1 vào số lần xuất hiện.
- Lặp lại các bước trên cho tất cả các tài liệu.

Để minh họa cho quá trình lập chỉ mục, chúng ta sẽ lấy 3 tài liệu trong tập Cranfield thực hiện ví dụ.

experimental investigation of the aerodynamics of a wing in a slipstream . an experimental study of a wing in a propeller slipstream was made in order to determine the spanwise distribution of the lift increase due to slipstream at different angles of attack of the wing and at different free stream to slipstream velocity ratios . the results were intended in part as an evaluation basis for different theoretical treatments of this problem . the comparative span loading curves, together with supporting evidence, showed that a substantial part of the lift increment produced by the slipstream was due to a /destalling/ or boundary layer control effect . the integrated remaining lift increment, after subtracting this destalling lift, was found to agree well with a potential flow theory . an empirical evaluation of the destalling effects was made for the specific configuration of the experiment .
simple shear flow past a flat plate in an incompressible fluid of small viscosity . in the study of high speed viscous flow past a two dimensional body it is usually necessary to consider a curved shock wave emitting from the nose or leading edge of the body . consequently, there exists an inviscid rotational flow region between the shock wave and the boundary layer . such a situation arises, for instance, in the study of the hypersonic viscous flow past a flat plate . the situation is somewhat different from prandtl's classical boundary layer problem . in prandtl's original problem the inviscid free stream outside the boundary layer is irrotational while in a hypersonic boundary layer problem the inviscid free stream must be considered as rotational . the possible effects of vorticity have been recently discussed by ferri and libby . in the present paper, the simple shear flow past a flat plate in a fluid of small viscosity is investigated . it can be shown that this problem can again be treated by the boundary layer approximation, the only novel feature being that the free stream has a constant vorticity . the discussion here is restricted to two dimensional incompressible steady flow .
the boundary layer in simple shear flow past a flat plate . the boundary layer equations are presented for steady incompressible flow with no pressure gradient .

Bảng 3: 3 tài liệu trong tập Cranfield.

Trước khi lập chỉ mục, đối với mô hình ngôn ngữ sẽ có thêm một bước là ta sẽ thêm vào đầu và cuối mỗi tài liệu "$< s >$" và "$</s >$" đại diện cho bắt đầu và kết thúc câu. Mục đích của việc thêm vào là để có thể tính được xác suất của t_1 của mỗi tài liệu đối với mô hình Bigram vì mô hình Bigram sẽ tách

các cặp term mà term đứng đầu sẽ bị bỏ qua vì trước term đó không có term nào khác. Vì thế thêm 2 kí tự trên để không bỏ sót term nào trong tài liệu.

Đầu tiên thì ta cần tiền xử lý cho tập tài liệu và truy vấn, tức là làm các thao tác sau:

- Tách từ.
- Chuyển về chữ thường.
- Loại bỏ punctuations.
- Loại bỏ từ dừng (Stopwords) và số thực.
- Lấy gốc từ (stemming).
- Thêm 2 kí tự "$< s >$" và "$</s >$" vào đầu và cuối mỗi tài liệu.

Lập bảng chỉ mục đối với Unigram:

Term	$\mathbf{< ID(doc) : freq >}$		
<s>	0: 1	1: 1	2: 1
experiment	0: 2		
investig	0: 1	1: 1	
aerodynam	0: 1		
	…		
pressur			2: 1
gradient			2: 1
</s>	0: 1	1: 1	2: 1

Bảng 4: Chỉ mục đối với Language Model – Unigram.

Lập bảng chỉ mục đối với Bigram:

Term	$\mathbf{< ID(doc) : freq >}$
<s>, experiment	0: 1
experiment, investig	0: 1
investing, aerodynam	0: 1

aerodynam, wing	
…	
pressure, gradient	2: 1
gradient, </s>	2: 1

Bảng 5: Chỉ mục đối với Language Model – Bigram.

5. XÂY DỰNG MÔ HÌNH NGÔN NGỮ CHO TÀI LIỆU

Việc tạo mô hinh ngôn ngữ cho tài liệu thực chất là việc gán trọng số là phân phối xác suất cho các term trong tài liệu. Công thức dưới đây biểu diễn phân phối xác suất cho mỗi term trong tài liệu.

$$M_d = p(t_1, t_2, t_3, \dots, t_n)$$

$$= p(t_1)p(t_2|t_1)p(t_3|t_2, t_1) \dots p(t_n|t_{n-1}, t_{n-2}, \dots, t_1)$$

Trong đó:

- $p(t_1)$ là xác suất của t_1 xuất hiện trong tài liệu d.
- $p(t_2|t_1)$ là xác suất của t_2 xuất hiện trong tài liệu theo thứ tự t_1 đứng trước t_2.
- $p(t_3|t_2, t_1)$ là xác suất của t_3 xuất hiện trong tài liệu theo thứ tự t_1 đứng trước t_2 và t_2 đứng trước t_3.
- $p(t_n|t_{n-1}, t_{n-2}, \dots, t_1)$ là xác suất của t_n xuất hiện trong tìa liệu theo thứ tự t_1 đứng trước t_2 và t_2 đứng trước t_3 và …và t_{n-1} đứng trước t_n.

Nhìn vào các tham số trên của công thức, ta có thể thấy với term nằm càng gần cuối thì trọng số của term đó càng phức tạp vì phụ thuộc vào các term xuất hiện trước term đó. Độ phức tạp cao và hiệu suất tính toán thấp khi tính toán các xác suất có điều kiện của các term trong tài liệu kể cả đối với tài liệu nhỏ, mà thực tế chúng ta thường làm việc với các bộ dữ liệu lớn nên không thể sử dụng công thức trên. Để có thể tính xấp xỉ $p(t_1, t_2, t_3, \dots, t_n)$ thì trong

thực tế thường sử dụng mô hình N-gram. Mô hình N-gram được định nghĩa theo công thức sau:

$$M_d = p(t_1, t_2, t_3, \dots, t_n)$$
$$\approx p(t_1)p(t_2|t_1)p(t_3|t_2, t_1) \dots p(t_n|t_{n-1}, t_{n-2}, \dots, t_{n-N})$$

Công thức N-gram khác với công thức gốc về số lượng phân phối xác suất có điều kiện của term được xét. Với công thức gốc, ta phải tính xác suất có điều kiện của term t_i mà t_{i-1} đứng trước t_i và t_{i-2} đứng trước t_{i-1} và xét tới t_1 mới dừng. Với tài liệu lớn thì xác suất này khó tính toán và vô cùng nhỏ và tốn nhiều thời gian tính toán. Mô hình N-gram thì chỉ xét tới N term đứng trước term đang cần xét, tương đương với tính xác suất có điều kiện của t_i với điều kiện t_{i-N} đứng trước t_{i-N+1} và … t_{i-1} đứng trước t_i. Như vậy với các term gần cuối thì độ phức tạp tính toán vẫn tương đương với các term đứng đầu.

Trong bài báo cáo này, ta sẽ tiến hành thực hiện với $N = 1$ là mô hình Unigram, $N = 2$ là mô hình Bigram.

Mô hình Unigram là mô hình chỉ tính xác suất của term t_i trong tài liệu.

$$M_d = p(t_1, t_2, t_3, \dots, t_n) \approx p(t_1)p(t_2)p(t_3) \dots p(t_n)$$

Mô hình Bigram là mô hình chỉ tính xác suất có điều kiện của term t dựa vào term đứng trước nó hay xác suất có điều kiện term t_i mà theo thứ tự xuất hiện là t_{i-1} đứng trước t_i.

$$M_d = p(t_1, t_2, t_3, \dots, t_n) \approx p(t_1)p(t_2|t_1)p(t_3|t_2) \dots p(t_n|t_{n-1})$$

Ngoài ra, ta có thể chọn với bất kỳ giá trị N. Như trong mô hình Trigram là mô hình chỉ tính xác suất có điều kiện của term t dựa vào hai term đứng trước nó hay xác suất có điều kiện của term t_i mà theo thứ tự xuất hiện là t_{i-2} đứng trước t_{i-1} và t_{i-1} đứng trước t_i.

$$M_d = p(t_1, t_2, t_3, \dots, t_n) \approx p(t_1)p(t_2|t_1)p(t_3|t_2, t_1) \dots p(t_n|t_{n-1}, t_{n-2})$$

Để tính được trọng số của $p(t_1, t_2, t_3, \dots, t_n)$ theo mô hình N-gram thì một phương pháp rất hiệu quả trong thực tế và dễ cài đặt là Maximum Likelihood Estimation (MLE). MLE là một phương pháp thống kê tối ưu các tham số trong mô hình để tạo ra xác suất xuất hiện của dữ liệu đầu vào là lớn nhất có thể.

Đối với Language Model thì MLE sẽ tối ưu giá trị của $p(t_n|t_{n-1}, t_{n-2}, .., t_{n-N})$ để xác suất $p(t_1), p(t_2|t_1), p(t_3|t_2, t_1) \dots p(t_n|t_{n-1}, t_{n-2}, \dots, t_{n-N})$ lớn nhất.

Trong mô hình Unigram, để tính MLE của $p(t_n|t_{n-1}, t_{n-2}, \dots, t_{n-N})$ thì với mỗi tài liệu, chúng ta sẽ đếm số lần xuất hiện trong tài liệu hay tần số xuất hiện của t_i. Sau đó tính xác suất của t_i theo MLE bằng cách lấy tần số xuất hiện của t_i chia cho tổng số term trong tài liệu tương ứng.

$$p(t_i) = \frac{c(t_i)}{\sum_{i=1}^{n} c(t_i)}; t_i \in d$$

Trong mô hình Bigram, để tính MLE của $p(t_n|t_{n-1}, t_{n-2}, \dots, t_{n-N})$ thì với mỗi tài liệu, chúng ta sẽ đếm số lần xuất hiện của cặp term (t_{i-1}, t_i). Sau đó tính xác suất của (t_i, t_{i-1}) theo MLE bằng cách lấy số lần xuất hiện của (t_i, t_{i-1}) chia cho tổng số cặp term (t_i, t_{i-1}) trong tài liệu tương ứng.

$$p(t_i|t_{i-1}) = \frac{c(t_i|t_{i-1})}{\sum_{i=1}^{n} c(t_i|t_{i-1})}; t_i, t_{i-1} \in d$$

Ví dụ: ta tiếp hành hành xây dựng M_d cho 3 tài liệu trong tập Cranfield đã được lập chỉ mục ở muc 4. Trong ví dụ này ta chỉ thực hiện trên Unigram và đối tương tự đối với Bigram.

Term	**< ID(doc): freq >**		
<s>	0: 0.012	1: 0.008	2: 0.053
experiment	0: 0.025		

investig	0: 0.012	1: 0.008	
aerodynam	0: 0.012		
	…		
pressur			2: 0.053
gradient			2: 0.053
</s>	0: 0.012	1: 0.008	2: 0.053

Bảng 6: Tần số xuất hiện của từng term trong tập tài liệu – Unigram.

6. TÍNH $Rel(q|d)$

Để tính $Rel(q|d)$ thì ta quy chiếu các term trong query sang trọng số của term tương ứng trong M_d. Sau đó tích tất cả trọng số của term trong query sau khi quy chiếu ta được $Rel(q|d)$.

$$Rel(q|d) = p(q|M_d) = \prod_{1}^{m} p_{M_d}(t_i) ; t_i \in q$$

Ví dụ: Thực hiện truy vấn câu truy vấn q sau:

```
what design factors can be used
```

- Bước 1: Tiền xử lý câu truy vấn ta được các term sau:

term	$< s >$	design	factor	use	$</s>$

Bảng 7: Các term trong câu truy vấn q.

- Bước 2: Tính $p\left(t_i \middle| M_{d_j}\right); t_i \in q$

DocID	$\boldsymbol{p(q\|M_d)}$
d_1	$p(< s > \|M_{d_1}) \times p(design\|M_{d_1}) \times p(factor\|M_{d_1}) \times p(use\|M_{d_1}) \times p(</s > \|M_{d_1})$
d_2	$p(< s > \|M_{d_2}) \times p(design\|M_{d_2}) \times p(factor\|M_{d_2}) \times p(use\|M_{d_2}) \times p(</s > \|M_{d_2})$

d_3	$p(< s > \|M_{d_3}) \times p(design\|M_{d_3}) \times p(factor\|M_{d_3}) \times p(use\|M_{d_3}) \times p(</s > \|M_{d_3})$

Bảng 8: Tính $Rel(q|d)$.

Sau khi tính $Rel(q|d)$ với từng tài liệu, ta tiến hành xếp hạng theo thứ tự giảm dần. Ví dụ, ta xếp hạng được kết quả sau $\mathrm{d}_3, \mathrm{d}_1, \mathrm{d}_2$; với kết quả này ta thấy được tài liệu thứ 3 có độ liên quan gần nhất đối với câu truy vấn q.

7. SMOOTHING

Tuy nhiên, có trường hợp trong tài liệu truy vấn không chứa term nằm trong truy vấn dẫn đến xác suất của term đó trong tài liệu bằng không dẫn đến độ liên quan giữa tài liệu và truy vấn sẽ bằng không. Điều này rất tệ với hệ thống truy vấn vì trong thực tế đa số các tài liệu không thể nào chứa hết toàn bộ term trong truy vấn, từ đó nếu một tài liệu chứa gần hết term trong truy vấn và có trọng số các term trong truy vấn cao mà chỉ thiếu 1 term thì độ liên quan về không, dẫn đến kết quả trả về sẽ không có độ chính xác cao và độ phủ rất thấp vì rất ít tài liệu được trả về.

Để tránh việc gán xác suất của term trong query bằng không, ta dùng một kỹ thuật gọi là smoothing bằng cách chia sẻ một lượng xác suất rất nhỏ từ các term có xác suất khác không cho các term có trọng số bằng không để tránh $Rel(q|M_d)$ về không mà chỉ giảm đi độ liên quan đối với truy vấn.

Ngoài ra khi sử dụng N-gram càng lớn như Bigram và Trigram, thì (t_{i-1}, t_i) và (t_{i-2}, t_{i-1}, t_i) trong truy vấn ít có khả năng xuất hiện trong tài liệu, bởi vì tổ hợp các term có 2 từ và 3 từ rất nhiều, nên một tài liệu không thể nào chứa hết toàn bộ bigram và trigram của một truy vấn kể cả với tài liệu lớn trong khi một truy vấn thường có tổ hợp Bigram và Trigram rất đa dạng về mặt từ ngữ.

Có nhiều kỹ thuật smoothing, trong bài báo cáo này sẽ đề cập tới 2 kỹ thuật smoothing phổ biến trong Language Model là Laplace (add-k) Smoothing và Linear Interpolation Smoothing.

7.1. LAPLACE SMOOTHING

Laplace Smoothing là một kỹ thuật smoothing đơn giản nhất trong Language Model. Laplace Smoothing sẽ cộng vào tần số xuất hiện của các term α lần. Hay nói cách khác là tần số xuất hiện của term đã xuất hiện sẽ có tần số xuất hiện mới là $c(t_i) + \alpha$ lần, còn các term không xuất hiện trong tài liệu sẽ có số lần xuất hiện mới là α lần. Khi này trọng số của các term trong tài liệu sẽ được tính theo tần số xuất hiện mới.

Unigram	**Bigram**
$\boldsymbol{p(t_i) = \frac{count(t_i) + \alpha}{\sum_{i=1}^{n} count(t_i) + \alpha}}$; $\boldsymbol{t_i \in d}$	$p(t_i \mid t_{i-1}) = \frac{count(t_i \mid t_{i-1}) + \alpha}{\sum_{i=1}^{n} count(t_i \mid t_{i-1}) + \alpha}$, $t_i, t_{i-1} \in d$

Bảng 9: Laplace Smoothing với Unigram và Bigram.

Áp dụng Laplace Smoothing với $\alpha = 1$ với ví dụ trên, ta có mô hình ngôn ngữ với mỗi tài liệu như sau:

Term	$<$ **ID(doc): freq** $>$		
<s>	0: 1 + 1	1: 1 + 1	2: 1 + 1
experiment	0: 2 + 1		
investig	0: 1 + 1	1: 1 + 1	
aerodynam	0: 1 + 1		
	…		
pressur			2: 1 + 1
gradient			2: 1 + 1
</s>	0: 1 + 1	1: 1 + 1	2: 1 + 1

Bảng 10: Chỉ mục đối với Language Model – Unigram dùng Laplace Smoothing.

Term	**< ID(doc): freq >**		
<s>	0: 0.009	1: 0.008	2: 0.014
experiment	0: 0.015		
investig	0: 0.009	1: 0.008	
aerodynam	0: 0.009		
	…		
pressur			2: 0.014
gradient			2: 0.014
</s>	0: 0.009	1: 0.008	2: 0.014

Bảng 11: Tần số xuất hiện của từng term trong tập tài liệu – Unigram dùng Laplace Smoothing.

Sau khi dùng Laplace Smoothing cập nhật lại trọng số ta tiến hành các bước tương tự trên.

Laplace Smoothing làm thay đổi rất lớn tần số xuất hiện của term, dẫn đến xác suất gốc của term xuất hiện thay đổi rất lớn. Laplace Smoothing còn một tham số rất quan trọng là α, thay đổi α cũng sẽ thay đổi đến xác suất của các term. Vì thế chọn α rất quan trọng trong Laplace Smoothing, chọn α tốt sẽ giúp cho kết quả trả về có độ chính xác cao hơn. Để chọn được α tối ưu thì nên thử nghiệm truy vấn với một tập dữ liệu thử nghiệm và đánh giá hệ thống qua mỗi lần thay đổi α và chọn α dựa trên kết quả đánh giá tốt nhất.

7.2. LINEAR INTERPOLATION SMOOTHING

Ngoài cách cộng thêm vào tần số xuất hiện của các term không xuất hiện để tránh vấn đề xác suất bằng không, còn có một phương pháp khác để tránh vấn

đề xác suất bằng không. Ví dụ với mô hình Trigram, khi tính xác suất của một term t_i bất kỳ thì ta phải tính $p(t_i|t_{i-1}, t_{i-2})$, như đã nói ở trên với các mô hình N-gram càng lớn thì khả năng xảy ra vấn đề xác suất bằng không càng cao, vì thế khi tính $p(t_i|t_{i-1}, t_{i-2})$ mà tần số của $(t_i|t_{i-1}, t_{i-2})$ trong tài liệu bằng không thì có thể sử dụng xác suất của Bigram đại diện cho xác suất Trigram, tức là thay vì tính $p(t_i|t_{i-1}, t_{i-2})$ thì thay thế bằng xác suất của $p(t_i|t_{i-1})$, và nếu Bigram (t_i, t_{i-1}) có tần số xuất hiện cũng bằng không thì sẽ tiếp tục đi lùi về Unigram và tính $p((t_i)$ để đại diện cho xác suất của Trigram $p(t_i|t_{i-1}, t_{i-2})$.

Đối với phương pháp Interpolation, chúng ta luôn phối hợp sử dụng tất cả xác suất từ tất cả n-gram. Linear Interpolation Smoothing là một phương pháp đơn giản phối hợp tất cả xác suất của N-gram bằng cách cộng tất cả xác suất của N-gram mà mỗi xác suất sau khi nhân một số λ đại diện cho độ quan trọng của loại N-gram đó sao cho tổng tất cả λ đều bằng 1. Điều này còn giúp giải quyết vấn đề của Laplace Smoothing khi $p(q|M_d)$ của câu truy vấn q dối với các tài liệu bằng nhau.

$$p(t_i|t_{i-1}, t_{i-2}, \dots, t_{i-N}) = \lambda_1 p(t_i) + \lambda_2 p(t_i|t_{i-1}) + \cdots + \lambda_n p(t_i|t_{i-1}, t_{i-2}, \dots, t_{i-N})$$

$$\sum_{i=1}^{n} \lambda_i = 1$$

Đối với mô hình Unigram, vì điểm xuất phát là mô hình Unigram nên khi t_i không xuất hiện trong tài liệu, chúng ta có thể đếm tần số xuất hiện của t_i trong toàn bộ corpus, sau đó tính xác suất của t_i trong corpus bằng cách lấy tần số xuất hiện trong corpus chia cho tổng số term xuất hiện trong corpus.

Đối với mô hình Bigram, nếu (t_i, t_{i-1}) không xuất hiện trong tài liệu thì chỉ cần tính xác suất của t_i, $p(t_i)$ có thể tính trong ngữ cảnh là tài liệu tương ứng hoặc toàn bộ corpus hoặc kết hợp cả 2 ngữ cảnh, nếu $p(t_i)$ bằng không trong tài liệu thì có thể thay thế bằng $p(t_i)$ của corpus.

Unigram	**Bigram**

$\boldsymbol{p(t_i) = \lambda_1 p(t_{i_d}) + \lambda_2 p\left(t_{i_{corpus}}\right)}$ $\boldsymbol{\lambda_1 + \lambda_2 = 1}$	$p(t_i \mid t_{i-1}) = \lambda_1 p(t_{i_d} \mid t_{i-1_d}) + \lambda_2 p\left(t_{i_{corpus}}\right)$ $\lambda_1 + \lambda_2 = 1$

Bảng 12: Linear Interpolation Smoothing với Unigram và Bigram.

Nhìn vào cả 2 công thức Linear Interpolation của mô hình Unigram và Bigram, ta có thể thấy khi xác suất của vế đầu bằng không thì xác suất của $p(\mathrm{t_i})$, $p(\mathrm{t_i}|\mathrm{t_{i-1}})$ vẫn còn phụ thuộc vào vế sau nên sẽ tránh được vấn đề xác suất bằng không. Khi còn cả 2 vế thì hệ số λ sẽ giúp xác định xác suất của vế đầu hay vế sau quan trọng hơn. Công thức còn chỉ ra rằng trong thực tế đôi khi nhìn vào ngữ cảnh lớn sẽ ít hiệu quả nhìn vào ngữ cảnh nhỏ hơn, điều này sẽ giúp xây dựng hệ thống phản ứng tốt hơn với câu truy vấn có nhiễu.

Áp dụng Linear Interpolation Smoothing với $\lambda_1 = 0.4$; $\lambda_2 = 0.6$ với ví dụ trên, ta có mô hình ngôn ngữ với mỗi tài liệu như sau:

$0.4 \times$

Terms	< ID (doc) : $p(t_{i_d})$ >		
<s>	0 : 0.012	1 : 0.008	2 : 0.053
experiment	0 : 0.025		
investig	0 : 0.012	1 : 0.008	
aerodynam	0 : 0.012		
...			
pressur			2 : 0.053
gradient			2 : 0.053

$+0.6 \times$

Terms	$p(t_{i_{corpus}})$
<s>	0.014
experiment	0.009
investig	0.009
aerodynam	0.005
...	
pressur	0.005
gradient	0.005

Hình 12: Cách thực hiện Linear Interpolation Smoothing.

Sau khi dùng Linear Interpolation Smoothing cập nhật lại trọng số ta tiến hành các bước tương tự trên.

Sau khi thử nghiệm với dữ liệu nhỏ, ta có thể thấy trong Linear Interpolation thì tham số $\boldsymbol{\lambda}$ đóng vai trò quan trọng trong việc chúng ta lựa chọn đối với mô hình Unigram thì nên xem xét giá trị của 1 tài liệu hay

toàn văn bản quan trọng hơn, còn với mô hình Bigram thì đánh giá xem một term có xác suất độc lập quan trọng hơn hay xác suất của term phụ thuộc term trước đó quan trọng hơn hay nói cách khác là xem xét ngữ cảnh của term có giúp tăng độ chính xác khi truy vấn của hệ thống hay với mỗi tài liệu, xem mỗi term là một tồn tại độc lập sẽ giúp hệ thống truy vấn tốt hơn.

Vì thế tìm kiếm tham số $\boldsymbol{\lambda}$ tối ưu rất quan trọng, tuy nhiên chúng ta không thể xét cứng giá trị $\boldsymbol{\lambda}$ đối với tùy trường hợp mà dữ liệu chúng ta sử dụng, việc chọn ưu tiên ngữ cảnh của term trong tài liệu hoặc chỉ xét sự tồn tại độc lập của term sẽ có kết quả khác nhau. Nên để chọn giá trị $\boldsymbol{\lambda}$ tối ưu chúng ta có thể chạy thuật toán brute force để tìm giá trị $\boldsymbol{\lambda}$ tối ưu với bộ dữ liệu thử nghiệm. Nhưng việc chọn giá trị $\boldsymbol{\lambda}$ còn có rất nhiều thuật toán hiệu quả khác giúp chúng ta tìm kiếm giá trị $\boldsymbol{\lambda}$ chính xác và nhanh hơn.

So sánh 2 kỹ thuật smoothing trên lí thuyết thì kỹ thuật Linear Interpolation smoothing hiệu quả hơn hẳn kỹ thuật Laplace Smoothing. Laplace Smoothing đối với các term không có trong tài liệu thì sẽ gán cho số lần xuất hiện của term đó thêm α lần tuy nhiên không có góp phần trong việc đánh giá tốt sự cần thiết của term đó trong sự liên quan giữa tài liệu và truy vấn mà chỉ cố gắng tránh vấn đề xác suất bằng không. Trong khi đó Linear Interpolation xem xét sự quan trọng của term đó kể cả khi term đó không có trong tài liệu đang xét thì vẫn xét tới ngữ cảnh nhỏ hơn của term đó từ đó giúp việc truy vấn tài liệu thấy được mối quan hệ giữa term đó và tài liệu đồng thời còn tránh đi việc chia sẻ xác suất từ các term có sự quan trọng cao trong tài liệu. Tuy nhiên kỹ thuật Laplace sẽ có ưu điểm hơn Linear Interpolation trong thời gian xây dựng chỉ mục bởi vì kỹ thuật Laplace chỉ cộng thêm vô tần số xuất hiện một số α, trong khi đó Linear Interpolation bắt buộc phải đi tính trước xác suất của các N-gram nhỏ, với

bộ corpus lớn thì độ phức tạp khi tính xác suất của Linear Interpolation sẽ cao hơn Laplace Smoothing.

8. NHẬN XÉT

8.1. ƯU ĐIỂM

- Thứ tự của từ trong truy vấn ảnh hưởng đến kết quả trả về.
- Language Model có quan tâm đến ý nghĩa theo thứ tự của một từ. Vì là một mô hình ứng dụng xác suất, nếu trong truy vấn có từ bị nhiễu, thì Language Model vẫn có thể dựa vào các từ xung quanh để tìm kiếm kết quả truy vấn và trả về tài liệu liên quan gần với nhu cầu thông tin của người dùng.

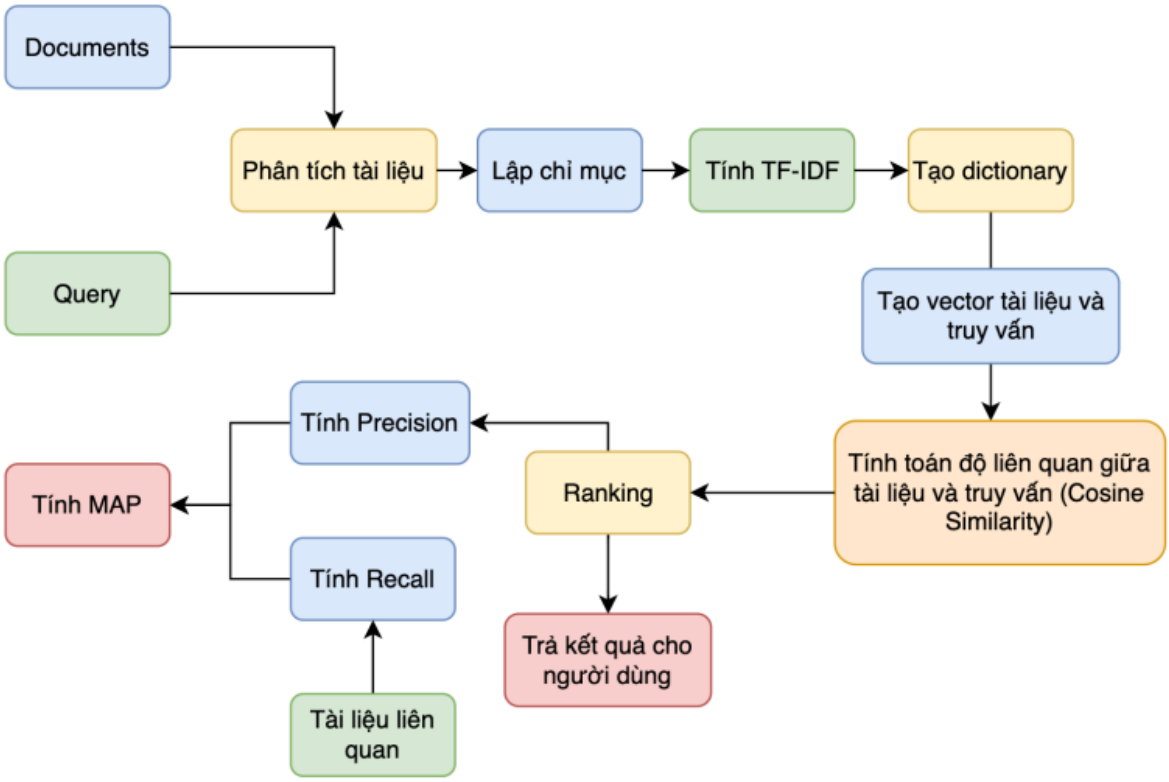
8.2. NHƯỢC ĐIỂM

- Mô hình N-gram sử dụng có N càng lớn (4-gram, 5-gram, …) thì số ngram phải xử lí càng nhiều và không gian lưu trữ càng lớn.
- Language Model dựa vào xác suất nên nếu một từ xuất hiện nhiều trong một tài liệu thì có khả năng xếp hạng của tài liệu đó những xếp hạng khá cao.
- Dù Language Model có quan tâm đến thứ tự của một từ nhưng không khai thác được khía cạnh nội dung của truy vấn.
- Với các mô hình N-gram ứng dụng kỹ thuật Smoothing Interpolation sẽ cần một không gian lưu trữ xác suất của các N-gram nhỏ hơn, điều đó cần không gian lưu trữ lớn.

IV. CÀI ĐẶT VÀ ĐÁNH GIÁ

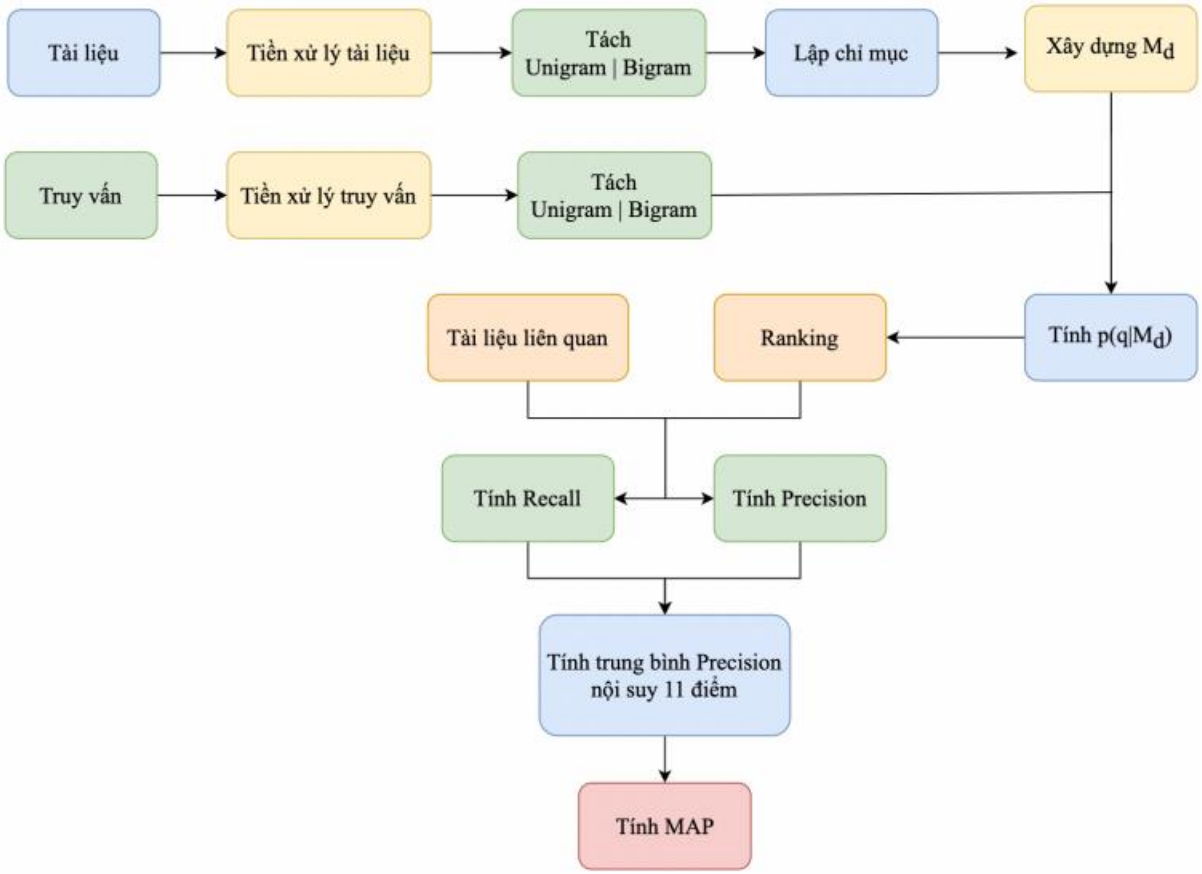
1. THIẾT KẾ VÀ CÀI ĐẶT

1.1. VECTOR SPACE MODEL



Hình 13: Các bước cài đặt trong Vector Space Model

1.2. LANGUAGE MODEL



Hình 14: Các bước cài đặt trong Language Model

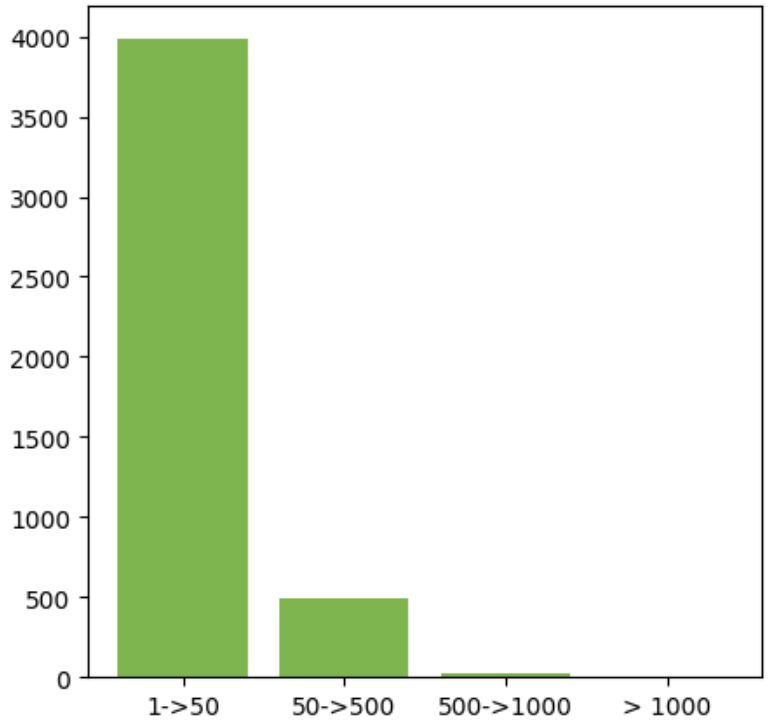
2. BỘ DỮ LIỆU CRANFIELD

Bộ dữ liệu Cranfield là một trong những bộ dữ liệu tiên phong và có vai trò quan trọng trong việc đánh giá hiệu quả của các hệ thống tìm kiếm thông tin. Được phát triển vào những năm 1960 và 1970 bởi Tổ chức Nghiên cứu Hàng không và Vũ trụ Mỹ (NASA) cùng với Đại học Cranfield ở Anh, bộ dữ liệu này đã trở thành một trong những bộ dữ liệu phổ biến nhất trong lĩnh vực Truy xuất thông tin (Information Retrieval).

Bộ dữ liệu Cranfield bao gồm các tài liệu khoa học và các câu hỏi liên quan đến các tài liệu này. Nó được sử dụng để đánh giá hiệu quả của các hệ thống truy xuất thông tin bằng cách so sánh kết quả truy xuất được với các câu trả lời chính xác của các câu hỏi. Điều này giúp các nhà nghiên cứu và chuyên

gia trong lĩnh vực truy xuất thông tin đánh giá và cải thiện hiệu suất của các hệ thống tìm kiếm thông tin.

Có 2 phiên bản của bộ dữ liệu Cranfield, trong đó Bộ dữ liệu Cranfield 2 bao gồm khoảng 1400 tài liệu và 225 câu hỏi truy vấn liên quan, được phát triển vào những năm 1960. Các tài liệu này bao gồm các báo cáo kỹ thuật, bài báo khoa học và các tài liệu khác liên quan đến lĩnh vực hàng không và vũ trụ. Mỗi tài liệu trong bộ dữ liệu này được phân loại vào một trong 6 chủ đề khác nhau, nhằm phục vụ cho việc đánh giá hiệu quả của các hệ thống truy xuất thông tin.



Hình 15: Biểu đồ thống kê số lượng term dựa trên tần suất.

Sau khi thực hiện các phương pháp tokenization, stemming, loại bỏ stopword ... để lựa chọn các term, chúng ta có được kết quả cụ thể là ở cột đầu tiên 1->50 chúng ta được 3989, cột thứ hai 50->500 được 498, cột thứ 3 500-1000 được 29 và cột chiếm số lượng ít nhất là 6 với giá trị >1000. Kết quả phân tích trên cho thấy rằng phân bố tần suất xuất hiện của các thuật ngữ trong

tập dữ liệu là không đồng đều. Các thuật ngữ có tần suất xuất hiện thấp từ 1 đến 50 chiếm tỷ lệ rất lớn, khoảng 88%, trong khi các thuật ngữ có tần suất xuất hiện từ 50 đến 500 chỉ chiếm một phần nhỏ hơn, khoảng 11%. Điều này cho thấy rằng, trong tập dữ liệu này, các thuật ngữ phổ biến (tần suất xuất hiện cao) chiếm phần lớn và sự đa dạng của thuật ngữ trong tập dữ liệu là không cao.

Ngoài ra, tần suất xuất hiện của các thuật ngữ từ 500 đến 1000 và trên 1000 rất hiếm, cho thấy rằng có một số thuật ngữ rất đặc biệt và ít xuất hiện trong tập dữ liệu này. Kết quả này có thể cung cấp thông tin hữu ích cho việc xây dựng các hệ thống truy xuất thông tin, giúp tối ưu hóa các kết quả truy vấn và cải thiện hiệu quả truy xuất thông tin.

3. PHƯƠNG PHÁP ĐÁNH GIÁ MÔ HÌNH

3.1. MEAN AVERAGE PRECISION (mAP)

mAP (Mean Average Precision) là một độ đo đánh giá hiệu suất cho các hệ thống truy xuất thông tin. Độ đo này được sử dụng rộng rãi trong các nghiên cứu và thử nghiệm hệ thống truy xuất thông tin, đặc biệt là trong việc đánh giá chất lượng các kết quả truy xuất thông tin.

Một cách đơn giản, mAP là trung bình của các giá trị precision ở các ngưỡng khác nhau. Precision là tỉ lệ giữa số lượng kết quả truy xuất đúng và tổng số lượng kết quả truy xuất. Giá trị precision tại một ngưỡng được tính bằng cách chia số lượng kết quả truy xuất đúng trước ngưỡng đó cho tổng số lượng kết quả truy xuất trước ngưỡng đó. Hơn nữa, mAP là một độ đo phù hợp với hầu hết các hệ thống truy xuất thông tin và không chỉ đáp ứng nhu cầu của người dùng trong việc cần có kết quả phù hợp ở các vị trí đầu tiên mà còn đảm bảo không bỏ lỡ các kết quả phù hợp khác.

Chúng ta có thể tính mAP theo các bước như sau:

- *Bước 1: Tính giá trị Precision*

Precision trong độ đo mAP là độ đo dùng để đánh giá chất lượng của các kết quả truy xuất. Một giá trị precision cao cho thấy rằng hệ thống truy xuất thông tin đưa ra nhiều kết quả truy xuất chính xác. Tuy nhiên, precision cũng cần được đánh giá trong bối cảnh của bài toán truy xuất thông tin cụ thể

$$precision = \frac{Tổng\ số\ tài\ liệu\ được\ truy\ xuất\ có\ liên\ quan}{Tổng\ số\ tài\ liệu\ được\ truy\ xuất}$$

- *Bước 2: Tính AP*

AP (Average Precision) là một thành phần quan trọng của độ đo mAP (Mean Average Precision) trong các bài toán truy xuất thông tin. AP được tính bằng cách tính toán precision tại các giá trị recall khác nhau và sau đó tính trung bình của các giá trị precision này.

$$AP = \frac{\sum_{k=0}^{n-1} precision(k)}{n}$$

Với n là số lượng precision.

3.2. mAP NỘI SUY

Mean Average Precision (MAP) là một độ đo để đánh giá hiệu quả của các thuật toán tìm kiếm và xếp hạng kết quả truy xuất. MAP được sử dụng phổ biến trong các bài toán tìm kiếm thông tin, như tìm kiếm trên web, tìm kiếm hình ảnh và video, xếp hạng trang web, v.v…MAP đo lường khả năng của một thuật toán tìm kiếm trả về các kết quả đúng và đánh giá tính tổng quan của thuật toán đó.

Để tính MAP nội suy, ta sẽ sử dụng 11 điểm nội suy, mỗi điểm là trung bình của các giá trị Precision tại 11 điểm Recall tương ứng (0, 0.1, 0.2, 0.3, ..., 1).

- *Bước 1: Tính giá trị precision*

$$precision = \frac{Tổng\ số\ tài\ liệu\ được\ truy\ xuất\ có\ liên\ quan}{Tổng\ số\ tài\ liệu\ được\ truy\ xuất}$$

- *Bước 2: Tính giá trị Recall*

$$recall = \frac{Tổng\ số\ tài\ liệu\ được\ truy\ xuất\ có\ liên\ quan}{Tổng\ số\ tài\ liệu\ liên\ quan}$$

- *Bước 3: Tính interpolate precision*

$$P'(r) = max_{r':r'>r} p(r')$$

- *Bước 4: Tính giá trị AP*

$$AP = \frac{\sum_{k=0}^{11} precision(k)}{11}$$

- *Bước 5: Tính mAP nội suy*

$$mAP = \frac{\sum_{k=0}^{n-1} AP(k)}{n}$$

Với giá trị MAP cao, ta có thể kết luận rằng hệ thống đang thể hiện khả năng tìm kiếm và xếp hạng kết quả truy xuất tốt hơn. Điều này có thể đưa ra được các quyết định cần thiết trong việc cải thiện hoặc tối ưu hệ thống truy xuất thông tin và mô hình xếp hạng, để đạt được hiệu quả tốt hơn trong công việc truy xuất thông tin và đáp ứng được nhu cầu của người dùng.

3.3. THỜI GIAN TRUY XUẤT

Đánh giá hệ thống truy vấn bằng thời gian truy xuất là một phương pháp đánh giá quan trọng trong truy xuất thông tin. Thời gian truy xuất đo lường thời gian mà hệ thống cần để trả về kết quả truy xuất cho người dùng sau khi nhận được yêu cầu truy xuất.

Đánh giá hệ thống truy vấn bằng thời gian truy xuất có thể giúp đánh giá hiệu quả và tốc độ của hệ thống truy vấn. Nếu thời gian truy xuất của

hệ thống nhanh, điều đó có thể cho thấy hệ thống đang hoạt động hiệu quả và có thể đáp ứng được nhu cầu của người dùng. Tuy nhiên, nếu thời gian truy xuất quá chậm, điều đó có thể gây khó khăn cho người dùng và gây mất mát cho doanh nghiệp.

Ngoài ra, đánh giá hệ thống truy vấn bằng thời gian truy xuất còn cho phép tối ưu hóa hệ thống truy vấn bằng cách tối đa hóa tốc độ truy xuất. Ví dụ, các kĩ thuật tối ưu hóa cơ sở dữ liệu, tối ưu hóa các thuật toán tìm kiếm và sử dụng bộ nhớ đệm có thể được sử dụng để cải thiện thời gian truy xuất của hệ thống. Tuy nhiên, khi sử dụng thời gian truy xuất để tiến hành đánh giá hệ thống, cần phải kết hợp nhiều yếu tố khác như mAP, recalll v… để đưa ra đánh giá chính xác và toàn diện về hiệu quả của hệ thống truy vấn.

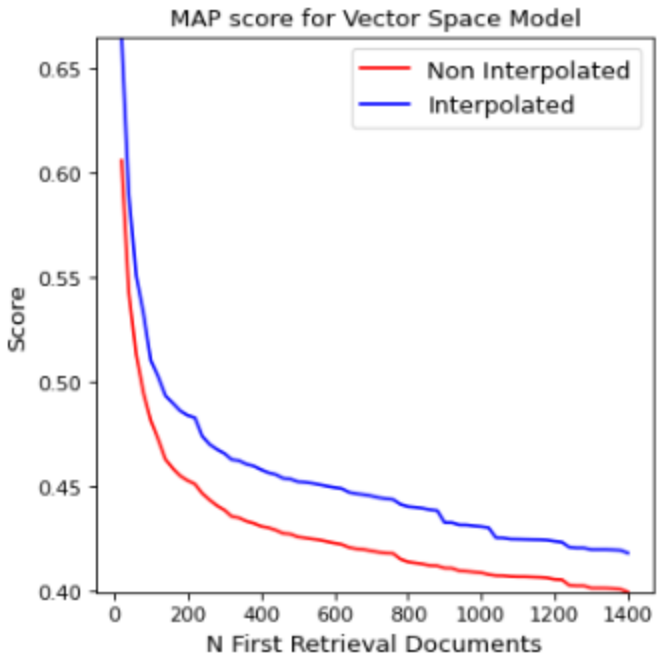
4. KẾT QUẢ THỰC NGHIỆM CỦA MÔ HÌNH

4.1. VECTOR SPACE MODEL

4.1.1. KHÔNG SỬ DỤNG SMART

Measure	Score
Recall	0.953
Precision	0.0099
mAP	0.415
mAP interpolated	0.442
Time process	21.04

Bảng 13: Kết quả đánh giá hệ thống



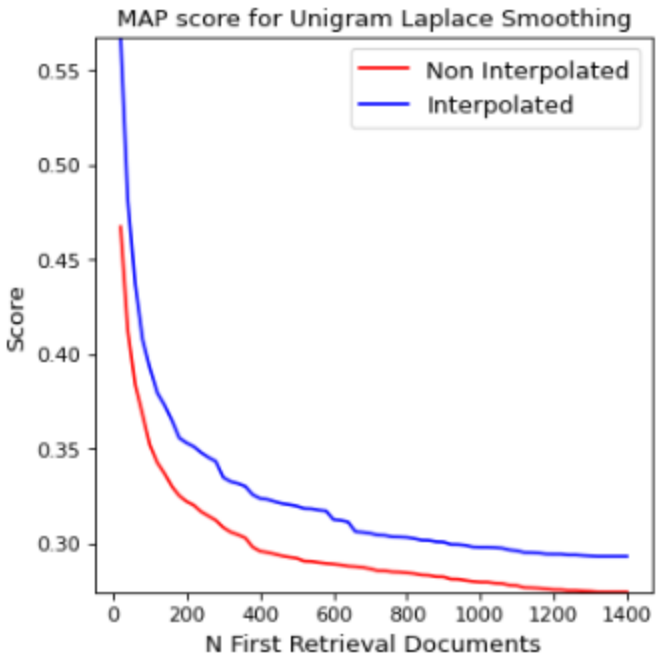
Hình 16: mAP khi không sử dụng SMART

Mô hình Vector Space không sử dụng SMART trả về được hầu hết các tài liệu liên quan, với Recall đạt 0,953. Tuy nhiên độ chính xác của các tài liệu trả về chưa cao, với số lượng tài liệu trả về lớn hơn nhiều so với số tài liệu liên quan, dẫn đến việc mô hình có Precision khá thấp chỉ 0,00995.

4.1.2. SỬ DỤNG SMART

Measure	Score
Recall	0.95399
Precision	0.00995
mAP	0.38975
mAP interpolated	0.41730
Time process	24.8621

Bảng 14: Kết quả đánh giá hệ thống khi sử dụng SMART



Hình 17: mAP khi sử dụng SMART

Mô hình Vector Space sử dụng SMART có độ đo Precision và Recall gần như nhau. Điều này chứng tỏ việc sử dụng SMART không giúp giảm số lượng các tài liệu không liên quan được trả về. Tuy việc sử dụng SMART có tác động lớn đến khả xếp hạng các tài liệu trả về. Cụ thể mAP đạt gấp đôi so với khi không sử dụng. Độ liên của các tài liệu được đánh giá tốt hơn, nhờ đó khả xếp hạng tài liệu được cải thiện.

4.1.3. KẾT LUẬN

Mô hình Vector Space là một trong những mô hình truy xuất thông tin phổ biến nhất được sử dụng trong lĩnh vực khoa học máy tính và thông tin học. Trong mô hình này, mỗi tài liệu và truy vấn được biểu diễn dưới dạng vector, trong đó mỗi chiều của vector tương ứng với một thuật ngữ trong tài liệu hoặc truy vấn. Phương pháp tính toán trọng số cho các thuật ngữ trong vector có vai trò quan trọng trong việc xác định độ tương đồng giữa tài liệu và truy vấn.

Việc lựa chọn phương pháp tính trọng số cho các thuật ngữ trong mô hình Vector Space đóng vai trò vô cùng quan trọng trong quá trình truy xuất thông tin. Điều này đã được chứng minh qua hai thí nghiệm khác nhau. Thí nghiệm đầu tiên đã chứng minh rằng mô hình sử dụng phương pháp SMART có khả năng xếp hạng các tài liệu trả về tốt hơn so với mô hình không sử dụng SMART. Phương pháp SMART sử dụng một hàm tính trọng số phức tạp, bao gồm việc tính toán tần số xuất hiện của thuật ngữ trong tài liệu, tần số của thuật ngữ trong toàn bộ tập tài liệu, và độ quan trọng của thuật ngữ đó trong tài liệu. Thí nghiệm thứ hai đã chứng minh rằng việc thay đổi phương pháp tính trọng số có thể góp phần quan trọng đến hiệu suất của mô hình Vector Space.

Với kết quả của hai thí nghiệm này, chúng ta có thể thấy rằng việc lựa chọn phương pháp tính trọng số thích hợp là rất quan trọng để đạt được hiệu suất truy xuất thông tin tốt nhất. Do đó, khi lựa chọn phương pháp tính trọng số, chúng ta cần xem xét kỹ lưỡng các yếu tố như loại tài liệu, độ dài tài liệu, và độ phổ biến của thuật ngữ để đạt được kết quả tốt nhất trong việc truy xuất thông tin.

4.2. LANGUAGE MODEL

Việc thực nghiệm mô hình Language Model với các kỹ thuật như sau: Language Model (Unigram và Bigram) với kỹ thuật Laplace Smoothing, Language Model (Unigram và Bigram) với kỹ thuật Linear Interpolation Smoothing. Đánh giá kết quả với 500 tài liệu trả về đầu tiên.

Đối với thực nghiệm Language Model với kỹ thuật Laplace Smoothing, thực nghiệm tìm kiếm giá trị của tham số α tối ưu nhất cho mỗi mô hình. Tìm kiếm tham số α tối ưu bằng cách phương pháp Brute Force với giá trị α trong khoảng [1,10] và bước nhảy là 1, và với mỗi giá

trị α, đánh giá thực nghiệm bằng 2 độ đô là mAP và mAP Interpolation. Lấy đánh giá có kết quả mAP và mAP Interpolation cao nhất để làm tham số α tốt ưu của mô hình.

Đối với thực nghiệm Language Model với kỹ thuật Linear Interpolation Smoothing, thực nghiệm tìm kiếm 2 giá trị λ tối ưu nhất cho mỗi mô hình. Tìm kiếm tham số λ tối ưu bằng phương pháp Brute Force cho giá trị λ_1 trong khoảng [0.05,0.95] với bước nhảy là 0.05 và $\lambda_2 = 1 - \lambda_1$, đánh giá thực nghiệm bằng 2 độ đô là mAP và mAP Interpolation. Lấy đánh giá có kết quả mAP và mAP Interpolation cao nhất để làm giá trị cho tham số λ tối ưu của mỗi mô hình.

4.2.1. UNIGRAM

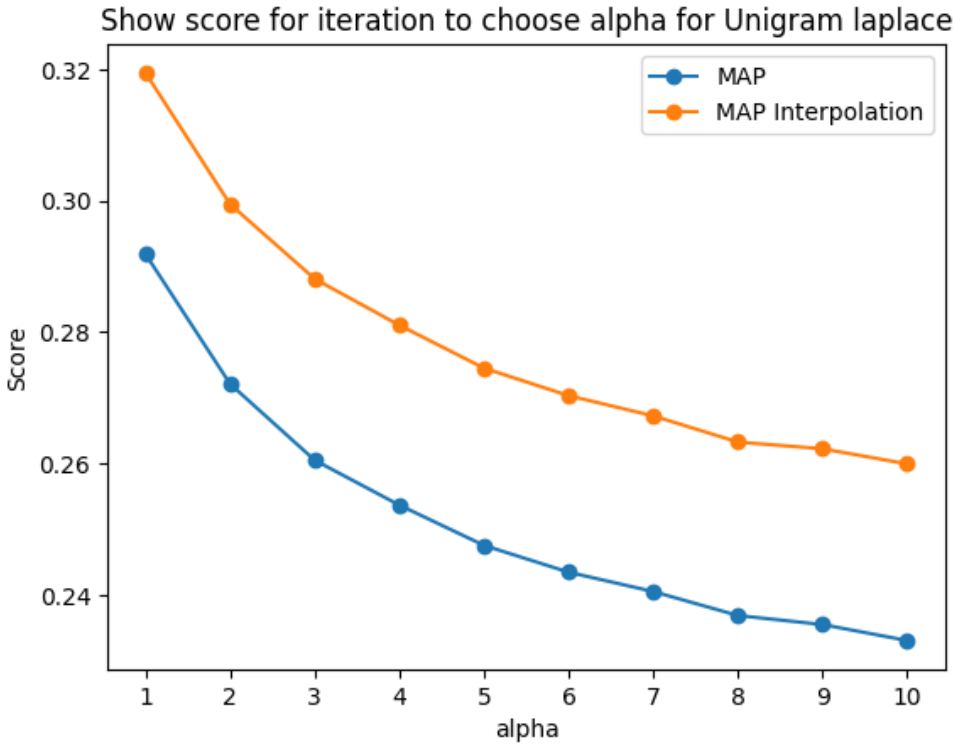
- Sử dụng Laplace Smoothing:

  Dưới đây là bảng kết quả chạy thực nghiệm mô hình Unigram với kỹ thuật Laplace Smoothing. Trong kỹ thuật Laplace Smoothing có một tham số quan trọng ảnh hưởng đến kết quả đánh giá là α. Chạy thực nghiệm với α bằng phương pháp Brute Force trong khoảng [1,10] và bước nhảy là 1 ta được kết quả như sau:

α	mAP	mAP Interpolation
1	0.2919	0.3195
2	0.2721	0.2995
3	0.2605	0.2881
4	0.2537	0.2811
5	0.2476	0.2745
6	0.2435	0.2703
7	0.2405	0.2673

8	0.2369	0.2633
9	0.2355	0.2627
10	0.2331	0.2600

Bảng 15: mAP và mAP nội suy với $\alpha = [1; 10]$ – Unigram.



Hình 18: mAP và mAP nội suy với $\alpha = [1; 10]$ – Unigram.

Nhìn vào bảng trên, ta có thể kết luận với corpus đang thử nghiệm thì khi α càng tăng thì hệ thống truy xuất càng kém. Vậy $\alpha = 1$ cho đánh giá kết quả truy xuất tốt nhất. Dưới đây là bảng kết quả thực nghiệm của mô hình Unigram Laplace với $\alpha = 1$.

Recall	Precision	mAP	mAP Interpolation
0.8990	0.0144	0.2919	0.3195

Bảng 16: Kết quả đánh giá Language Model – Unigram sử dụng Laplace Smoothing.

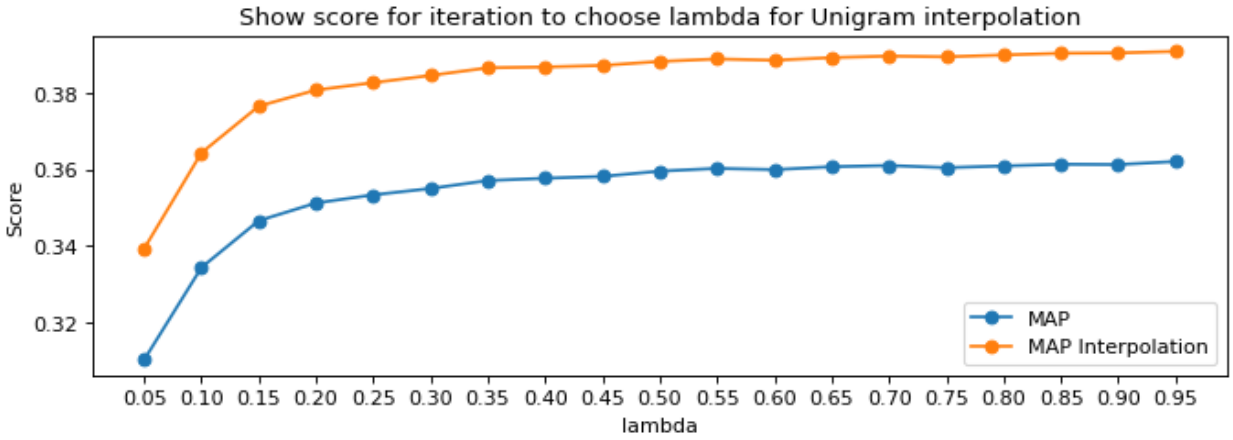
- Sử dụng Linear Interpolation Smoothing:

Dưới đây là bảng kết quả chạy thực nghiệm mô hình Unigram với kỹ thuật Linear Interpolation Smoothing. Trong kỹ thuật Linear interpolation của Unigram thì tham số λ có vai trò quan trọng trong việc xác suất của term không phụ thuộc vào term nào xuất hiện trong tài liệu nhỏ và xác suất của term không phụ thuộc vào term nào xuất hiện trong cả một corpus thì tham số nào quan trọng hơn trong hệ thống truy vấn sử dụng mô hình Unigram. Chạy thực nghiệm với λ_1 bằng phương pháp Brute Force trong khoảng [0.05,0.95] và bước nhảy là 0.05, $\lambda_2 = 1 - \lambda_1$.

$\boldsymbol{\lambda_1}$	$\boldsymbol{\lambda_2}$	***mAP***	***mAP Interpolation***
0.05	*0.95*	0.3102	0.3393
0.10	*0.90*	0.3343	0.3644
0.15	*0.85*	0.3466	0.3766
0.20	*0.80*	0.3512	0.3808
0.25	*0.75*	0.3533	0.3827
0.30	*0.70*	0.3550	0.3846
0.35	*0.65*	0.3571	0.3866
0.40	*0.60*	0.3577	0.3868
0.45	*0.55*	0.3582	0.3872
0.50	*0.50*	0.3596	0.3883
0.55	*0.45*	0.3603	0.3889
0.60	*0.40*	0.3600	0.3886
0.65	*0.35*	0.3607	0.3893
0.70	*0.30*	0.3610	0.3897
0.75	*0.25*	0.3605	0.3895

0.80	*0.20*	0.3609	0.3900
0.85	*0.15*	0.3614	0.3904
0.90	*0.10*	0.3613	0.3905
0.95	*0.05*	0.3621	0.3909

Bảng 17: mAP và mAP nội suy với $\lambda_1, \lambda_2 = [0.05; 0.95]$ – Unigram.



Hình 19: mAP và mAP nội suy với $\lambda_1, \lambda_2 = [0.05; 0.95]$ – Unigram.

Nhìn vào bảng trên, ta có thể thấy kỹ thuật Linear Interpolation Smoothing cho kết quả đánh giá tốt hơn rất nhiều so với kỹ thuật Laplace dù kể cả với kết quả đánh giá thấp nhất của Linear Interpolation. Mặc khác, ta còn thấy được với cặp tham số $\lambda_1 = 0.95$ và $\lambda_2 = 0.05$ cho kết quả truy xuất tốt nhất. Từ 2 kết quả đánh giá tốt nhất, ta có thể thấy một điều đối với mô hình Unigram sử dụng kỹ thuật Linear Interpolation smoothing thì xác suất của một term nằm trong corpus quan trọng hơn một xác suất của một term trong tài liệu.

Dưới đây là bảng kết quả thí nghiệm đầy đủ của 2 cặp tham số $\lambda_1 = 0.95$ và $\lambda_2 = 0.05$.

Recall	Precision	mAP	mAP Interpolation
0.8978	0.0144	0.3577	0.3868

Bảng 18: Kết quả đánh giá Language Model – Unigram sử dụng

Linear Interpolation Smoothing.

Nhìn vào bảng kết quả trên, ta có thể đối với mô hình Unigram Linear Interpolation đều vượt trội hơn hẳn so với mô hình Unigram Laplace.

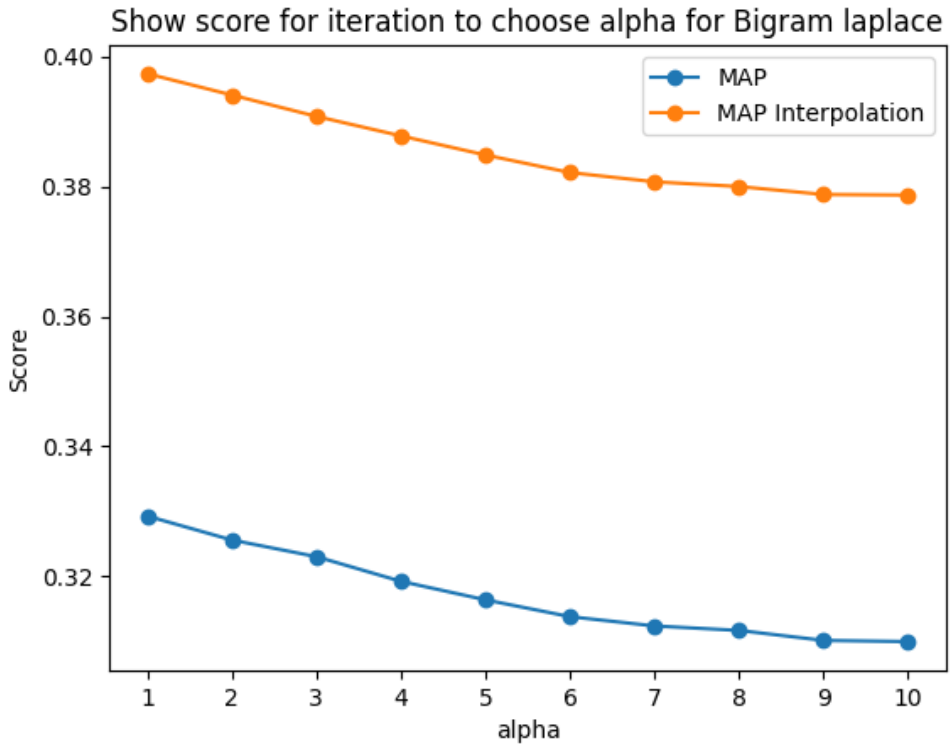
4.2.2. BIGRAM

- Sử dụng Laplace Smoothing:

  Dưới đây là bảng kết quả chạy thực nghiệm mô hình Unigram với kỹ thuật Laplace Smoothing. Trong kỹ thuật Laplace Smoothing có một tham số quan trọng ảnh hưởng đến kết quả đánh giá là α. Chạy thực nghiệm với α bằng phương pháp Brute Force trong khoảng [1,10] và bước nhảy là 1 ta được kết quả như sau:

α	mAP	$mAP\ Interpolation$
1	0.3293	0.3974
2	0.3256	0.3941
3	0.3230	0.3908
4	0.3192	0.3878
5	0.3164	0.3848
6	0.3138	0.3821
7	0.3123	0.3807
8	0.3116	0.3800
9	0.3101	0.3788
10	0.3099	0.3787

Bảng 19: mAP và mAP nội suy với $\alpha = [1; 10]$ *– Bigram.*



Hình 20: mAP và mAP nội suy với $\alpha = [1; 10]$ – Bigram.

Nhìn vào bảng kết quả trên, ta cũng có thể thấy kết quả đánh giá tốt hơn rất nhiều so với mô hình Unigram dùng chung một kỹ thuật và vẫn giống với kết quả đánh giá Unigram là giá trị của α càng lớn thì kết quả đánh giá càng kém ở cả 2 mô hình. Vậy có thể nói $\alpha = 1$ là giá trị α tối ưu nhất cho cả 2 mô hình Unigram và Bigram sử dụng kỹ thuật Laplace Smoothing.

Dưới đây là bảng kết quả thực nghiệm của mô hình Bigram Laplace với $\alpha = 1$.

Recall	Precision	mAP	mAP Interpolation
0.6175	0.0099	0.3293	0.3974

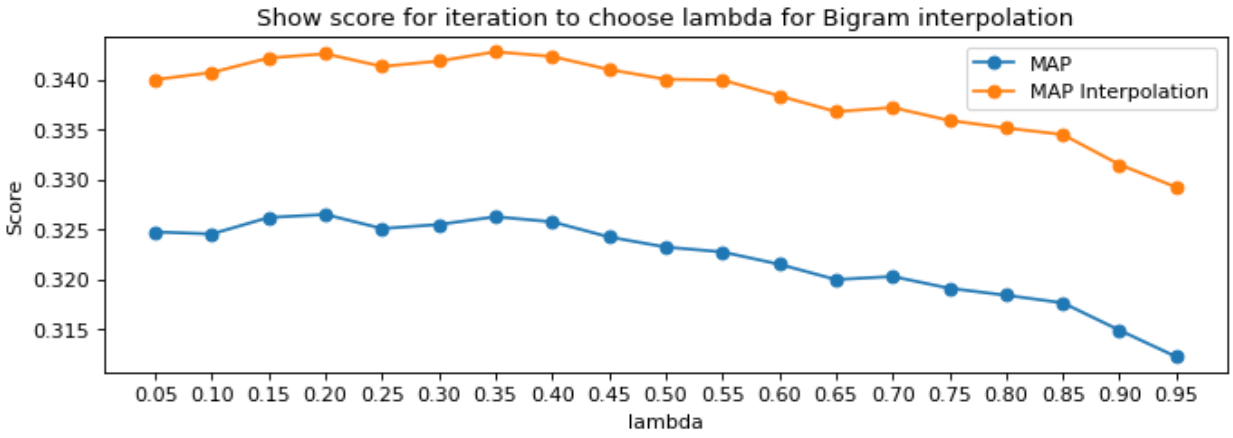
Bảng 20: Kết quả đánh giá Language Model – Bigram sử dụng Laplace Smoothing.

Từ bảng trên, ta có thể thấy với mặc dù kết quả đánh giá của 2 độ đo mAP và mAP Interpolation của Bigram sử dụng kỹ thuật Laplace có kết quả cao hơn rất nhiều so với mô hình Unigram sử dụng chung kỹ thuật, tuy nhiên có một điều mà mô hình Unigram có lợi thế hơn là là recall của hệ thống cao hơn rất nhiều so với mô hình Bigram.

- Sử dụng Linear Interpolation Smoothing:

λ_1	λ_2	mAP	$mAP\ Interpolation$
0.05	*0.95*	0.3247	0.3400
0.10	*0.90*	0.3245	0.3407
0.15	*0.85*	0.3262	0.3422
0.20	*0.80*	0.3265	0.3426
0.25	*0.75*	0.3250	0.3413
0.30	*0.70*	0.3255	0.3418
0.35	*0.65*	0.3262	0.3428
0.40	*0.60*	0.3257	0.3423
0.45	*0.55*	0.3242	0.3410
0.50	*0.50*	0.3232	0.3400
0.55	*0.45*	0.3227	0.3399
0.60	*0.40*	0.3215	0.3384
0.65	*0.35*	0.3199	0.3368
0.70	*0.30*	0.3202	0.3372
0.75	*0.25*	0.3191	0.3359
0.80	*0.20*	0.3184	0.3351
0.85	*0.15*	0.3176	0.3345
0.90	*0.10*	0.3149	0.3315
0.95	*0.05*	0.3122	0.3292

Bảng 21: mAP và mAP nội suy với $\lambda_1, \lambda_2 = [0.05; 0.95]$ *– Bigram.*



Hình 21: mAP và mAP nội suy với $\lambda_1, \lambda_2 = [0.05; 0.95]$ – Bigram.

Dưới đây là bảng kết quả thí nghiệm đầy đủ của 2 cặp tham số λ tối ưu là $\lambda_1 = 0.2$ và $\lambda_2 = 0.8$.

Recall	Precision	mAP	mAP Interpolation
0.8555	0.0136	0.3215	0.3383

Bảng 22: Kết quả đánh giá Language Model – Bigram sử dụng Linear Interpolation Smoothing.

Nhìn vào bảng trên, ta có thể thấy đối với mô hình Bigram sử dụng kỹ thuật Linear Interpolation so với mô hình Unigram dùng chung kỹ thuật thì kết quả đánh giá có phần kém hơn nhiều nhưng so mô hình Unigram sử dụng kỹ thuật Laplace thì kết quả đánh giá vẫn tốt hơn rất nhiều. Nhưng so với mô hình Bigram sử dụng kỹ thuật Laplace thì kết quả đánh giá vẫn kém hơn tuy nhiên không quá chênh lệch như giữa mô hình Unigram Laplace và Unigram Linear Interpolation.

Đối với giá trị λ tối ưu thì với mô hình Bigram sử dụng kỹ thuật Linear Interpolation thì các cặp giá trị λ đều cho kết quả đánh giá không chênh lệch với nhau, nên với mô hình này thì giá trị λ không chỉ ra được xác suất của một term độc lập trong tài liệu quan trọng hay xác suất của một term phụ thuộc vào một term trước đó quan trọng hơn.

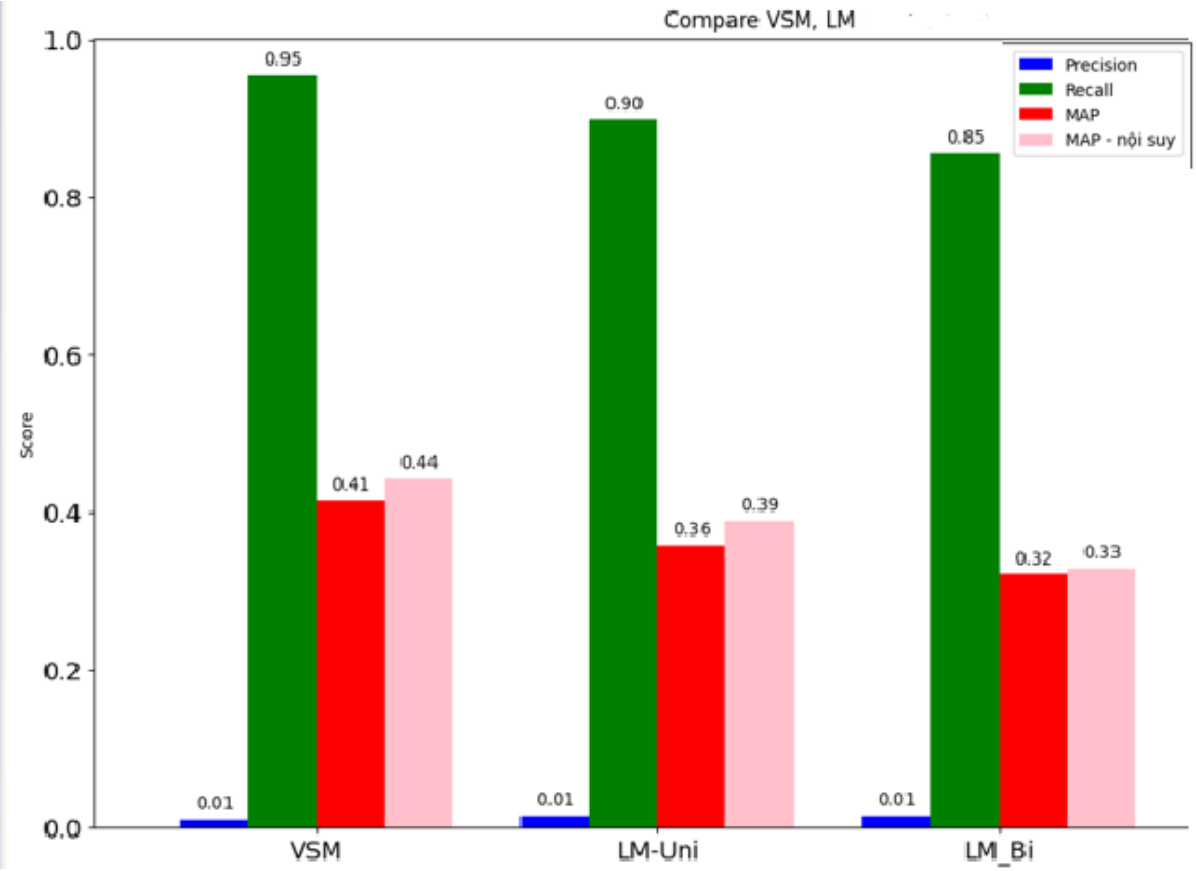
4.2.3. KẾT LUẬN

Sau khi phân tích kết quả đánh giá của 4 thí nghiệm trên mô hình Language Model, chúng ta có thể nhận thấy rằng kỹ thuật Linear Interpolation Smoothing được sử dụng trong mô hình Unigram đã mang lại kết quả đánh giá tốt nhất trong số các phương pháp được sử dụng. Mô hình Unigram với kỹ thuật này cũng cho thấy hiệu suất tốt trong một hệ thống truy vấn. Trong khi đó, kỹ thuật Laplace Smoothing trong mô hình Unigram lại cho kết quả đánh giá thấp nhất. Điều này có thể dễ hiểu vì Laplace Smoothing là một kỹ thuật đơn giản và đã lỗi thời trong thời điểm hiện tại.

Với mô hình Bigram, hai kỹ thuật được sử dụng chỉ cho thấy chênh lệch nhỏ về hiệu suất giữa chúng. Kết quả đánh giá của mô hình Bigram vẫn thấp hơn so với mô hình Unigram với kỹ thuật Linear Interpolation Smoothing, tuy nhiên hiệu suất của nó đã được cải thiện đáng kể so với mô hình Unigram thông thường.

Kết quả các thực nghiệm trên mô hình Language Model chỉ ra rằng việc chọn phương pháp Smoothing phù hợp là rất quan trọng để tối ưu hiệu suất của mô hình. Tuy nhiên, việc lựa chọn phương pháp Smoothing thích hợp không đơn giản và phụ thuộc vào nhiều yếu tố như loại dữ liệu, độ dài văn bản, số lượng thuật ngữ và nhu cầu truy xuất thông tin của người dùng. Vì vậy, cần tiến hành nhiều thí nghiệm trên các tập dữ liệu khác nhau để lựa chọn phương pháp Smoothing phù hợp nhất cho từng trường hợp.

5. SO SÁNH KẾT QUẢ



Hình 22: Biểu đồ kết quả thực nghiệm của VSM và Language model

Sau khi thực hiện hai mô hình trên bộ ngữ liệu Cranfield, ta có thể thấy rằng mô hình Vector Space có kết quả về mAP và mAP nội suy tốt hơn so với Language Model. Tuy nhiên, để đạt được kết quả tốt hơn, ta phải đánh đổi thời gian truy xuất giữa hai mô hình.

Mô hình Vector Space mất nhiều thời gian hơn để truy xuất vì cả document và query đều được biểu diễn dưới dạng vector, trong khi đó Language Model sử dụng công nghệ ngôn ngữ nên sẽ nhanh hơn. Tuy nhiên, Language Model lại có điểm yếu là việc xử lý các từ trong câu có thể gặp khó khăn khi có nhiều từ đồng nghĩa hoặc từ viết tắt.

Do đó, để lựa chọn mô hình phù hợp, ta cần cân nhắc giữa hiệu suất và thời gian truy xuất. Nếu yêu cầu tốc độ truy xuất cao hơn, ta có thể sử dụng

Language Model, nhưng nếu yêu cầu độ chính xác cao hơn, ta nên sử dụng mô hình Vector Space. Không chỉ như vậy, chúng ta cần phải xem xét đặc trưng của bộ dữ liệu, tùy theo nhu cầu và dữ liệu để lựa chọn được mô hình phù hợp.

6. SO SÁNH VỚI THƯ VIỆN WHOOSH

6.1. GIỚI THIỆU VỀ THƯ VIỆN WHOOSH

Whoosh được biết đến là một thư viện mã nguồn mở phổ biến trong lĩnh vực truy xuất thông tin trên Python. Thư viện này cung cấp cho người dùng các công cụ để xây dựng và quản lý một hệ thống tìm kiếm đơn giản nhưng hiệu quả. Với Whoosh, người dùng có thể dễ dàng tạo ra các hệ thống tìm kiếm chuyên nghiệp cho các ứng dụng của mình. Thư viện này được đánh giá cao về tính linh hoạt và khả năng tùy chỉnh, giúp người dùng dễ dàng thích nghi với nhiều loại dữ liệu và yêu cầu truy xuất khác nhau.

Whoosh là một thư viện hỗ trợ nhiều tính năng trong lĩnh vực truy xuất thông tin. Nó cung cấp cho người dùng các công cụ để tạo chỉ mục (index), tìm kiếm full-text, tìm kiếm theo truy vấn và hỗ trợ các tính năng khác như phân tích từ ngữ (tokenization), phân tích ngữ nghĩa (stemming), và thực thi truy vấn phức tạp.

Với tính năng tạo chỉ mục, người dùng có thể lưu trữ dữ liệu của mình dưới dạng chỉ mục, giúp tăng tốc độ truy vấn. Tính năng tìm kiếm full-text cho phép người dùng tìm kiếm thông tin trong toàn bộ văn bản, bao gồm cả nội dung và tiêu đề. Ngoài ra, tính năng tìm kiếm theo truy vấn cho phép người dùng tìm kiếm thông tin theo các tiêu chí tùy chỉnh, giúp tìm kiếm kết quả chính xác hơn.

Các tính năng phân tích từ ngữ và phân tích ngữ nghĩa giúp tối ưu hóa truy vấn, giúp đảm bảo rằng các từ khóa trong truy vấn được xử lý đúng cách và kết quả truy vấn sẽ chính xác hơn. Cuối cùng, tính năng thực thi truy vấn phức tạp cho phép người dùng tìm kiếm thông tin với các tiêu chí phức tạp, bao gồm kết hợp các tiêu chí tìm kiếm, sắp xếp kết quả và giới hạn số lượng kết quả trả về.

6.2. SO SÁNH KẾT QUẢ VỚI THƯ VIỆN WHOOSH

Để so sánh với hai phương pháp trước đó, chúng ta cần hiểu rõ những đặc trưng và tính năng quan trọng của thư viện Whoosh trong lĩnh vực truy xuất thông tin trên Python.

Whoosh cung cấp các tính năng và đặc trưng quan trọng để xử lý các tác vụ truy xuất, bao gồm:

- *Tìm kiếm theo từ khóa:* Whoosh hỗ trợ tính năng tìm kiếm theo từ khóa, cho phép người dùng chỉ định các từ khóa trong truy vấn và tìm kiếm các tài liệu có chứa các từ khóa này. Kết quả trả về sẽ là các tài liệu chứa ít nhất một trong các từ khóa được chỉ định trong truy vấn
- *Tìm kiếm theo thuộc tính:* Whoosh cho phép người dùng xác định các thuộc tính cho các tài liệu trong bộ sưu tập và tìm kiếm các tài liệu theo thuộc tính này, chẳng hạn như tác giả, ngày xuất bản, v.v. Tính năng này giúp người dùng có thể tìm kiếm các tài liệu cụ thể dựa trên các thuộc tính của chúng, giúp đơn giản hóa quá trình truy xuất thông tin.
- *Tìm kiếm phủ định:* Người dùng có thể chỉ định các từ hoặc thuộc tính không muốn xuất hiện trong kết quả tìm kiếm khi sử dụng

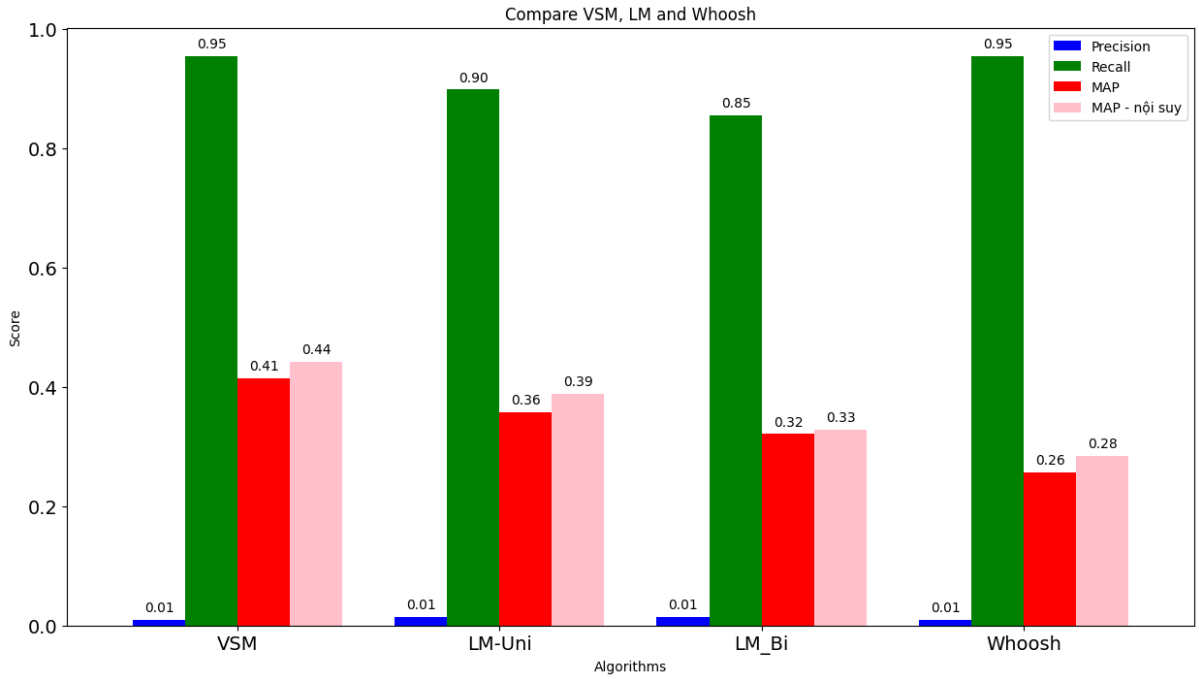
Whoosh. Ví dụ, nếu người dùng muốn tìm kiếm các tài liệu liên quan đến "máy tính", nhưng không muốn kết quả trả về chứa từ "laptop", họ có thể chỉ định từ "laptop" là từ không mong muốn trong truy vấn tìm kiếm.

- *Sắp xếp kết quả:* Whoosh hỗ trợ tính năng sắp xếp kết quả tìm kiếm theo một hoặc nhiều thuộc tính, chẳng hạn như thứ tự bảng chữ cái, số lượng lượt xem, v.v. Tính năng này giúp người dùng có thể xem kết quả tìm kiếm theo thứ tự ưu tiên của mình và tìm kiếm các tài liệu có giá trị cao hơn.
- *Phân tích ngôn ngữ tự nhiên:* Whoosh cung cấp các công cụ phân tích ngôn ngữ tự nhiên (NLP) để giúp người dùng xây dựng các hệ thống truy xuất thông tin thông minh hơn. Các công cụ NLP này giúp Whoosh hiểu được ngôn ngữ tự nhiên và cung cấp kết quả tìm kiếm chính xác hơn. Ví dụ, Whoosh có thể sử dụng công cụ NLP để phân tích từng từ trong truy vấn tìm kiếm và tìm kiếm các từ đồng nghĩa hoặc các biến thể của từ đó, giúp đảm bảo rằng kết quả trả về là chính xác và đầy đủ.

Whoosh hỗ trợ nhiều định dạng tài liệu khác nhau, bao gồm văn bản thuần túy, HTML, PDF, các tài liệu Microsoft Word và nhiều hơn nữa. Thư viện này cũng hỗ trợ nhiều loại truy vấn khác nhau, bao gồm các truy vấn đơn giản, truy vấn đa dạng và truy vấn phức tạp. Các ứng dụng của Whoosh rất đa dạng, bao gồm các hệ thống quản lý tài liệu, các trang web thương mại điện tử, các công cụ tìm kiếm cho các ứng dụng máy tính cá nhân. Tóm lại, Whoosh là một thư viện truy xuất thông tin đa năng và mạnh mẽ trên Python, hỗ trợ nhiều tính năng và định dạng

tài liệu khác nhau để giúp người dùng xử lý các tác vụ truy xuất thông tin một cách dễ dàng và hiệu quả.

Recall	Precision	MAP	MAP nội suy	Time process
0.954	0,01	0,257	0,284	16.310



Hình 23: Kết quả thực nghiệm của VSM, LM và thư viện Whoosh

Theo kết quả thu được ở bảng và biểu đồ trên, Cả hai mô hình Vector Space và Language Model cùng như thư viện Whoosh đều cho giá trị độ đo Recall khá cao (trên 0.85). Tuy nhiên, cả ba đều có Precision ở mức khá thấp so với Recall. Khi đánh giá khả năng xếp hạng các tài liệu trả về, cả hai mô hình nhóm đang sử dụng đều có hiệu suất tốt hơn nhiều so với thư viện Whoosh. Tuy nhiên, mô hình Vector Space có thời gian truy xuất lâu hơn rất nhiều so với thư viện Whoosh, trong khi đó, Language Model lại có thời gian truy xuất nhanh hơn đáng kể so với cả hai mô hình còn lại.

WHOOSH có kết quả truy xuất thấp hơn so với VSM và Language model có thể do một số lí do sau:

- WHOOSH sử dụng trọng số TF-IDF để truy xuất thông tin, điều này có thể không đủ để đánh giá mức độ tương đồng giữa các văn bản một cách chính xác.
- WHOOSH không phân tích ngữ cảnh và ý nghĩa của các từ trong văn bản và câu truy vấn, điều này có thể làm giảm độ chính xác của kết quả truy xuất.
- VSM và LMcó thể tích ngữ cảnh để cải thiện độ chính xác của kết quả truy xuất.

V. TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

1. TỔNG KẾT

Trong bài báo cáo môn học truy xuất thông tin, nhóm chúng em đã tiến hành chạy thực nghiệm một vài model, thư viện và các công cụ hỗ trợ truy xuất thông tin, bao gồm thư viện Whoosh và một số mô hình truy xuất thông tin khác (VSM, LM) để hiểu rõ hơn về một hệ thống truy xuất. Các công cụ này đều có những ưu điểm và hạn chế riêng, và người dùng cần lựa chọn công cụ phù hợp với nhu cầu của mình. Trong quá trình tiến hành thực nghiệm, nhóm chúng tôi đã sử dụng thư viện Whoosh và hai mô hình truy xuất thông tin phổ biến là Vector Space Model (VSM) và Language Model (LM) để so sánh và đánh giá hiệu suất của chúng.

Ngoài ra, chúng tôi cũng đã đánh giá khả năng xếp hạng các tài liệu trả về của ba công cụ và mô hình truy xuất thông tin. Kết quả cho thấy, cả hai mô hình VSM và LM đều có hiệu suất tốt hơn nhiều so với thư viện Whoosh trong việc xếp hạng các tài liệu trả về. Tuy nhiên, mô hình VSM lại có thời gian

truy xuất lâu hơn rất nhiều so với thư viện Whoosh, trong khi đó mô hình LM lại có thời gian truy xuất nhanh hơn đáng kể so với cả hai mô hình còn lại.

Tổng quan, các công cụ và mô hình truy xuất thông tin đang được phát triển và cải tiến liên tục với sự hỗ trợ của trí tuệ nhân tạo và các công nghệ liên quan. Tuy nhiên, để đạt được hiệu quả tốt nhất trong việc truy xuất thông tin, người dùng cần phải lựa chọn công cụ và mô hình phù hợp với nhu cầu và đặc điểm của tài liệu và truy vấn tìm kiếm.

2. HƯỚNG PHÁT TRIỂN

Cụ thể, chúng tôi đã sử dụng các độ đo như Precision và Recall để đánh giá độ chính xác của các công cụ và mô hình truy xuất thông tin. Kết quả cho thấy, cả ba công cụ đều cho giá trị độ đo Recall khá cao (trên 0.85), tuy nhiên cả ba đều có Precision ở mức khá thấp so với Recall. Điều này cho thấy các công cụ và mô hình truy xuất thông tin còn nhiều hạn chế và cần được cải tiến để đem lại hiệu quả tốt hơn.

Không chỉ như thế , với sự phát triển của công nghệ và trí tuệ nhân tạo, các công cụ truy xuất thông tin ngày càng trở nên thông minh và có khả năng xử lý các tác vụ phức tạp một cách hiệu quả hơn, giúp đáp ứng nhu cầu tìm kiếm thông tin của người dùng một cách nhanh chóng và chính xác. Ví dụ như các mô hình học máy như Deep Learning hay các thuật toán tối ưu hóa đang được sử dụng nhiều trong các công cụ truy xuất thông tin để cải thiện độ chính xác và hiệu quả của quá trình truy xuất. Do đó, trong tương lai nhóm chúng em sẽ tiếp tục nghiên cứu để khắc phục hạn chế này, nâng cao việc xử lý dữ liệu để tạo được kết quả recall tốt hơn và một hệ thống truy xuất hoàn chỉnh.

TÀI LIỆU THAM KHẢO

[1] Tf – Idf algorithm, Text retrieval and Search engines
https://viblo.asia/p/tf-idf-algorithm-text-retrieval-and-search-engines

[2] IR-Vector-Space-Model
https://blog.duyet.net/2019/08/ir-vector-space-model.html

[3] Web-information-retrieval-vector-space-model
https://www.geeksforgeeks.org/web-information-retrieval-vector-space-model/

[4] IR Models The Vector Space Model
https://redirect.cs.umbc.edu/~ian/irF02/lectures/07Models-VSM.pdf

[5] Vector space model an information retrieval systerm
https://www.researchgate.net/publication/362060638_VECTOR_SPACE_MODEL_AN_INFORMATION_RETRIEVAL_SYSTEM

[6] Applying Vector Space Model (VSM) Techniques in Information Retrieval for Arabic Language
https://arxiv.org/ftp/arxiv/papers/1801/1801.03627.pdf