BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

NGÔ THỊ HIỀN TRANG

NGHIÊN CỨU, THỬ NGHIỆM VÀ ĐÁNH GIÁ CÁC PHƯƠNG PHÁP XẾP HẠNG KẾT QUẢ TÌM KIẾM

Chuyên ngành: Khoa học máy tính
Mã số: 60.48.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

Đà Nẵng - Năm 2012

Công trình được hoàn thành tại
ĐẠI HỌC ĐÀ NẴNG

Người hướng dẫn khoa học: **TS. Huỳnh Công Pháp**

Phản biện 1:
TS. Trương Ngọc Châu

Phản biện 2:
TS. Trương Công Tuấn

Luận văn sẽ được bảo vệ tại Hội đồng chấm Luận văn tốt nghiệp Thạc sĩ kỹ thuật họp tại Đại học Đà Nẵng vào ngày 04 tháng 03 năm 2012.

** Có thể tìm hiểu luận văn tại:*

- Trung tâm Thông tin - Học liệu, Đại học Đà Nẵng
- Trung tâm Học liệu, Đại học Đà Nẵng.

MỞ ĐẦU

1. Lý do chọn đề tài

Hiện nay, Công nghệ Thông tin được ứng dụng rộng rãi trong nhiều lĩnh vực của đời sống xã hội. Dữ liệu được thu thập và lưu trữ trong quá trình ứng dụng công nghệ thông tin ngày càng được tích luỹ nhiều lên. Theo thống kê đến tháng 4/2010 số lượng máy chủ hơn 46 triệu máy, trên đó cài đặt hơn 240 triệu website [12]. Theo một tính toán khác, đến cuối năm 2009, đã có 20 tỷ trang Web đã được Google đánh chỉ mục [13].

Tìm kiếm thông tin là nhu cầu thiết thực của tất cả mọi người. Tuy nhiên, người sử dụng gặp nhiều khó khăn khi tiếp nhận kết quả trả về. Để hỗ trợ người dùng, các máy tìm kiếm thực hiện việc xếp hạng (ranking) các tài liệu để sắp xếp theo thứ tự ưu tiên. Có nhiều phương pháp đưa ra để thực hiện việc xếp hạng tài liệu nhưng chưa có đánh giá nào được thực hiện nhằm phân tích tính hiệu quả của các phương pháp này. Với lý do như vậy, tôi chọn đề tài "Nghiên cứu, thử nghiệm và đánh giá các phương pháp xếp hạng kết quả tìm kiếm" làm cơ sở cho việc chọn lựa phương pháp xếp hạng phù hợp.

2. Mục đích nghiên cứu

Mục đích của đề tài là tìm hiểu, đánh giá các phương pháp xếp hạng tài liệu để chọn lựa phương pháp xếp hạng phù hợp và sau đó là tiến hành thực nghiệm phương pháp xếp hạng đã lựa chọn. Để hoàn thành mục đích đề ra cần nghiên cứu các nội dung như sau:

- Về mặt lý thuyết: Tìm hiểu kiến thức về tìm kiếm thông tin (Information Retrieval), vai trò của xếp hạng (ranking) trong hệ thống tìm kiếm thông tin, các phương pháp xếp hạng tài liệu; tiêu chí đánh giá kết quả xếp hạng.

- Về mặt thực nghiệm: đánh giá các phương pháp xếp hạng và chọn lựa thực nghiệm phương pháp tốt nhất.

3. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu là các phương pháp xếp hạng tài liệu.
- Phạm vi nghiên cứu là thực nghiệm xếp hạng kết quả tìm kiếm đơn ngữ.

4. Phương pháp nghiên cứu

- Phương pháp phân tích: Thu thập và đánh giá độ liên quan giữa câu truy vấn và bộ dữ liệu.
- Phương pháp thực nghiệm: Thực hiện việc cài đặt, thử nghiệm phương pháp xếp hạng tài liệu; Đánh giá kết quả đạt được theo bảng đánh giá độ liên quan đã xây dựng.

5. Ý nghĩa khoa học và thực tiễn của đề tài

Sau khi thực hiện nghiên cứu và đánh giá hiệu quả các phương pháp xếp hạng kết quả trả về làm cơ sở cho việc lựa chọn mô hình xếp hạng phù hợp trong việc xây dựng một hệ truy tìm thông tin.

6. Cấu trúc luận văn

Nội dung chính của luận văn này được chia thành ba chương:

Chương 1 – Cơ sở lý thuyết

Các khái niệm cơ bản trong tìm kiếm thông tin.

Các khái niệm về Ma trận, giá trị riêng.

Chương 2 – Các phương pháp xếp hạng kết quả tìm kiếm

Nội dung chính là tìm hiểu các phương pháp, mô hình xếp hạng kết quả tìm kiếm. So sánh, đánh giá các phương pháp xếp hạng.

Chương 3 – Cài đặt thử nghiệm

Mô tả kiến trúc và cài đặt thử nghiệm hệ tìm kiếm thông tin theo mô hình chỉ mục ngữ nghĩa ngầm LSI.

CHƯƠNG 1
CƠ SỞ LÝ THUYẾT

1.1.CÁC KHÁI NIỆM CƠ BẢN

1.1.1. Tài liệu - Document

Tài liệu giữ vai trò trung tâm và là sản phẩm của quá trình tìm kiếm, chứa thông tin cần thiết. Việc tìm kiếm được thực hiện trên bộ sưu tập tài liệu (document collection).

1.1.2. Thuật ngữ - Term

Mỗi tài liệu được biểu diễn một cách lô-gic như một tập hợp các thuật ngữ (term). Các hệ thống tìm kiếm có các cách tiếp cận khác nhau. Một tài liệu tương ứng với tập hợp các từ, hay cụm từ chứa trong nó.

1.1.3. Lập chỉ mục cho tài liệu – Index

Lập chỉ mục cho tài liệu 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 đó cũng như các thông tin đi kèm với mỗi thuật ngữ (term) (vị trí, tần suất, độ quan trọng, …). Các thông tin này sẽ được tổ chức theo một cấu trúc dữ liệu riêng và được gọi là chỉ mục. Lúc này các thao tác tìm kiếm sẽ được tiến hành dựa trên chỉ mục thay vì được thực hiện trực tiếp trên file văn bản.

Chỉ mục của tài liệu (index) tương ứng với tập hợp các thuật ngữ chứa trong nó. Các tài liệu được biểu diễn dưới dạng:

	$\mathbf{t_1}$	$\mathbf{t_2}$	$\mathbf{t_3}$	$\mathbf{t_4}$			$\mathbf{t_m}$
$\mathbf{d_1}$	1	1	0	0			1
…	0	0	0	1			0
$\mathbf{d_n}$	1	0	0	0			0

trong đó d_i là tài liệu thứ i trong bộ sưu tập tài liệu (document collection), t_j là thuật ngữ thứ j chứa trong tài liệu. 1 thể hiện thuật ngữ t_j có chứa trong tài liệu d_i và 0 là ngược lại. Các số 1 trong bảng trên có thể thay bằng số lần xuất hiện của thuật ngữ trong tài liệu.

Trong khi đó, chỉ mục ngược (inverted index), mỗi thuật ngữ sẽ tương ứng với danh sách các tài liệu chứa nó.

$\mathbf{t_1}$	d_1	d_3	d_{51}	d_{151}	d_{2011}		
$\mathbf{t_2}$	d_2	d_{10}	d_{61}				
…							
$\mathbf{t_m}$	d_{100}	d_{1001}	d_{3000}	d_{3001}	d_{5001}		

1.1.4. Ma trận từ chỉ mục – Term - Document

Một tập văn bản có n văn bản được biểu diễn bởi m từ chỉ mục được vector hóa thành ma trận A – ma trận này được gọi là ma trận từ chỉ mục (*term document).* Trong đó n văn bản trong tập văn bản được biểu diễn thành n vector cột, m từ chỉ mục được biểu diễn thành m dòng. Phần tử d_{ij} của ma trận A chính là trọng số của từ chỉ mục i xuất hiện trong văn bản j. Thông thường, trong một tập văn bản số từ chỉ mục lớn hơn rất nhiều so với văn bản m >> n.

1.1.5. Trọng số của thuật ngữ - Term – weight

Dựa vào số lần xuất hiện của thuật ngữ của tài liệu (term count), tính ra tần suất xuất hiện của thuật ngữ (term frequency), với ký hiệu là tf_t.

Giá trị df_t (document frequency) tương ứng với số lượng tài liệu chứa thuật ngữ t.

Tần số nghịch đảo tài liệu (inverse document frequency), được tính bằng công thức: $idf_t = \log(\frac{N}{df_t})$. Trong đó, N là tổng số tài liệu, df_t là số tài liệu chứa thuật ngữ t.

Dựa trên các giá trị tf và idf, giá trị trọng số (term-weight) của một thuật ngữ trong một tài liệu được xác định bằng công thức: $w_{t,d} = tf_{t,d}*idf_t$.

Giá trị trọng số này được sử dụng trong ma trận từ chỉ mục, các giá trị khác 0 trong ma trận thể hiện trọng số của thuật ngữ trong tài liệu.

1.1.6. Truy vấn - Query

Truy vấn (query) là cách biểu diễn yêu cầu thông tin từ người sử dụng. Thông thường nó chứa các thuật ngữ và các toán tử kết hợp các thuật ngữ như AND, OR, LIKE, NEAR.

1.1.7. Sự phù hợp - Relevant

Một tài liệu được coi là phù hợp nếu người sử dụng đánh giá rằng nó chứa thông tin có giá trị phù hợp với nhu cầu tìm kiếm thông tin. Bên cạnh sự phụ thuộc vào tính chủ quan của người sử dụng, có nhiều kiểu phù hợp dựa trên nguồn tư liệu, cách biểu diễn yêu cầu cũng như ngữ cảnh tìm kiếm (context of the search).

1.2. HỆ TÌM KIẾM THÔNG TIN – Information Retrieval

1.2.1. Tổng quan về tìm kiếm thông tin và hệ thống tìm kiếm thông tin

Tìm kiếm thông tin (*Information Retrieval - IR*) là tìm kiếm tài nguyên trên một tập lớn các dữ liệu phi cấu trúc được lưu trữ trên máy tính nhằm thỏa mãn nhu cầu về thông tin.[2]

Để tìm kiếm thông tin, trước hết, hệ thống tìm kiếm xử lý tài liệu thô thành những tài liệu được **tách từ, phân đoạn (tokennized documents**) và sau đó **lập chỉ mục (index)** dựa trên vị trí của từ. Khi

người dùng đưa vào câu truy vấn, hệ thống tìm kiếm thông tin xử lý các câu truy vấn thành ngôn ngữ chỉ mục mô tả các yếu tố thông tin cần tìm kiếm và thực hiện đối chiếu với chỉ mục tài liệu để tìm ra các tài liệu liên quan. Cuối cùng, các tài liệu liên quan sẽ được trả về cho người dùng theo một **danh sách được sắp xếp** theo độ ưu tiên chính xác giảm dần (**ranked list**).

1.2.2. Cách thức hoạt động của hệ tìm kiếm thông tin

1.2.3. Các bộ phận cấu thành của hệ tìm kiếm thông tin

Một hệ thống tìm kiếm thông tin hoạt động trên môi trường mạng (internet) hay trên môi trường máy tính cá nhân (PC) đều gồm có các thành phần chính sau:

1.2.3.1. Bộ thu thập thông tin - Crawler

1.2.3.2. Bộ lập chỉ mục – Index

1.2.3.3. Bộ tìm kiếm thông tin – Search Engine

1.2.4. Mục tiêu của hệ tìm kiếm thông tin

1.2.5. Tách từ

1.3. ĐÁNH GIÁ CÁC HỆ THỐNG TÌM KIẾM THÔNG TIN

1.3.1. Nền tảng đánh giá các hệ tìm kiếm thông tin

1.3.2. Khái niệm về độ liên quan giữa câu truy vấn và tài liệu

Độ liên quan là một khái niệm đa khía cạnh (multifaceted), đa chiều (multidimension). Theo nghiên cứu có nhiều loại độ liên quan. Độ liên quan mang tính chủ quan, và phụ thuộc vào tính cá nhân hoặc nhân tố thời gian.

Có hai loại độ liên quan:

- Độ liên quan nhị phân (binary relevance): là độ liên quan chỉ có 2 giá trị: hoặc là có liên quan (relevant _ 1), hoặc không có liên quan (not relevant _ 0).

- Độ liên quan nhiều mức độ (độ liên quan đa cấp độ): độ liên quan được xét ở nhiều mức độ, có nhiều giá trị.

Trong hầu hết các thử nghiệm đánh giá hệ thống tìm kiếm thông tin người ta thường quan tâm độ liên quan nhị phân (tài liệu có liên quan (1) hoặc không có liên quan (0)).

1.3.2. Các tiêu chí đánh giá hiệu quả hệ truy tìm thông tin

Để đánh giá hiệu quả của hệ truy tìm thông tin có thể dựa theo các tiêu chuẩn sau [5]:

- Dựa trên hai độ đo :

Độ chính xác (Precision): được đo bởi tỉ lệ của tài liệu trả về chính xác trên tổng các tài liệu nhận được.

Độ bao phủ (Recall): được đo bởi tỉ lệ của tài liệu trả về chính xác trên tổng các tài liệu có liên quan.

- Hiệu quả thực thi của hệ thống(*Execution efficiency*) được đo bởi thời gian thực hiện thủ tục tìm kiếm các văn bản liên quan đến câu truy vấn được cho.
- Hiệu quả lưu trữ được đo bởi dung lượng bộ nhớ cần thiết để lưu trữ dữ liệu.

1.4. ĐẠI SỐ TUYẾN TÍNH

1.4.1. Định nghĩa các loại ma trận

1.4.2. Các phép toán cơ bản trên ma trận

1.4.3. Tính định thức của Ma trận

1.4.4. Tính hạng của Ma trận

1.4.5. Giải HPTTT bằng phương pháp GAUSS

1.4.6. Tính trị riêng và vector riêng của Ma trận

1.4.6.1. Định nghĩa

1.4.6.2. Cách tính trị riêng và vector riêng

CHƯƠNG 2
XẾP HẠNG TRONG CÁC MÔ HÌNH TÌM KIẾM THÔNG TIN

Các mô hình bao gồm: mô hình so khớp (Boolean model), mô hình tính điểm trọng số(term-weight), mô hình không gian vec-tơ (Vector Space Model), mô hình chỉ mục ngữ nghĩa ngầm (Latent Sematic Indexing), mô hình xác suất (Probabilistic model). Trừ mô hình Boolean, trong các mô hình khác sử dụng các công thức xếp hạng, cho phép người sử dụng nhập câu truy vấn và nhận được danh sách các tài liệu được xếp hạng theo mức độ phù hợp [8].

2.1. MÔ HÌNH SO KHỚP CHÍNH XÁC – Boolean Model

2.1.1. Giới thiệu

Đây là mô hình sử dụng nguyên tắc so sánh chính xác khi tìm kiếm tài liệu. Hệ thống yêu cầu người sử dụng cung cấp câu truy vấn dưới hình thức là các từ khoá kèm theo các toán tử AND, OR, NOT.

2.1.2. Cách tổ chức dữ liệu

Một tập văn bản có n văn bản được biểu diễn bởi m từ chỉ mục được vector hóa thành ma trận A – ma trận này được gọi là ma trận từ chỉ mục (*term document).* Trong đó n văn bản trong tập văn bản được biểu diễn thành n cột, m từ chỉ mục được biểu diễn thành m dòng. Phần tử d_{ij} của ma trận A là hai giá trị 1 hoặc 0. Một ma trận nhị phân mục từ với giá trị 1 biểu diễn mục từ k_i có trong tài liệu d_i và 0 là ngược lại.

	Antony and Cleopatra	**Julius Caesar**	**The Tempest**	**Hamlet**	**Othello**	**Macbeth**	**…**
Antony	1	1	0	0	0	1	…

Brutus	1	1	0	1	0	0	…
Caesar	1	1	0	1	1	1	…
Mercy	1	0	1	1	1	1	…
Worser	1	0	1	1	1	0	…
…	…	…	…	…	…	…	…

Hình 2.1 Ví dụ ma trận mục từ cho các tác phẩm của Shakespeare

2.1.3. Truy vấn trong mô hình Boolean

Trong mô hình Boolean, câu truy vấn được thiết lập bằng cách các mục từ kết hợp với các toán tử AND, OR, NOT. *Ví dụ: Brutus AND Caesar AND NOT Calpurnia*. Để truy vấn trong mô hình Boolean: dựa trên ma trận nhị phân mục từ và câu truy vấn thực hiện lấy các vector mục từ và so khớp theo toán tử bit.

Giả sử có ma trận nhị phân mục từ như hình 2.1. Để trả lời cho câu truy vấn *Brutus AND Caesar AND NOT Calpurnia*, chúng ta thực hiện lấy các vector và so khớp theo toán tử bit như sau:

Vector mục từ *Brutus* trên ma trận tương đương: 110100. Tương tự *Caesar* tương đương: 110111, *Calpurnia*: 010000

Thực hiện so khớp các toán tử bít như sau: *Brutus* AND *Caesar* AND NOT *Calpurnia*. Tương đương với: 110100 AND 110111 AND NOT 010000 = **100100**

Sau khi thực hiện so khớp các giá trị 1 tương đương với cột thứ i (văn bản thứ i) trong ma trận mục từ thoả mãn điều kiện. Như vậy kết quả trả lời sẽ là **Antony and Cleopatra** (d_1) và **Hamlet (**d_4**)**.

2.1.4. Đánh giá mô hình Boolean

Ưu điểm:

- Đơn giản và dễ sử dụng.

Nhược điểm:

- Chuyển câu truy vấn sang dạng boolean là không đơn giản;
- Văn bản trả về không quan tâm đến thứ tự quan hệ với câu truy vấn.

2.2. MÔ HÌNH TÍNH ĐIỂM VÀ TRỌNG SỐ CHO MỤC TỪ - TERM WEIGHT

2.2.1. Giới thiệu

Mô hình so khớp chính xác chỉ trả về giá trị logic là có hoặc không có trong tài liệu tìm kiếm, kết quả trả về không có thứ hạng. Để cải tiến mô hình này, người ta áp dụng cách tính điểm cho kết quả trả về, dựa trên trọng số của mục từ trên tài liệu.

Mỗi mục từ trong ma trận từ chỉ mục được gán một trọng số, giá trị này phụ thuộc vào số lần xuất hiện của mục từ trên tài liệu chứa mục từ và tập tài liệu. Tính kết quả độ liên quan của câu truy vấn trên từng văn bản và sau đó sắp xếp kết quả trả về.

2.2.2. Cách tổ chức dữ liệu

Một ma trận mục từ được xây dựng với n cột tương ứng với n văn bản trong tập tài liệu, m dòng tương ứng với m mục từ. Phần tử d_{ij} của ma trận A thay vì chỉ có 2 giá trị là 1 hoặc 0 như trong mô hình Boolean được thay bằng trọng số của mục từ (term weight). Trọng số của mục từ được tính bằng công thức (2.1)

2.2.3. Công thức tính trọng số của từ chỉ mục

Định nghĩa một hàm tính trọng số của từ chỉ mục như sau:

$$w_{ij} = l_{ij} * g_i * n_j \qquad (2.1)$$

Trong đó:

l_{ij} : hàm đếm số lần xuất hiện của từ chỉ mục trong một VB.

g_i là trọng số toàn cục của từ chỉ mục *i* - là hàm đếm số lần xuất hiện của mỗi từ chỉ mục trong toàn bộ tập văn bản

$\mathbf{n_j}$ là hệ số được chuẩn hoá của văn bản j - là hệ số cân bằng chiều dài của các văn bản trong tập văn bản.

2.2.3.1. Các công thức tính trọng số cục bộ l_{ij}

2.2.3.2. Các công thức tính trọng số toàn cục g_i

2.2.3.3. Công thức tính hệ số chuẩn hoá n_j

2.2.4. Cách truy vấn trong mô hình tính điểm, trọng số mục từ

Điểm số của tài liệu d là tổng điểm của các mục từ trên câu truy vấn q có mặt trong tài liệu d. Truy vấn trong mô hình tính điểm và trọng số được tính theo công thức: $Score(q,d_i) = \sum wq_{ij}$

Ví dụ 2.2: với 1000 tài liệu có 100 tài liệu chứa mục từ "tin" và 150 tài liệu chứa mục từ "học", giả sử tài liệu thứ nhất d có 3 lần xuất hiện mục từ "tin" và 4 lần xuất hiện mục từ "học", khi đó điểm số của câu truy vấn q=tin học trên tài liệu d sẽ là:

$$Score(q,d) = tf_{tin,d} - idf_{tin} + tf_{học,d} - idf_{học}$$

$$= tf_{tin,d} * \log\frac{N}{df_{tin}} + tf_{học,d} * \log\frac{N}{df_h}$$

$$= 3 * \log(1000/100) + 4 * \log(1000/150) = 6.23$$

2.2.5. Đánh giá mô hình tính điểm, trọng số mục từ

Ưu điểm:

- Trọng số từ chỉ mục không giới hạn bởi hai trị 0 hoặc 1, các trọng số này được sử dụng để tính toán độ đo tương tự của mỗi văn bản với câu truy vấn. Kết quả trả về có quan tâm đến thứ tự xuất hiện.

Nhược điểm:

- Kết quả tính trọng số chưa xét vai trò của các mục từ trong câu truy vấn. Có thể số lượng các mục từ như nhau nhưng vai trò khác nhau hoàn toàn.

2.3. MÔ HÌNH KHÔNG GIAN VECTOR – Vector Space Model

2.3.1. Giới thiệu

Mô hình không gian vector được phát triển bởi Gerard Salton, trong đó tài liệu và câu truy vấn được biểu diễn dưới dạng các vector. Một văn bản d được biểu diễn như một vector của các từ chỉ mục $d = (t_1, t_2, \ldots, t_n)$. Tương tự, câu truy vấn cũng được biểu diễn như một vector $q = (t_1, t_2, \ldots, t_n)$. Sau khi biểu diễn tập văn bản và câu truy vấn thành các vector trong không gian vector, sử dụng độ đo cosin để tính độ đo tương tự giữa các vector văn bản và vector truy vấn. Kết quả sau khi tính toán được dùng để xếp hạng độ liên quan giữa văn bản và câu truy vấn.

2.3.2. Số hoá tập văn bản

2.3.2.1. Cách tổ chức dữ liệu – Ma trận từ chỉ mục

Trong mô hình không gian vector, một tập văn bản có n văn bản được biểu diễn bởi m từ chỉ mục được vector hóa thành ma trận A – ma trận này được gọi là ma trận từ chỉ mục (*term document).* Trong đó n văn bản trong tập văn bản được biểu diễn thành n vector cột, m từ chỉ mục được biểu diễn thành m dòng. Do đó phần tử d_{ij} của ma trận A chính là trọng số của từ chỉ mục i xuất hiện trong văn bản j.

2.3.2.2. Công thức tính trọng số của từ chỉ mục

Trong ma trận từ chỉ mục, các phần tử của ma trận trọng số của từ chỉ mục i đối với tập văn bản được tính bằng công thức:

$$w_{ij} = l_{ij} * g_i * n_j$$

2.3.3. Truy vấn trong mô hình không gian vector

Trong mô hình không gian vector, một câu truy vấn được xem như tập các từ chỉ mục và được biểu diễn như các văn bản trong tập văn bản. Số lượng từ chỉ mục câu truy vấn ngắn là rất ít so với số

lượng từ chỉ mục nên có rất nhiều từ chỉ mục của tập văn bản không xuất hiện trong câu truy vấn, có nghĩa là hầu hết các thành phần của vector truy vấn là 0. Thủ tục truy vấn chính là tìm các văn bản trong tập văn bản liên quan với câu truy vấn hay còn gọi là các văn bản có độ đo tương tự "cao" với câu truy vấn. Theo cách biểu diễn hình học, các văn bản được chọn là các văn bản gần với câu truy vấn nhất theo một độ đo (*measure*) nào đó. Độ đo thường được sử dụng nhất là độ đo *cosin* của góc giữa vector truy vấn và vector văn bản được tính theo công thức:

$$\cos\theta_j = \frac{d_j^T q}{\|d_j\|_2 \|q\|_2} = \frac{\sum_{i=1}^{m} d_{ij} q_i}{\sqrt{\sum_{i=1}^{m} d_{ij}^2} \sqrt{\sum_{i=1}^{m} q_i^2}}$$

Trong đó d_{ij} là giá trị trọng số của phần tử trong ma trận từ chỉ mục; q_i là giá trị trọng số của phần tử thứ i trong vector câu truy vấn.

2.3.4. Đánh giá mô hình không gian vector

Ưu điểm:

- Đưa ra khái niệm phù hợp một phần; công thức xếp hạng cô-sin cho phép đồng thời xác định sự phù hợp và phục vụ sắp xếp danh sách kết quả..

Nhược điểm:

- Số chiều biểu diễn cho tập văn bản có thể rất lớn nên tốn nhiều không gian lưu trữ;
- Không xét quan hệ về ngữ nghĩa với câu truy vấn.

2.4. MÔ HÌNH XÁC SUẤT - Probabilistic model

2.4.1. Giới thiệu

Cho câu truy vấn của người dùng q và văn bản d trong tập văn bản. Mô hình xác suất tính xác suất mà văn bản *d* liên quan đến cấu truy vấn của người dùng. Mô hình giả thiết xác suất liên quan của một văn bản với câu truy vấn phụ thuộc cách biểu diễn chúng. Tập văn bản kết quả được xem là liên quan và có tổng xác suất liên quan với câu truy vấn lớn nhất [11].

2.4.2. Mô hình tìm kiếm nhị phân độc lập - Binary independence retrieval -BIR

2.4.3. Mô hình mức độ đáng kể (eliteness)

2.4.4. Công thức BM25

2.4.5. Đánh giá mô hình xác suất

2.5. MÔ HÌNH CHỈ MỤC NGỮ NGHĨA NGẦM - LSI

2.5.1. Giới thiệu

Latent Semantic Indexing (LSI) là phương pháp tạo chỉ mục ngữ nghĩa ngầm dựa trên khái niệm để khắc phục hai hạn chế tồn tại trong mô hình không gian vector chuẩn về vấn đề đồng nghĩa (synoymy) và đa nghĩa (polysemy) [14]. Với synoymy, nhiều từ có thể được sử dụng để biểu diễn một khái niệm, vì vậy hệ thống không thể trả về những văn bản liên quan đến câu truy vấn của người dùng khi họ sử dụng những từ trong câu truy vấn đồng nghĩa với những từ trong văn bản. Với polysemy, một từ có thể có nhiều nghĩa, vì vậy hệ thống có thể trả về những văn bản không liên quan. Điều này thực tế rất thường xảy ra bởi vì các văn bản trong tập văn bản được viết bởi rất nhiều tác giả, với cách dùng từ rất khác nhau. Một cách tiếp cận tốt hơn cho phép người dùng truy vấn văn bản dựa trên khái niệm (concept) hay nghĩa (meaning) của văn bản.

Mô hình LSI khắc phục hai hạn chế trên trong mô hình không gian vector bằng cách chỉ mục khái niệm được tạo ra bởi phương

pháp phân tích giá trị đơn (Single Value Decomposition - SVD) từ ma trận từ chỉ mục (term – document A).

2.5.2. Phân tích giá trị đơn (Single Value Decomposition - SVD) của ma trận từ chỉ mục

Vấn đề cơ bản của mô hình LSI là dùng kỹ thuật phân huỷ giá trị đơn SVD trên ma trận từ chỉ mục để tạo ra một ma trận ngữ nghĩa. Mục đích của việc phân tích SVD là phát hiện ra mối quan hệ ngữ nghĩa trong cách dùng từ trong toàn bộ văn bản $A = U\Sigma V^T$ và giảm số chiều ma trận sau khi phân tích.

Đầu tiên, từ tập dữ liệu xây dựng ma trận từ chỉ mục được biểu diễn trong đó mỗi dòng tương ứng với một từ chỉ mục (term) xác định quan hệ (số lần xuất hiện, hay trọng số) của thuật ngữ đối với các tài liệu. Tương tự, mỗi cột biểu diễn cho 01 tài liệu.

Tiếp theo, LSI áp dụng kỹ thuật phân hủy giá trị đơn (SVD) trên ma trận từ chỉ mục. Ma trận từ chỉ mục A bị phân hủy thành sản phẩm của ba ma trận khác: $A = U\Sigma V^T$.

Khi rút gọn ma trận ∑, giữ lại một số k phần tử đầu tiên và rút gọn tương ứng các ma trận U và V^T, sẽ tạo ra một xấp xỉ gần đúng cho ma trận từ chỉ mục A.

2.5.3. Chọn hệ số k trong mô hình LSI

Trong mô hình LSI, việc chọn hệ số k để xây dựng ma trận xấp xỉ là một việc hết sức quan trọng đến hiệu quả của thuật toán. Theo các tài liệu nghiên cứu về LSI [6] qua thực nghiệm trên các tập dữ liệu văn bản cụ thể, các tác giả chọn k từ 50 đến 100 cho các tập dữ liệu nhỏ và từ 100 đến 300 cho các tập dữ liệu lớn.

Một phương pháp đề nghị chọn hệ số k gần đây nhất (2003) được đưa ra bởi Miles Efron trong tài liệu [26], tác giả sử dụng phương pháp phân tích giá trị riêng (Eigenvalue) của ma trận từ chỉ

mục và sử dụng kiểm định thống kê để chọn hệ số k tốt nhất trên dãy các hệ số k được chọn thử nghiệm.

2.5.4. Truy vấn trong mô hình LSI

Để truy vấn trong mô hình LSI: Tính độ đo *cosines* của các góc giữa vector truy vấn *q* và các vector văn bản trong ma trận xấp xỉ A_k *(Độ đo cô-sin được tính theo công thức trong mô hình không gian vector).* Hoặc các văn bản có thể được so sánh với nhau bằng cách tính độ đo *cosines* các vector văn bản trong "không gian văn bản" (*document space*) – chính là so sánh các vector cột trong ma trận V_k^T. Một câu truy vấn *q* được xem như là một văn bản và giống như một vector cột được thêm vào ma trận V_k^T. Để thêm *q* như một cột mới vào V_k^T ta phải chiếu *q* vào không gian văn bản *k* chiều.

Từ công thức: $A = U\Sigma V^T$

$\Rightarrow$ $A^T = (U\Sigma V^T)^T = V\Sigma U^T$

$\Leftrightarrow$ $A^T U\Sigma^{-1} = V\Sigma U^T U\Sigma^{-1}$

$\Rightarrow$ $V = A^T U\Sigma^{-1}$

Ma trận V gồm n dòng (n>1), mỗi dòng của ma trận V thể hiện 01 vector tài liệu d: $d = d^T U\Sigma^{-1}$

Việc giảm chiều trong không gian k chiều, vector d có thể được viết lại như sau: $d = d^T U_k \Sigma_k^{-1}$

Một câu truy vấn *q* được xem như là một văn bản và giống như một vector cột được thêm vào ma trận V_k^T. Để thêm *q* như một cột mới vào V_k^T ta phải chiếu *q* vào không gian văn bản *k* chiều: $q = q^T U_k \Sigma_k^{-1}$

Tính độ liên quan giữa vector truy vấn q và vector tài liệu d_i trong ma trận V_k^T bằng công thức sau:

$$sim(q,d) = sim(q^T U_k \Sigma_k^{-1}, d^T U_k \Sigma_k^{-1}) = \frac{q.d}{|q|.|d|}$$

Sắp kết quả trả về theo giảm dần độ liên quan.

2.5.5. Cập nhật giá trị trong mô hình LSI

Thông tin thì luôn luôn được thêm vào hay bị xóa đi, điều đó có nghĩa rằng ma trận chỉ mục cũng luôn bị biến động. Trong mô hình LSI, khi có một văn bản mới được thêm vào hay bị xóa đi đều ảnh hưởng đến việc tính toán lại giá trị trong ma trận từ chỉ mục và ma trận xấp xỉ thông qua kỹ thuật phân tích SVD. Đối với các ma trận lớn, việc tính toán lại tốn rất nhiều chi phí và thời gian.

2.5.5.1. Cập nhật văn bản (SVD- Updating document)

2.5.5.2. Cập nhật từ chỉ mục (SVD- Updating terms):

2.5.5.3. Xoá từ chỉ mục(Downdating)

2.5.6. Đánh giá mô hình LSI

Ưu điểm:

- LSI là phương pháp tạo chỉ mục tự động dựa trên khái niệm để khắc phục hạn chế tồn tại trong mô hình không gian vector về hai vấn đề đồng nghĩa (synoymy) và đa nghĩa (polysemy) [9];
- Việc giảm số chiều cải thiện đáng kể chi phí lưu trữ và thời gian thực thi.

Nhược điểm:

- Việc tìm kiếm cũng phải quét qua tất cả các cột trong ma trận LSI nên cũng tốn nhiều chi phí và thời gian.

2.6. ĐÁNH GIÁ CÁC MÔ HÌNH XẾP HẠNG

2.6.1. Đánh giá theo lý thuyết

Do tính hiệu quả thấp của mô hình Boolean, mô hình xác suất, nên hiện nay mô hình VSM và mô hình LSI đang được nghiên cứu phục vụ cho việc xây dựng các hệ thống IR hiện đại [6]. Mô hình LSI được đưa ra để khắc phục những hạn chế của mô hình VSM là vấn đề

đồng nghĩa và đa nghĩa. Hiệu quả của mô hình LSI được đánh giá là cao hơn so với mô hình VSM [6], [7].

2.6.2. Đánh giá theo thử nghiệm trên hai mô hình VSM và LSI

Như đã trình bày trong chương 1, hiệu quả của một hệ IR cơ bản được đánh giá dựa trên 3 tiêu chuẩn: hiệu quả truy tìm, hiệu quả lưu trữ dữ liệu chỉ mục; Thời gian thực hiện thủ tục truy vấn.

2.6.2.1. Đánh giá hiệu quả truy tìm

Trên thực tế việc sử dụng hai độ đo *precision* và *recall* để đánh giá hiệu quả của hệ thống bất kỳ là rất khó, vì thực tế không thể xác định được số văn bản liên quan đến câu truy vấn cụ thể trong tập văn lớn là bao nhiêu, chỉ có thể thực hiện điều này trên tập văn bản nhỏ, được chọn lựa và phân loại chi tiết. Một khó khăn nữa gặp phải là trong việc đánh giá kết quả trả về của tập văn bản liên quan đến câu truy vấn phụ thuộc rất nhiều vào tính chủ quan của người đánh giá và nhu cầu. Vì vậy chỉ đánh giá và so sánh hiệu quả của hệ IR bằng cách so sánh tổng số văn bản liên quan được trả về của hai hệ VSM_IR và LSI_IR khi thử nghiệm trên cùng một tập câu truy vấn.

2.6.2.2. Đánh giá dung lượng lưu trữ dữ liệu chỉ mục

Dung lượng bộ nhớ RAM cho mỗi hệ IR lưu trữ dữ liệu chỉ mục khi thực thi được đo bởi ma trận chỉ mục. Công thức tính sau:

RAM = (<*số văn bản*> x <*số từ chỉ mục*>) x (sizeof(<*kiểu dữ liệu*>))

2.6.2.3. Đánh giá thời gian thực thi thủ tục truy vấn

2.6.3. Xác định mô hình cài đặt thử nghiệm

Qua các phân tích đánh giá, đề tài xác định mô hình cho việc cài đặt thử nghiệm là mô hình xếp hạng tài liệu pheo phương pháp chỉ mục ngữ nghĩa tiềm ẩn LSI.

CHƯƠNG 3
CÀI ĐẶT THỬ NGHIỆM HỆ IR THEO MÔ HÌNH LSI

3.1. MÔ TẢ KIẾN TRÚC HỆ IR THEO MÔ HÌNH LSI

Hình 3.1 sau mô tả kiến trúc hệ tìm kếm theo mô hình LSI, gồm các bước:

- Xử lý văn bản và tạo các tập tin chỉ mục từ (*Term_ Index.out)* và tập tin chỉ mục văn bản (*Doc_ Index.out)*
- Tạo ma trận chỉ mục từ (Term – Document A)
- Tính SVD ma trận chỉ mục từ (Term – Document) $A = U\Sigma V^T$
- Chọn hệ số k
- Tạo ma trận xấp xỉ $A_k = U_k \Sigma_k V_k^T$
- Xử lý truy vấn
- Xếp hạng kết quả trả về theo thứ tự giảm dần độ đo cosines

3.2. ĐẶT TẢ CÁC BƯỚC XÂY DỰNG HỆ LSI-IR

3.2.1. Xây dựng file từ chỉ mục

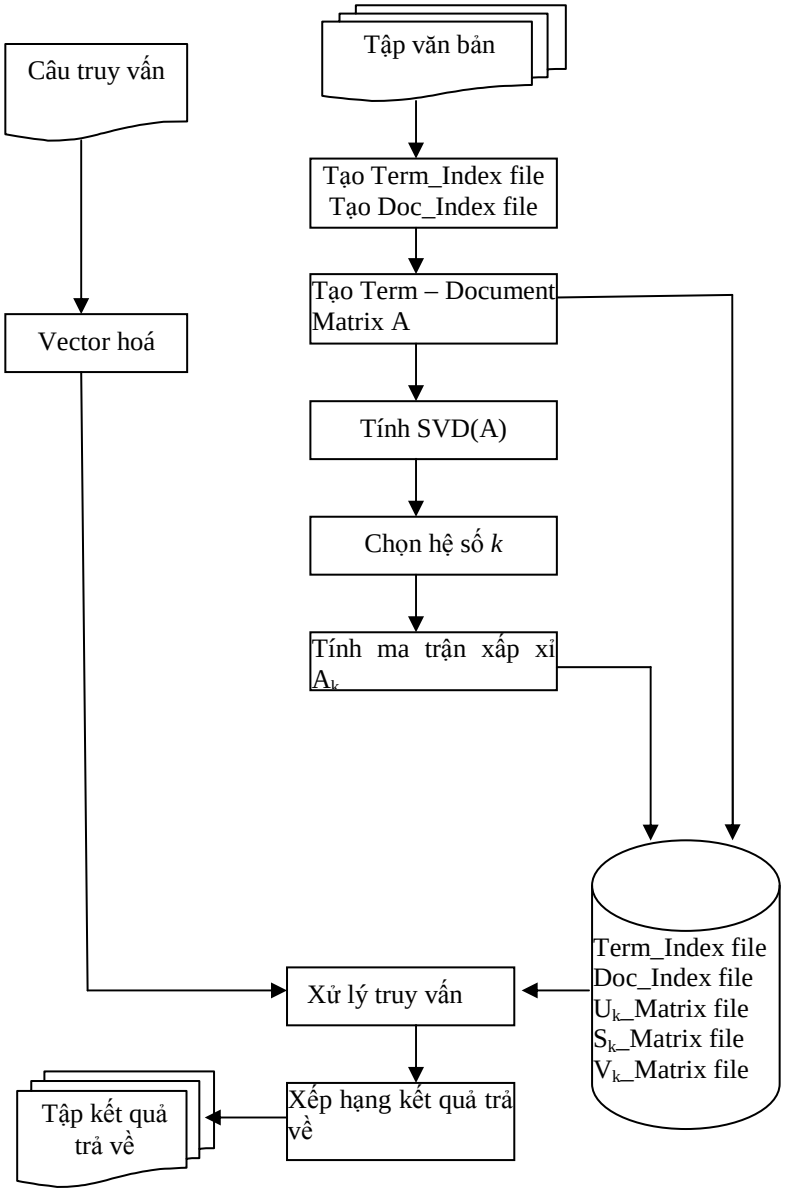
3.2.2. Xây dựng ma trận từ chỉ mục

3.2.3. Phân tích SVD ma trận từ chỉ mục A

3.2.4. Xác định hệ số k

3.2.5. Xây dựng ma trận xấp xỉ A_k

3.2.6. Thực hiện truy vấn và xếp hạng kết quả trả về



Hình 3.1 Kiến trúc hệ LSI-IR

3.3. BỘ DỮ LIỆU THỬ NGHIỆM VÀ MÔI TRƯỜNG PHÁT TRIỂN

3.3.1. Bộ dữ liệu thử nghiệm

Bộ dữ liệu phục vụ thử nghiệm hệ thống: tập Cranfield collection được lấy từ Internet [24] với kích thước

- Tập văn bản (documnetn collection):1.400 văn bản, kích thước 1.57MB
- Tập truy vấn (query): 365 câu truy vấn, kích thước 28KB.
- Bảng đánh giá độ liên quan giữa câu truy vấn và văn bản
- 3763 từ chỉ mục trên tập văn bản, kích thước 1.98MB
- Hệ số k cho mô hình LSI: k=185. Hệ số này đã được kiểm thử có hiệu quả nhất trên tập CRAN [24].

3.3.2. Môi trường cài đặt hệ thống

3.4. KẾT QUẢ THỬ NGHIỆM

3.4.1. Bộ dữ liệu

3.4.2. Ma trận từ chỉ mục

3.4.3. Bộ câu hỏi thực hiện truy vấn

3.4.4. Bảng đánh giá độ liên quan giữa bộ câu hỏi trên tập dữ liệu thử nghiệm

3.4.5. Đánh giá kết quả thử nghiệm

Kết quả thử nghiệm độ đo Precision trên tập dữ liệu 1400 văn bản và 3763 từ chỉ mục với 20 câu truy vấn. Chọn hệ số *k* = 185 cho mô hình LSI.

Bảng 3.2 Độ đo Precision trung bình của mô hình LSI với k=185

STT	Câu truy vấn	Precision LSI
1	001	75%
2	002	56%

3	003	79%
4	004	74%
5	005	78%
6	006	93%
7	007	88%
8	008	94%
9	009	100%
10	010	94%
Precision trung bình		**81%**

Qua kết quả thử nghiệm trên tập dữ liệu 1400 văn bản và 3763 từ chỉ mục với 20 câu truy vấn và căn cứ vào bảng đánh giá độ liên quan, kết quả đạt được của độ đo precision trung bình là 81% .

Với việc thử nghiệm trên cùng một tập câu truy vấn cho cả hai hệ IR, thời gian cho thủ tục tìm kiếm trên LSI_IR nhanh hơn trên dưới 30 lần so với VSM_IR. Hệ VSM thời gian tìm kiếm là 13.344 giây, hệ LSI là 0.407 giây.

Dung lượng bộ nhớ RAM cho mỗi hệ IR lưu trữ dữ liệu chỉ mục khi thực thi được đo bởi ma trận chỉ mục.

- Với hệ VSM_IR, ma trận chỉ mục *A* (1400 x 3763) mỗi phần tử ma trận có kiểu *float* trong java chiếm 4 byte.

RAM = (1400 x 3763) x 4(byte) = 20MB

- Với LSI_IR lưu ba ma trận $U_{3763x185}$, $\Sigma_{185*185}$, và $V^T_{185*1400}$.

RAM =(3763 x 185 + 185 x 185 + 185 x 1400) x 4(byte) = 3.8 MB

Với kết quả như trên: có thể thấy rằng dung lượng lưu trữ dữ liệu chỉ mục của mô hình LSI giảm hơn 90% so với VSM. Điều này cho thấy thông qua kỹ thuật phân huỷ VSD chi phí lưu trữ giảm đi rất nhiều.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Đề tài "Nghiên cứu, thử nghiệm và đánh giá các phương pháp xếp hạng kết quả tìm kiếm" đã tập trung nghiên cứu các phương pháp xếp hạng tài liệu theo các mô hình khác nhau như: mô hình không gian vector VSM, chỉ mục ngữ nghĩa LSI, các công thức và cách kết hợp giữa các công thức phục vụ cho việc tính trọng số của từ chỉ mục. Từ những nghiên cứu về lý thuyết này đã đưa ra được kiến trúc cơ bản của một hệ IR dựa trên mô hình LSI.

Đánh giá hiệu quả thực thi của hai mô hình về các tiêu chí hiệu quả truy tìm, thời gian và dung lượng bộ nhớ cần thiết lưu trữ dữ liệu số hoá cho mỗi mô hình. Từ đó, thấy được hiệu quả của mô hình ngữ nghĩa LSI cao hơn so với mô hình không gian vector rất nhiều. Từ kết quả này, hỗ trợ cho việc xây dựng các hệ IR thực tế có hiệu quả truy tìm cao. Những kết quả đạt được làm cơ sở lý thuyết và thực nghiệm cho việc xây dựng các hệ IR thực tế hoạt động hiệu quả về sau.

2. Hướng phát triển

Trong mô hình LSI, việc phân tích SVD cho ma trận từ chỉ mục trong mô hình không gian vector làm giảm đi số chiều của ma trận A rất nhiều và việc giải quyết được quan hệ ngữ nghĩa các văn bản liên quan đến câu truy vấn mà được xem là điểm yếu trong mô hình không gian vector, nên mô hình LSI được đánh giá rất cao. Tuy vậy, để trả về các văn bản liên quan thì cũng phải đi so sánh với tất cả các văn bản trong ma trận xấp xỉ A_k. Điều này dẫn đến việc hạn chế tốc độ tìm kiếm của giải thuật. Để khắc phục điều này, đề nghị

một phương pháp, là trước khi thực hiện tính Cosines giữa vector truy vấn với các vector văn bản trong ma trận A_k ta tiến hành gom cụm văn bản trước trong ma trận A_k. Kết hợp LSI vào trong bài toán gom cụm văn bản.

Đối với mô hình LSI hiệu quả truy tìm của hệ thống cũng như hiệu quả về dung lượng lưu trữ và thời gian tìm kiếm phụ thuộc vào việc chọn hệ số *k*. Bài toán này hiện nay vẫn đang là bài toán mở chưa có lời giải tổng quát, chỉ giải quyết bằng thực nghiệm trên tập dữ liệu cụ thể. Hướng phát triển tương lai là sử dụng các công cụ toán học về tối ưu hoá để giải quyết bài toán chọn hệ số *k* sao cho hệ thống hoạt động tối ưu trong mô hình LSI này.