CƠ SỞ DỮ LIỆU VĂN BẢN
Text/Document Databases
Hệ cơ sở dữ liệu đa phương tiện

HK1, 2023 - 2024

Giới thiệu

Mỗi tài liệu văn bản là chuỗi các từ

`Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains.`

- Từ đồng nghĩa: xem – coi, siêng năng – chăm chỉ - cần cù
- Từ đa nghĩa: mũi (mũi người, mũi dao...)
- Thứ tự các từ: ra đi – đi ra, cơm bữa – bữa cơm

Tập văn bản$\sim$ tập các chuỗi

Giới thiệu

- Mỗi tài liệu văn bản D được biểu diễn bằng *chuỗi (string) từ*
  - Toàn văn
  - Tiêu đề
  - Tóm tắt
- CSDL văn bản: tập hợp các chuỗi được lập chỉ mục hợp lý
- Tìm kiếm: tìm các văn bản trong CSDL có chứa các từ trong văn bản truy vấn
  - Bài toán khớp xâu (string-matching, substring-finding)

Ví dụ

DocumentID	String
d_1	Jose Orojuelo's Operations in Bosnia
d_2	The Medellin Cartel's Financial Organization
d_3	The Cali Cartel's Distribution Network
d_4	Banking Operation and Money Laundering
d_5	Profile of Hector Gomez
d_6	Connection between Terrorism and Asian Dope Operations
d_7	Hector Gomez: How He Gave Agents the Slip in Cali
d_8	Drugs, and Videotape
d_9	The Iranian Connection
d_{10}	Boating and Drugs: Slips Owned by the Cali Cartel

Vấn đề khi khớp xâu

Vấn đề từ đồng nghĩa (Synonymy): từ truy vấn không xuất hiện trong tài liệu nhưng D liên quan đến chủ đề truy vấn

- Với chủ đề "*money laundering*": tìm được d_4 nhưng không tìm được d_2
- Với từ "'*drugs*"
  - Tìm được d_8, $d_1$0
  - Không tìm được d_6 (có từ đồng nghĩa dope)
  - Bị bỏ qua d_2, d_3 (đề cập đến tập đoàn ma túy drug cartel)

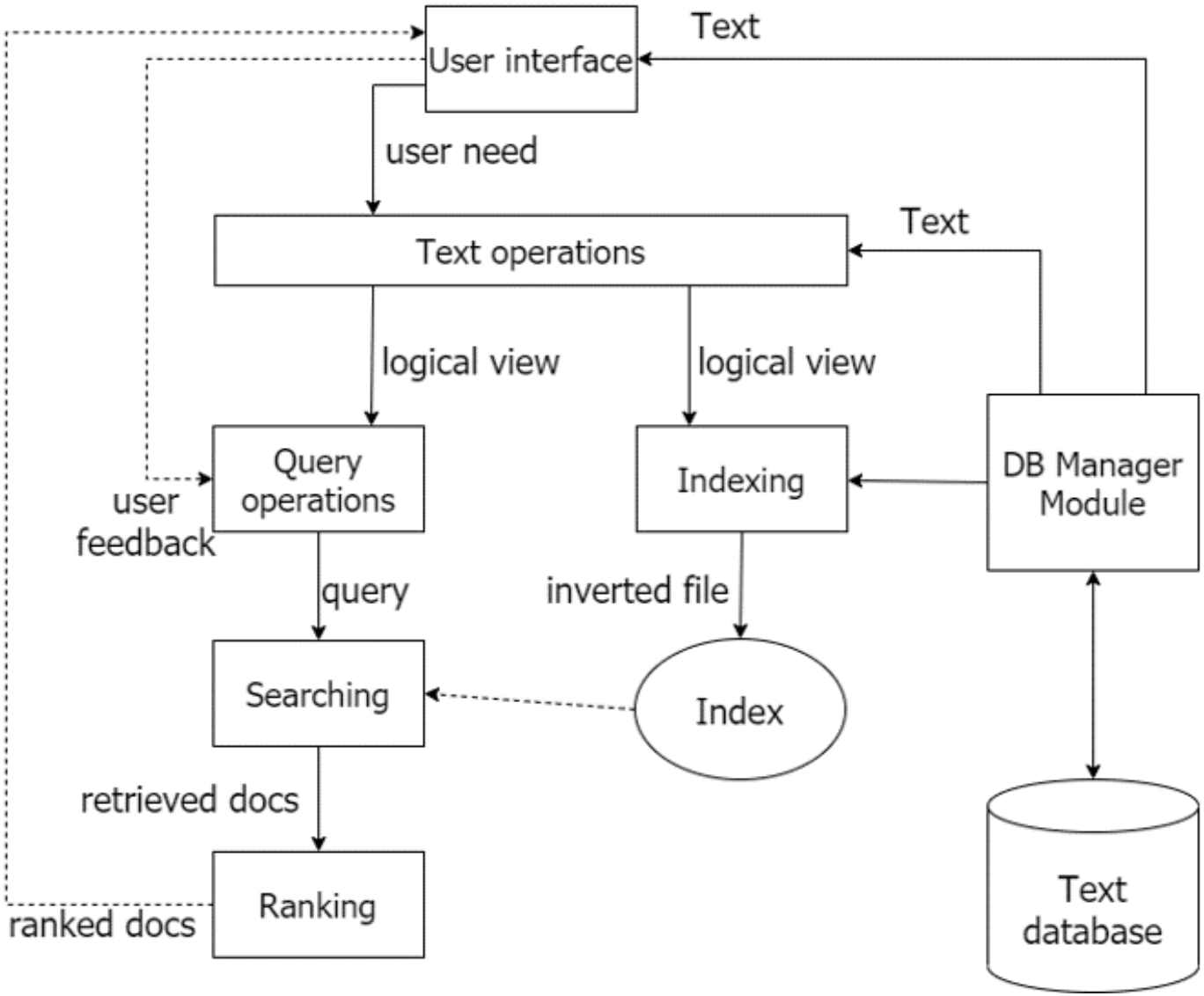
Vấn đề khi khớp xâu

Vấn đề đa nghĩa (Polysemy): một từ có thể có nhiều nghĩa tùy theo bối cảnh

- Từ *bank*: financial institution, river bank, bank on,...
- Nếu truy vấn tài liệu liên quan đến tài chính: bỏ qua các tài liệu có tựa "Otters on the Banks of the Colorado River"

Xử lý trật tự từ

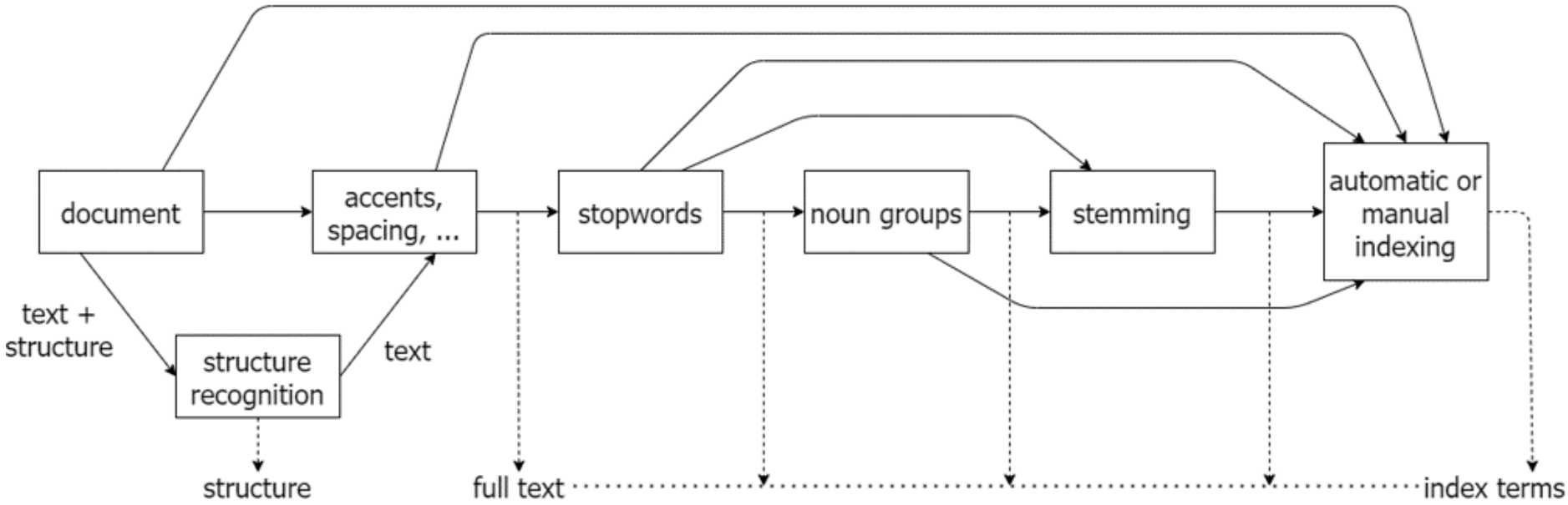
Kiến trúc tổng thể hệ thống IR



Biểu diễn văn bản

- Mỗi tài liệu được biểu diễn bởi một *tập các từ (bag of words)*
  - Ví dụ: “Game of Thrones”: {“Game”, “of”, “Thrones”}
  - Mỗi từ được xem là một chiều trong không gian từ điển
  - Số chiều = kích thước của từ điển
- Một số kỹ thuật xử lý
  - Stop list
  - Stemming
  - Frequency table

Biểu diễn văn bản



Lược đồ logic của một document

Biểu diễn văn bản

Stop list: các từ không giúp phân biệt các tài liệu trong 1 tập các tài liệu được xem xét

- Chung: *the*, *a*, *of*, *at*, *are*...
- Tùy vào bản chất của tập dữ liệu
  - Ví dụ: báo cáo kỹ thuật của đề tài liên quan đến *computer science*
    "computer" thuộc stop list
  - Ví dụ: tài liệu về trồng trọt
    "computer" không thuộc stop list

Biểu diễn văn bản

Stemming: nhóm các biến thể của một từ gốc thành 1 nhóm, biểu diễn bởi 1 từ đại diện

- retrieved, retrieval, retrieving, retrieve $\rightarrow$ retriev
- drug, drugs, drugged $\rightarrow$ drug

Thesaurus: nhóm các từ gần nghĩa $\rightarrow$ sử dụng từ điển đồng nghĩa hoặc có liên quan

- learning, school work, study, reading $\rightarrow$ study

Biểu diễn văn bản

Frequency table (bảng tần suất): hỗ trợ xác định mức độ quan trọng khác nhau của các từ trong văn bản khi thực hiện

- $\mathcal{D}$: tập N văn bản
- $\mathcal{T}$: tập M từ trong các tài liệu thuộc $\mathcal{D}$
- Frequency table: $M \times N$
- tf(i, j) (term frequency): số lần xuất hiện các từ t_i trong văn bản d_j

Biểu diễn văn bản

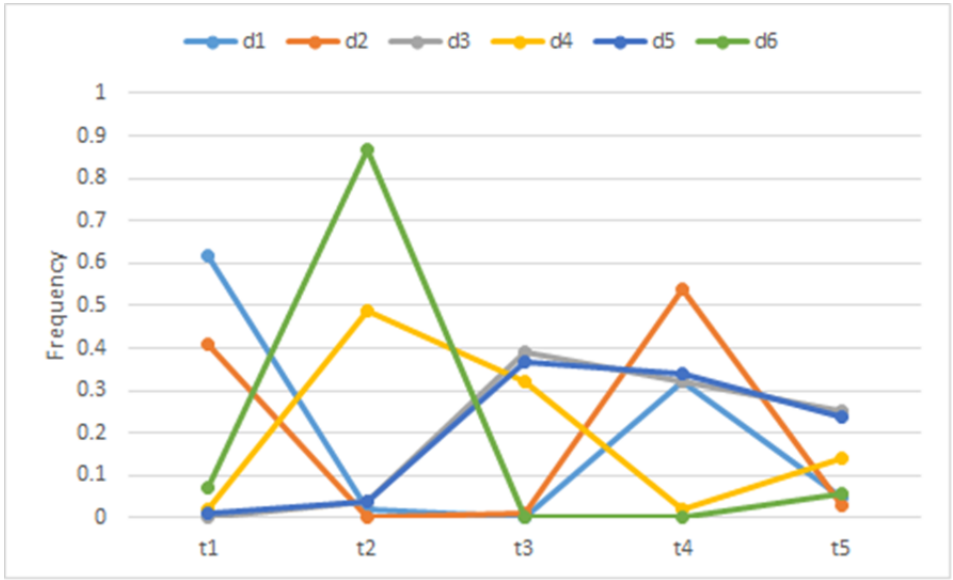
Term/doc	d_8	d_9	d_{10}
drug	1	0	1
videotape	1	0	0
iran	0	1	0
connection	0	1	0
boat	0	0	1
slip	0	0	1
own	0	0	1
cali	0	0	1
cartel	0	0	1

Biểu diễn văn bản

Term/doc	d_1	d_2	d_3	d_4	d_5	d_6
t_1	615	390	10	10	18	65
t_2	15	4	76	217	91	816
t_3	2	8	815	142	765	1
t_4	312	511	677	11	711	2
t_5	45	33	516	64	491	59

- Mỗi văn bản d_j được biểu diễn bởi 1 vector chỉ tần suất xuất hiện của các từ trong văn bản đó: $(\text{tf}_{1,j}, \text{tf}_{2,j}, \ldots, \text{tf}_{M,j})$
- Thường được chuẩn hóa về $[0, 1]$: để tính đến ảnh hưởng của độ dài văn bản

Biểu diễn văn bản



Term/doc	d_1	d_2	d_3	d_4	d_5	d_6
t_1	0.62	0.41	0.00	0.02	0.01	0.07
t_2	0.02	0.00	0.04	0.49	0.04	0.87
t_3	0.00	0.01	0.39	0.32	0.37	0.00
t_4	0.32	0.54	0.32	0.02	0.34	0.00
t_5	0.05	0.03	0.25	0.14	0.24	0.06

- (d_1, d_2) và (d_3, d_5): tương đồng
- (d_3, d_6): khác biệt

Biểu diễn văn bản

- **idf** (inverse document frequency): xác định độ quan trọng của mỗi từ trong tập dữ liệu văn bản đang xem xét

$$\text{idf}_i = \log\left(\frac{N}{\text{df}_i}\right)$$

  - N: tổng số văn bản trong tập dữ liệu
  - df_i: số văn bản có chứa từ t_i

- Trọng số tf . idf của từ t_i trong văn bản d_j là:

$$w_{i,j} = \text{tf}_{i,j} \times \text{idf}_{i,j}$$

- Mỗi văn bản d_j được biểu diễn bởi 1 vector tf . idf

$$(w_{1,j}, w_{2,j}, \ldots, w_{M,j})$$

Ví dụ

- D_1: "John has some cats"
- D_2: "Cats eat fish"
- D_3: "I eat a big fish"

Bài tập

Cho các document

- Doc1: Ben studies about computers in Computer Lab.
- Doc2: Steve teaches at Brown University.
- Doc3: Data Scientists works on large datasets.
- Doc4: Data Science is a subject taught by the head of Computer Lab.
- Query: Data Scientist

Đánh chỉ mục (Indexing)

- Nếu dùng flat-files $\rightarrow$ không hiệu quả
- Inverted files:
  - Hiệu quả
  - Dễ cài đặt
  - Thông dụng trong hệ thống tìm kiếm văn bản
- Signature files (PAT trees, graphs)

File đảo – inverted file

Document

DocID	Postings_list
DocID1	Term1, Term2, Term4
DocID2	Term2, Term3, Term4
DocID3	Term1, Term3, Term4

Term

Term	Postings_list
Term1	DocID1, DocID3
Term2	DocID1, DocID2
Term3	DocID2, DocID3
Term4	DocID1, DocID2, DocID3

- **Term**
  Lưu các từ/khái niệm/từ khóa
- **Postings_list**
  Chỉ ra văn bản [, vị trí trong văn bản] mà term xuất hiện

File đảo – inverted file

Mỗi bản ghi của bảng term

- Có thể chứa thông tin chi tiết vị trí của mỗi xuất hiện trong từng tài liệu
  - term i: Doc id, paragraph number, sentence number, word number
  - Information: R99, 10, 8, 3; R155, 15, 3, 6; R166, 2, 3, 1
  - Retrieval: R77, 9, 7, 2; R99, 10, 8, 4; R166, 10, 2, 5
- Có thể có thông tin về tần suất xuất hiện của term trong tài liệu
  - Term1: R1, 0.3; R3, 0.5; R6, 0.8; R7, 0.2; R11, 1
  - Term2: R2, 0.7; R3, 0.6; R7, 0.5; R9, 0.5
  - Term3: R1, 0.8; R2, 0.4; R9, 0.7

Tìm kiếm (retrieving textual documents)

- Truy vấn hiệu quả các tài liệu đã được đánh chỉ mục
  - Câu truy vấn Q được biểu diễn tương tự các tài liệu
  - So sánh Q và các tài liệu trong CSDL

  $\rightarrow$ Xác định khoảng cách giữa Q và các d_j
- 03 loại phương pháp truy vấn
  - Boolean Models: Fuzzy, Extended Boolean Models
  - Vector Models: Generalized vector, Latent Semantic Index, Neural Networks...
  - Probabilistic Models: Inference Network, Belief Network...

Boolean Model

- Mỗi văn bản trong CSDL: tập các từ khóa
- Câu truy vấn Q:
  - Biểu diễn bằng các từ khóa
  - Các phép toán logic: AND, OR, NOT
  - Ví dụ: information AND retrieval
- Thực hiện dễ dàng với Inverted File thông qua các phép hợp, giao, trừ

Vector Model

- Giả sử các văn bản và truy vấn đều được biểu diễn bởi 1 tập cố định M khái niệm/từ (term) có trọng số
- Mỗi văn bản D_j, truy vấn Q_i được biểu diễn bàng vector

$$\overrightarrow{D_j} = [w_{1,j}, w_{2,j}, \ldots, w_{M,j}]$$
$$\overrightarrow{Q_i} = [w_{1,i}, w_{2,i}, \ldots, w_{M,i}]$$

Với:

- $w_{k,j}$, $w_{k,i}$: trọng số của từ k trong D_j và Q_i
- $w_{k,l}$: $\{0, 1\}$, tf . idf, tf,... $\rightarrow$ thường nhận trọng số tf . idf

Vector Model

Khoảng cách D_j và Q_i

- Khoảng cách khái niệm

$$d(Q_i, D_j) = \sqrt{\sum_{k=1}^{M} (w_{k,i} - w_{k,j})^2}$$

- Khoảng cách cosine: $1 - S(Q_i, D_j)$

$$S(Q_i, D_j) = \frac{\overrightarrow{Q_i}.\overrightarrow{D_j}}{\|\overrightarrow{Q_i}|.\|\overrightarrow{D_j}\|} = \frac{\sum_{k=1}^{M} w_{k,i}.w_{k,j}}{\sqrt{\sum_{k=1}^{M} w_{k,i}^2} \cdot \sqrt{\sum_{k=1}^{M} w_{k,j}^2}}$$

Vector Model

Kết quả thu được sẽ được sắp xếp (ranking) theo thứ tự giảm dần của độ tương tự

- Ví dụ:

$$D_1 = [0.2, 0.1, 0.4, 0.5],\ D_2 = [0.5, 0.6, 0.3, 0],$$
$$D_3 = [0.4, 0.5, 0.8, 0.3],\ D_4 = [0.1, 0, 0.7, 0.8],$$
$$Q = [0.5, 0.5, 0, 0]$$

$$S(Q, D_1) = 0.31,\ S(Q, D_2) = 0.93,$$
$$S(Q, D_3) = 0.66,\ S(Q, D_4) = 0.07$$

$\rightarrow$ Kết quả: ?

Vector Model

Ưu điểm:

- Cho phép tìm kiếm gần đúng (partial matching)
- Đo được mức độ giống nhau giữa văn bản và truy vấn
- Đơn giản
- Thích hợp với các văn bản ngắn

Hạn chế:

- Coi các term không có liên quan với nhau
- Chưa tính đến mối liên hệ không gian giữa các từ
- Độ phức tạp tìm kiếm: $O(M \times N)$ lớn khi M, N lớn

LSI

- Mô hình **Latent Semantic Indexing**: mô hình chỉ mục ngữ nghĩa tiềm năng
- Một biến thể của Vector Models
- **Ý tưởng**
  - Văn bản thường liên quan đến khái niệm (concept) hơn là liên quan trực tiếp đến các từ dùng trong văn bản

    Đồi, sườn dốc, núi, hang động, đá $\rightarrow$ thuộc 1 concept

  $\rightarrow$ Tìm kiếm dựa trên khái niệm
  - Biểu diễn văn bản với K chiều với $K \ll M$

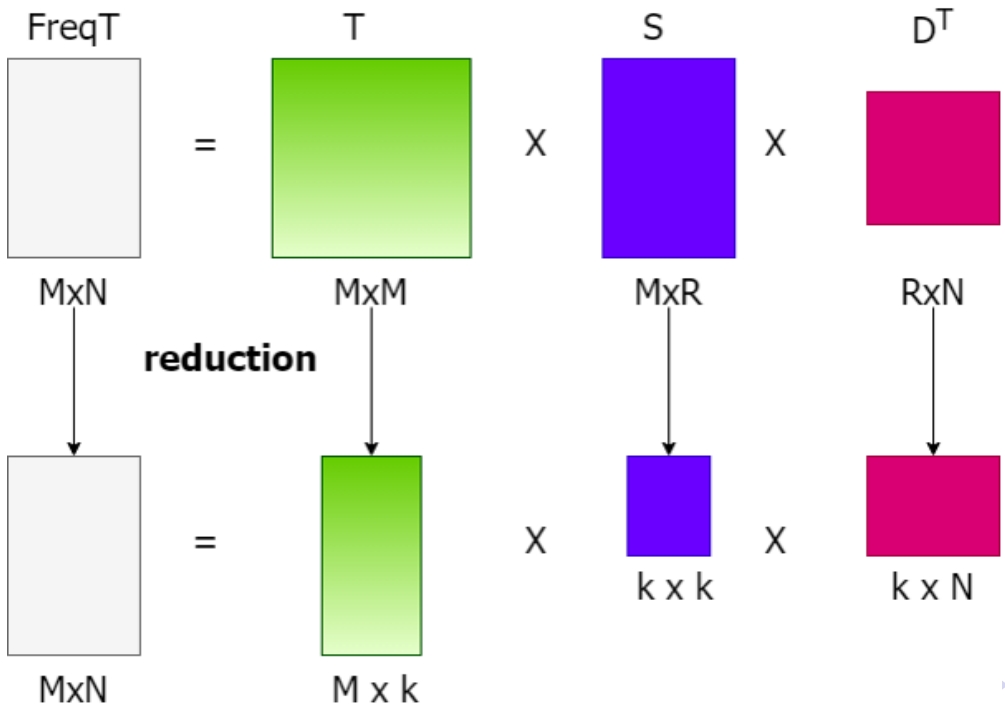
LSI

Kỹ thuật giảm số chiều: **SVD (Singular Valued Decomposition)**

$$T^T T = I_R,\ D^T D = I_R$$

$$S(i,j) = 0\ ,\ i \neq j$$

$$S(1,1) \geq S(2,2) \geq \cdots \geq S(R,R)$$



LSI

- Giả sử FreqT có SVD

$$\begin{pmatrix} a_1^1 & a_2^1 & a_3^1 & a_4^1 & a_5^1 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 & a_5^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_1^M & a_2^M & a_3^M & a_4^M & a_5^M \end{pmatrix} \begin{pmatrix} 20 & 0 & 0 & 0 & 0 \\ 0 & 16 & 0 & 0 & 0 \\ 0 & 0 & 12 & 0 & 0 \\ 0 & 0 & 0 & 0.08 & 0 \\ 0 & 0 & 0 & 0 & 0.004 \end{pmatrix} \begin{pmatrix} b_1^1 & b_2^1 & b_3^1 & \cdots & b_N^1 \\ b_1^2 & b_2^2 & b_3^2 & \cdots & b_N^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_1^5 & b_2^5 & b_3^5 & \cdots & b_N^5 \end{pmatrix}$$

- Nếu đặt ngưỡng là 3, thì kết quả là

$$\begin{pmatrix} a_1^1 & a_2^1 & a_3^1 \\ a_1^2 & a_2^2 & a_3^2 \\ \vdots & \vdots & \vdots \\ a_1^M & a_2^M & a_3^M \end{pmatrix} \begin{pmatrix} 20 & 0 & 0 \\ 0 & 16 & 0 \\ 0 & 0 & 12 \end{pmatrix} \begin{pmatrix} b_1^1 & b_2^1 & b_3^1 & \cdots & b_N^1 \\ b_1^2 & b_2^2 & b_3^2 & \cdots & b_N^2 \\ b_1^3 & b_2^3 & b_3^3 & \cdots & b_N^3 \end{pmatrix}$$

LSI

- Kích thước của bảng tần suất ban đầu là $M \times N$
  - Kích thước của có thể lên đến $M = 1$ triệu và $N =$10,000 ngay cả với CSDL tài liệu nhỏ
- Sau SVD: kích thước của các ma trận đơn với $k = 200$
  - Ma trận T: $M \times k \sim 1$ triệu $\times 200 = 200$ triệu đầu vào
  - Ma trận S: $k \times k \sim 200 \times 200 = 40,000$ đầu vào (chỉ 200 giá trị cần lưu trữ; toàn bộ các đầu vào còn lại có giá trị 0)
  - Ma trận D: $k \times N \sim 200 \times 10,000 = 2$ triệu đầu vào

  $\rightarrow$ Tổng số dữ liệu cần lưu trữ xấp xỉ 202 triệu thay vì $10,000$ triệu

LSI

Các bước của LSI

- Tạo ma trận: tính bảng tần suất FreqT ($M \times N$)
- Áp dụng SVD để phân rã FreqT thành T, S, D
- Xác định vector biểu diễn cho mỗi văn bản d(vec(d)): các phần tử trong FreqT tương ứng với dòng không bị loại bỏ trong ma trận S
- Tạo chỉ số: lưu lại các vec(d) của CSDL (sử dụng cấu trúc dữ liệu đa chiều, vd: R-tree, k-D tree, TV-tree)

LSI - Truy vấn

- Giả sử sau khi loại bỏ các thành phần ít quan trọng, SVD cho FreqT được biểu diễn bởi T^*, S^*, D^{*T}
- Sự tương tự giữa 2 văn bản d_i, d_j trong CSDL

$$\sum_{z=1}^{k} D^{*T}[i,z] \times D^{*T}[j,z]$$

LSI - Truy vấn

- Tìm kiếm p văn bản phù hợp đầu tiên cho truy vấn Q
  - Xem Q như 1 tài liệu để tính vector biểu diễn cho Q: vec_Q
  - Điểm khác biệt: chỉ xét trên k khái niệm chứ (không phải M)
- p tài liệu $d_{\alpha(1)}, d_{\alpha(2)}, \ldots, d_{\alpha(p)}$ phù hợp với Q:

  $\forall i, j$: $0 \leq i \leq j \leq p$

  $\text{similarity}(\text{vec}_Q, d_{\alpha(i)}) \geq \text{similarity}(\text{vec}_Q, d_{\alpha(j)})$

  $\neg\exists z \notin \{\alpha(1), \alpha(2), \ldots, \alpha(p)\}$

  $\text{similarity}(\text{vec}_Q, d_z) \geq \text{similarity}(\text{vec}_Q, d_{\alpha(p)})$

LSI - Truy vấn

$\text{FreqT} = T^* \times S^* \times {D^*}^T \Rightarrow D = \text{FreqT}^T \times T^* \times {S^*}^{-1}$

- Xác định vector vec_Q biểu diễn cho Q từ T^*, S^*, ${D^*}^T$
  - Vector tần số cho truy vấn Q trên M từ $f_Q : M \times 1$
    $\text{vec}_Q = f_Q^T \times T^* \times {S^*}^-1$
- Xác định độ tương tự giữa vector vec_Q và các vector tương ứng với các cột trong ${D^*}^T$

Probabilistic Model

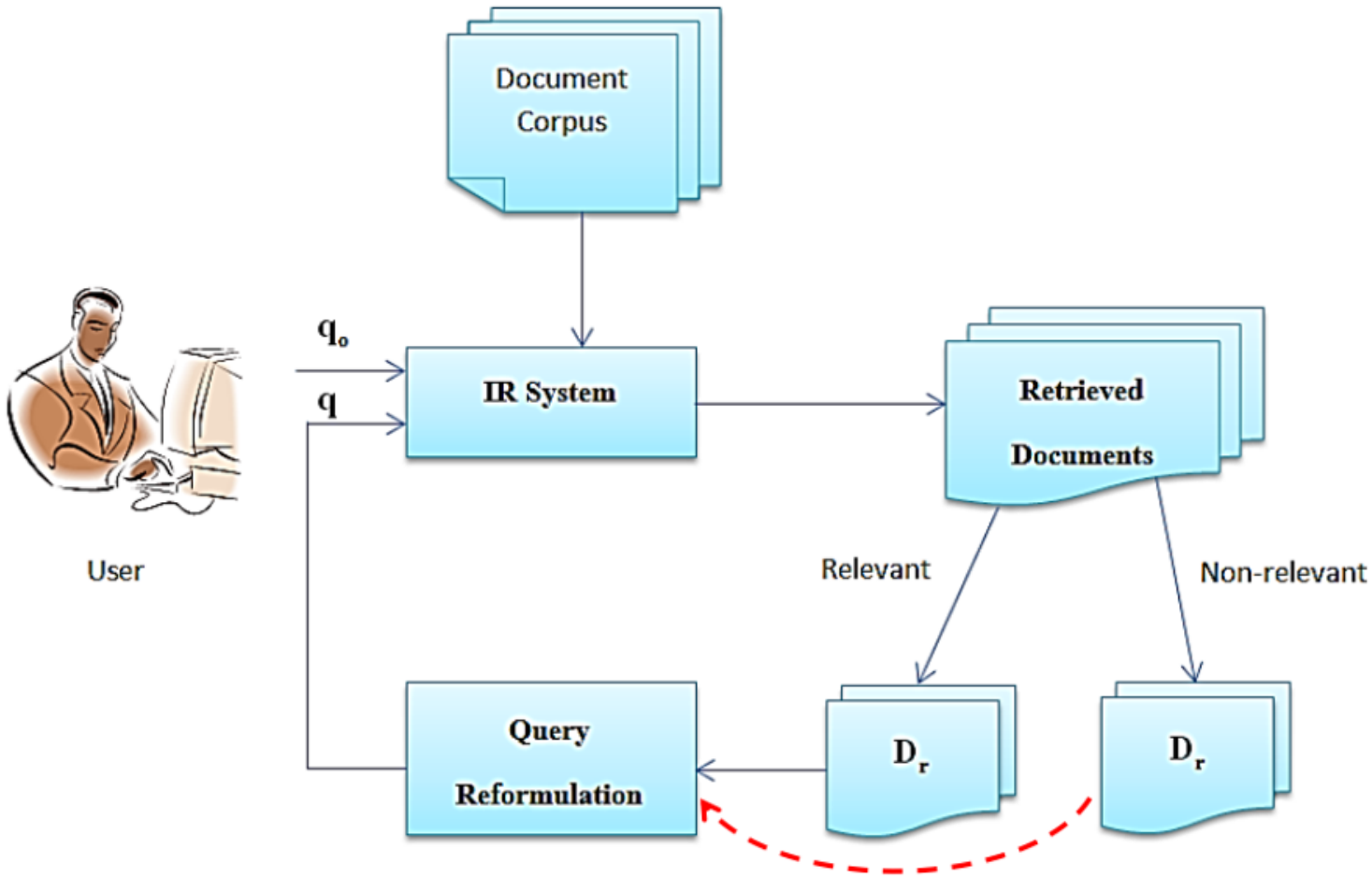
- Dựa trên lý thuyết xác suất gồm các tham số
  - $P(\text{rel}|d_j)$: xác suất 1 văn bản liên quan (relevant) tới truy vấn Q
  - $P(\text{nonrel}|d_j)$: xác suất 1 văn bản không liên quan (non-relevant, irrelevant) tới truy vấn Q
  - Chi phí tương ứng khi trả về tài liệu non-relevant
  - Chi phí tương ứng khi không lấy tài liệu relevant
- Không hiệu quả trong truy vấn do khó xác định $P(\text{rel}|d_j)$, $P(\text{nonrel}|d_j)$

Phản hồi có liên quan (Relevance Feedback)

- RF – Relevance Feedback
  - Cho phép người sử dụng đánh dấu các câu trả lời đúng (relevant) và chưa đúng (irrelevant)

  $\rightarrow$ Cải tiến hiệu năng của hệ thống
  - Thích hợp với Vector Model
- 2 hướng tiếp cận
  - Query Modification
  - Document Modification

Phản hồi có liên quan (Relevance Feedback)



Phản hồi có liên quan (Relevance Feedback)

- Thay đổi biểu diễn câu truy vấn (Query Modification)

$$Q^{i+1} = Q^i + \alpha \sum_{D^i \in \text{rel}} D^i - \beta \sum_{D^j \in \text{rel}} D^j$$

  - Thông dụng
  - Cải tiến hiệu năng của hệ thống
  - Chỉ cho 1 người sử dụng, không tận dụng được cho người dùng khác
- Thay đổi biểu diễn văn bản trong CSDL (Document modification)
  - Có thể tận dụng cho người dùng khác nhau
  - Có thể giảm hiệu quả do các truy vấn sau khác câu truy vấn đã thay đổi văn bản

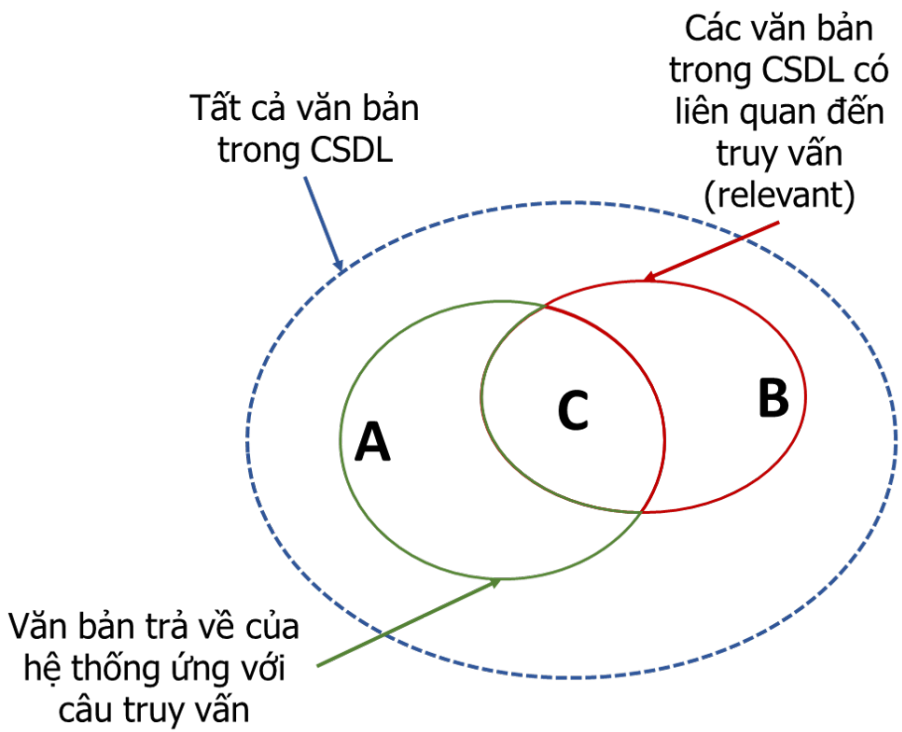
Các độ đo thông dụng

- Độ chính xác (Precision)

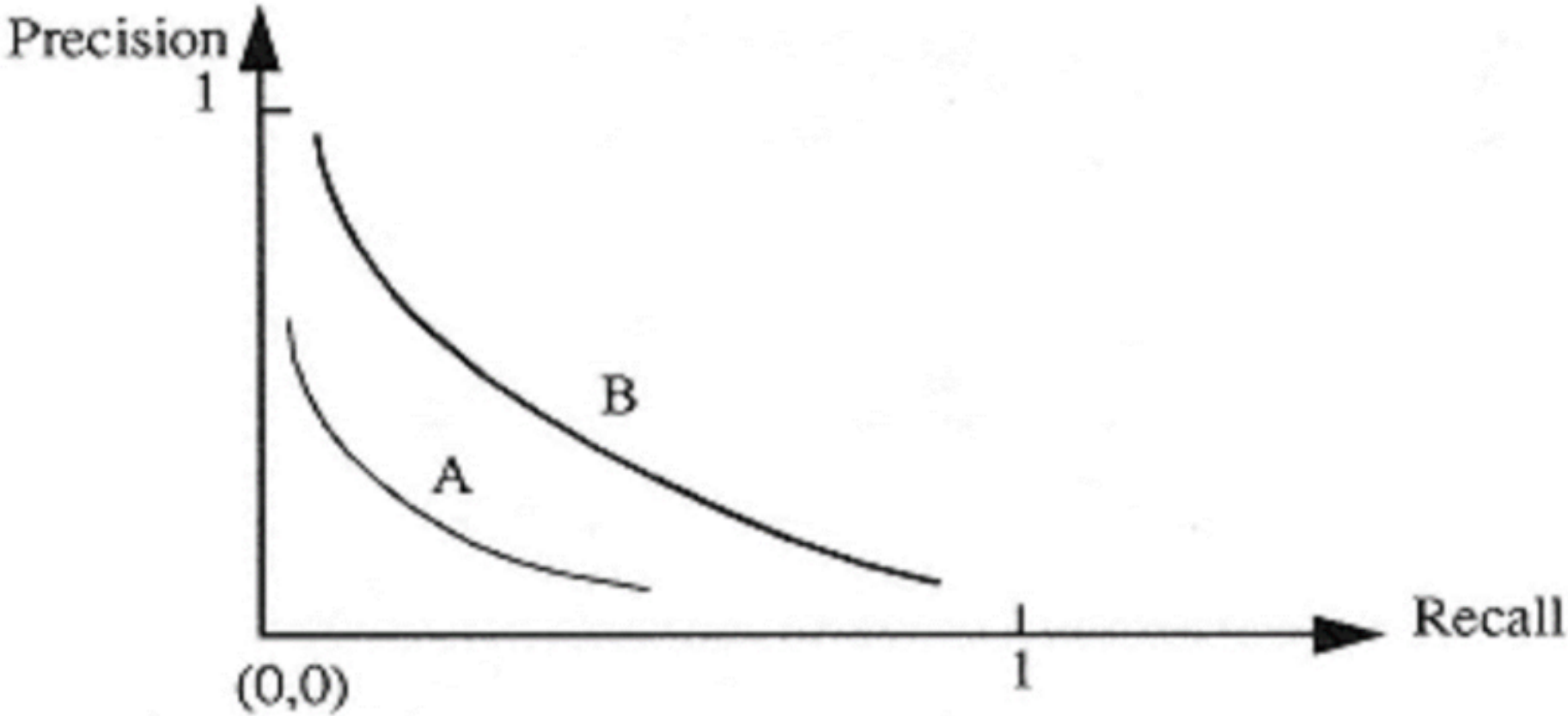
$$\text{Prec} = \frac{C}{A+C}$$

- Độ nhạy (Recall)

$$\text{Recall} = \frac{C}{B+C}$$



Các độ đo thông dụng



Đường cong precision - recall

Các độ đo thông dụng

- P@n, R@n: độ chính xác tính trên n kết quả trả về gần nhất
- F-score:

$$F = \frac{2 \times \text{Prec} \times \text{Recall}}{\text{Prec} + \text{Recall}}$$

- Average precision
- Mean average precision
- ...