ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÀI 4: TÌM KIẾM THÔNG TIN

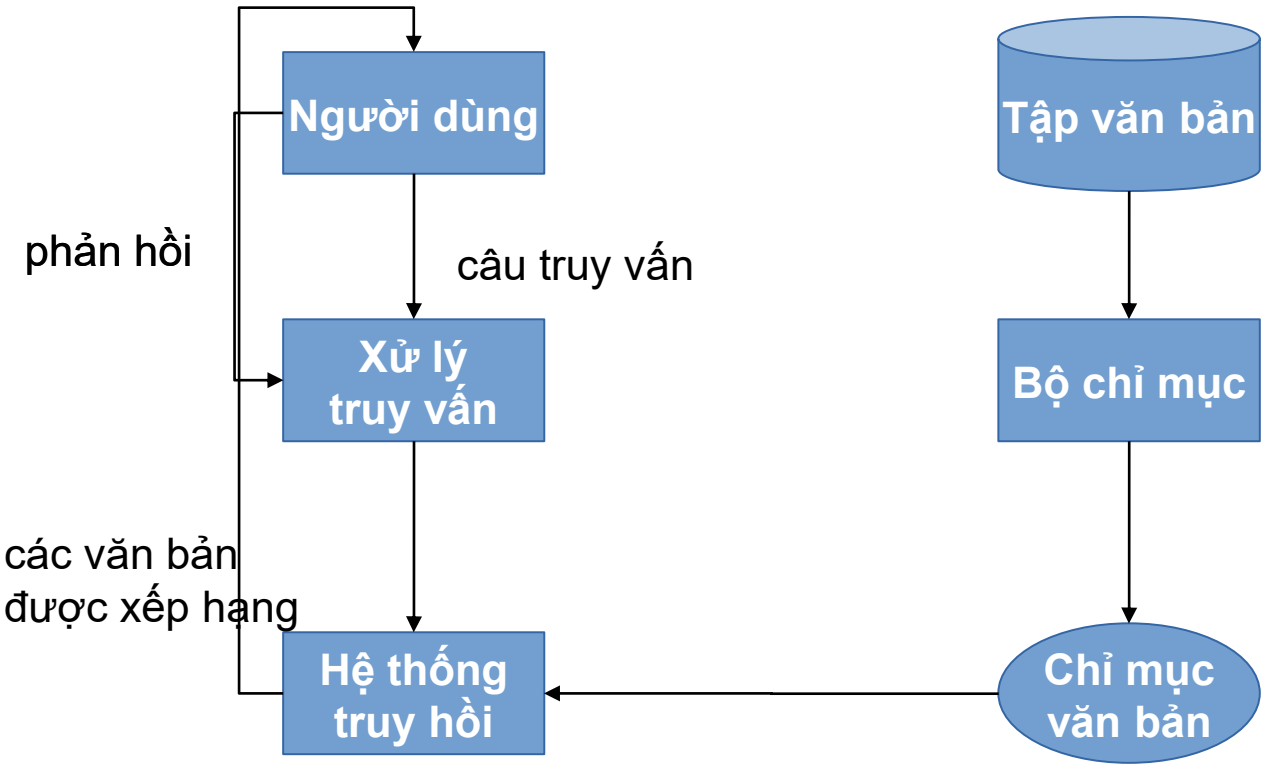
Nội dung

1. Các khái niệm cơ bản
2. Các mô hình tìm kiếm thông tin
3. Phản hồi liên quan
4. Các phương pháp đánh giá
5. Tiền xử lý văn bản
6. Chỉ mục ngược
7. Đánh chỉ mục ngữ nghĩa ẩn
8. Tìm kiếm web
9. Siêu tìm kiếm
10. Web spam

1. Các khái niệm cơ bản

- Tìm kiếm thông tin giúp người dùng tìm kiếm thông tin phù hợp với nhu cầu của họ
- Tìm kiếm thông tin nghiên cứu việc thu thập, tổ chức, lưu trữ, truy hồi, và phân phối thông tin
- Hệ thống tìm kiếm thông tin truyền thống coi văn bản là đơn vị cơ bản
- Người dùng với nhu cầu thông tin đưa ra một câu truy vấn tới hệ thống truy hồi thông qua các thao tác truy vấn. Thành phần truy hồi sử dụng chỉ mục văn bản để lấy các văn bản chứa các từ khóa trong câu truy vấn (các văn bản này có nhiều khả năng phù hợp với câu truy vấn), tính toán điểm phù hợp, và xếp hạng các văn bản theo điểm. Các văn bản được xếp hạng được trả về cho người dùng. Tập văn bản (CSDL văn bản) được đánh chỉ mục để tăng hiệu quả truy vấn

Các khái niệm cơ bản (tiếp)



Các khái niệm cơ bản (tiếp)

- Các loại câu truy vấn
  1. Truy vấn từ khóa: Câu truy vấn gồm một danh sách các từ khóa. Các văn bản trả về có thể chứa một, một vài, hoặc tất cả các từ khóa. Trật tự của các từ khóa có thể được bảo đảm. Vd: *information retrieval*
  2. Truy vấn nhị phân: Các từ khóa được kết hợp bởi các thao tác nhị phân AND, OR và NOT. Vd: *information* OR *retrieval*
  3. Truy vấn cụm: Gồm một chuỗi các từ hình thành nên một cụm. Văn bản trả về phải chứa cụm truy vấn. Vd "*information retrieval systems*"
  4. Truy vấn lân cận: Xếp hạng các văn bản dựa trên độ lân cận của các từ khóa trong câu truy vấn
  5. Truy vấn văn bản: Tìm kiếm các văn bản tương tự văn bản truy vấn
  6. Hỏi – đáp: Câu truy vấn dưới dạng câu hỏi tự nhiên, hệ thống trả về câu trả lời. (vd câu hỏi định nghĩa)

Các khái niệm cơ bản (tiếp)

- Xử lý truy vấn bao gồm các thao tác tiền xử lý như loại bỏ từ dừng (các từ xuất hiện nhiều và có ít ý nghĩa như ‘it’, ‘from’, ‘are’); các thao tác biến đổi câu truy vấn thành các truy vấn thực thi được; sử dụng phản hồi của người dùng để mở rộng và tinh chỉnh câu truy vấn
- Bộ đánh chỉ mục chuyển các tài liệu thô thành các cấu trúc dữ liệu đặc biệt phục vụ cho việc truy vấn gọi là chỉ mục văn bản; kĩ thuật chỉ mục ngược rất dễ dàng áp dụng và hiệu quả cho việc truy vấn
- Hệ thống truy hồi tính điểm liên quan của mỗi văn bản đối với câu truy vấn. Các văn bản được xếp hạng theo điểm liên quan và và trả về cho người dùng. Lưu ý: Chỉ các văn bản có chứa ít nhất một từ khóa truy vấn mới được tính điểm.

2. Các mô hình tìm kiếm thông tin

- Mô hình tìm kiếm thông tin thực hiện biểu diễn và tính điểm liên quan của văn bản và câu truy vấn
- Văn bản và câu truy vấn được biểu diễn dưới dạng túi từ, không quan tâm đến vị trí và trật tự xuất hiện
- D là tập các văn bản
- $V = \{t_1, t_2, \ldots, t_{|V|}\}$ là tập hợp các từ trong D trong đó t_i là một từ xuất hiện trong D, V được gọi là từ vựng với $|V|$ là kích thước từ vựng
- Mỗi từ t_i trong văn bản $\boldsymbol{d}_j \in D$ có trọng số w_{ij} thể hiện mức độ quan trọng của t_i trong $\boldsymbol{d}_j$; $\boldsymbol{d}_j = (w_{1j}, w_{2j}, \ldots, w_{|V|j})$

2.1 Mô hình Boolean

- Biểu diễn văn bản: Văn bản và câu truy vấn là tập hợp các từ khóa

$$w_{ij} = \begin{cases} 1 \text{ nếu } t_i \text{ xuất hiện trong } \boldsymbol{d}_j \\ 0 \text{ nếu ngược lại} \end{cases}$$

- Câu truy vấn Boolean: Các từ khóa được kết hợp bởi các toán tử logic AND, OR và NOT. Vd:
  - *((x AND y) AND (NOT z))* văn bản trả về chứa cả *x* và *y* nhưng không chứa *z*
  - *(x OR y)* văn bản chứa ít nhất *x* hoặc *y*
- Truy hồi: Tất cả các văn bản thỏa mãn câu truy vấn được trả về và không được xếp hạng

2.2 Mô hình không gian véc-tơ

- Biểu diễn văn bản:
  - Tf: Trọng số của từ t_i trong văn bản $\boldsymbol{d}_j$ là số lần t_i xuất hiện trong $\boldsymbol{d}_j$ (f_{ij}). Nhược điểm: không phân biệt được các từ xuất hiện trong nhiều văn bản

$$tf_{ij} = \frac{f_{ij}}{max(f_{1j}, f_{2j}, \ldots, f_{|V|j})}$$

  - Tf-idf: Các từ xuất hiện trong nhiều văn bản có trọng số thấp

$$idf_i = log \frac{N}{df_i} \qquad w_{ij} = tf_{ij} \times idf_i$$

N: tổng số văn bản; df_i là số văn bản trong đó t_i xuất hiện

- Câu truy vấn (Salton and Buckley)

$$w_{iq} = \left(0.5 + \frac{0.5 f_{ij}}{max(f_{1q}, f_{2q}, \ldots, f_{|V|q})} \right) \times log \frac{N}{df_i}$$

Mô hình không gian véc-tơ (tiếp)

- Xếp hạng: Dựa trên độ tương đồng của văn bản $\boldsymbol{d}_j$ và câu truy vấn q
- Độ tương đồng cosine là độ đo phổ biến nhất
- Okapi hiệu quả hơn đ/v các câu truy vấn ngắn

  dl_j là độ dài (bytes) của $\boldsymbol{d}_j$

  $avdl$ là độ dài trung bình

$$cosine(\mathbf{d}_j,\mathbf{q}) = \frac{\langle \mathbf{d}_j \bullet \mathbf{q} \rangle}{\| \mathbf{d}_j \| \times \| \mathbf{q} \|} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}}.$$

$$sim(\mathbf{d}_j,\mathbf{q}) = \langle \mathbf{d}_j \bullet \mathbf{q} \rangle.$$

$$okapi(d_j,q) = \sum_{t_i \in q, d_j} \ln \frac{N - df_i + 0.5}{df_i + 0.5} \times \frac{(k_1+1) f_{ij}}{k_1(1 - b + b\frac{dl_j}{avdl}) + f_{ij}} \times \frac{(k_2+1) f_{iq}}{k_2 + f_{iq}},$$

$$pnw(d_j,q) = \sum_{t_i \in q, d_j} \frac{1 + \ln(1 + \ln(f_{ij}))}{(1-s) + s\frac{dl_j}{avdl}} \times f_{iq} \times \ln \frac{N+1}{df_i},$$

2.3 Mô hình ngôn ngữ thống kê

- Xếp hạng văn bản dựa trên khả năng sinh ra câu truy vấn của các mô hình ngôn ngữ của mỗi văn bản
- Câu truy vấn q là một chuỗi từ $q=q_1q_2...q_m$, D là tập văn bản $D = \{d_1,d_2,...,d_N\}$. Ta cần ước lượng khả năng sinh ra câu truy vấn q từ mô hình ngôn ngữ xác suất của mỗi văn bản d_j: $Pr(q|d_j)$

$$Pr(d_j|q) = \frac{Pr(q|d_j)Pr(d_j)}{Pr(q)}$$

Mô hình ngôn ngữ thống kê (tiếp)

- Mô hình ngôn ngữ unigram coi mỗi từ được sinh ra độc lập với các từ khác trong văn bản theo một phân phối multinomial

$$\Pr(q = q_1 q_2 ... q_m \mid d_j) = \prod_{i=1}^{m} \Pr(q_i \mid d_j) = \prod_{i=1}^{|V|} \Pr(t_i \mid d_j)^{f_{iq}},$$

trong đó f_{iq} là số lần xuất hiện của t_i trong q và

$$\sum_{i=1}^{|V|} \Pr(t_i \mid d_j) = 1.$$

- $Pr(t_i|d_j)$ có thể được ước lượng như sau

$$\Pr(t_i \mid d_j) = \frac{f_{ij}}{|d_j|}.$$

trong đó $|d_j|$ là tổng số từ trong văn bản d_j

- Làm mịn: Nhằm tránh các xác suất bằng không (các từ không xuất hiện trong văn bản) (λ=1 làm mịn Laplace)

$$\Pr_{add}(t_i \mid d_j) = \frac{\lambda + f_{ij}}{\lambda |V| + |d_j|}.$$

3. Phản hồi liên quan

- Người dùng xác định một số văn bản liên quan và không liên quan từ danh sách trả về ban đầu. Hệ thống dựa trên đó bổ sung thêm từ khóa vào câu truy vấn cho lượt truy vấn tiếp theo. Quá trình có thể tiếp diễn tới khi người dùng hài lòng với kết quả truy vấn
- Hệ thống có thể phân loại tập văn bản vào hai lớp *liên quan* và *không liên quan* (dựa trên các văn bản do người dùng xác định)

3.1 P^2 Rocchio

- Câu truy vấn $\boldsymbol{q}$, tập các văn bản liên quan D_r, tập các văn bản không liên quan D_{ir}, câu truy vấn mở rộng $\boldsymbol{q}_e$:

$$\mathbf{q}_e = \alpha \mathbf{q} + \frac{\beta}{|D_r|} \sum_{\mathbf{d}_r \in D_r} \mathbf{d}_r - \frac{\gamma}{|D_{ir}|} \sum_{\mathbf{d}_{ir} \in D_{ir}} \mathbf{d}_{ir},$$

- Câu truy vấn gốc $\boldsymbol{q}$ được mở rộng bằng các từ trong tập văn bản liên quan D_r
- Tập các văn bản không liên quan D_{ir} được sử dụng để làm giảm mức độ ảnh hưởng của các từ không đặc trưng (x/h trong cả hai tập) và các từ chỉ x/h trong D_{ir}

3.2 P^2 phân loại Rocchio

- Một véc-tơ $\boldsymbol{c}_i$ được xây dựng cho mỗi lớp *i*, *liên quan* và *không liên quan* (các thành phần âm thường được gán = 0)

$$\mathbf{c}_i = \frac{\alpha}{|D_i|}\sum_{\mathbf{d}\in D_i}\frac{\mathbf{d}}{\|\mathbf{d}\|} - \frac{\beta}{|D-D_i|}\sum_{\mathbf{d}\in D-D_i}\frac{\mathbf{d}}{\|\mathbf{d}\|},$$

- Bộ phân loại so sánh văn bản $\boldsymbol{d}_t$ với mỗi véc-tơ $\boldsymbol{c}_i$ dựa trên độ tương đồng *cosine*. Văn bản $\boldsymbol{d}_t$ được gán vào lớp có độ tương đồng cao nhất

for mỗi lớp *i* **do**
    xây dựng véc-tơ $\boldsymbol{c}_i$
for mỗi văn bản $\boldsymbol{d}_t$ **do**
    $class(\boldsymbol{d}_t) = argmax_i\ cosine(\boldsymbol{d}_t, \boldsymbol{c}_i)$

Các phương pháp khác

- Học LU: Số lượng văn bản được người dùng chọn *liên quan* và *không liên quan* rất nhỏ. Các văn bản này có thể được xếp vào văn bản có nhãn, các văn bản không được người dùng chọn có thể được coi là văn bản không có nhãn và được tận dụng để cải thiện bộ phân loại
- Học PU: Trong nhiều trường hợp (vd tìm kiếm web), người dùng chỉ lựa chọn các văn bản liên quan (dựa trên tiêu đề và tóm tắt). Các văn bản này được coi như các ví dụ tích cực. Các văn bản khác được coi như không có nhãn (phản hồi ẩn).
- SVM xếp hạng: Sử dụng SVM để xếp hạng các văn bản chưa được lựa chọn dựa trên các văn bản được lựa chọn
- Mô hình ngôn ngữ

3.3 Phản hồi liên quan giả

- Trích rút các từ khóa phổ biến trong các văn bản có hạng cao nhất để bổ sung vào câu truy vấn và thực hiện truy vấn mới
- Quá trình có thể lặp lại đến khi người dùng hài lòng với kết quả truy vấn
- Người dùng không tác động vào quá trình lựa chọn các văn bản liên quan không liên quan. Giả sử các văn bản xếp hạng cao nhất là liên quan

4. Phương pháp đánh giá

- R^q: $<d_1^q, d_2^q, ..., d_N^q>$ xếp hạng của các văn bản theo thứ tự điểm liên quan
- D_q là tập các văn bản liên quan
- Độ phủ tại vị trí thứ i

$$r(i) = \frac{s_i}{|D_q|}$$

trong đó s_i là số văn bản liên quan từ d_1^q tới d_i^q

- Độ chính xác tại ví trí thứ i

$$p(i) = \frac{s_i}{i}$$

- Độ chính xác trung bình

$$p_{avg} = \frac{\sum_{d_i^q \in D_q} p(i)}{|D_q|}.$$

Phương pháp đánh giá (tiếp)

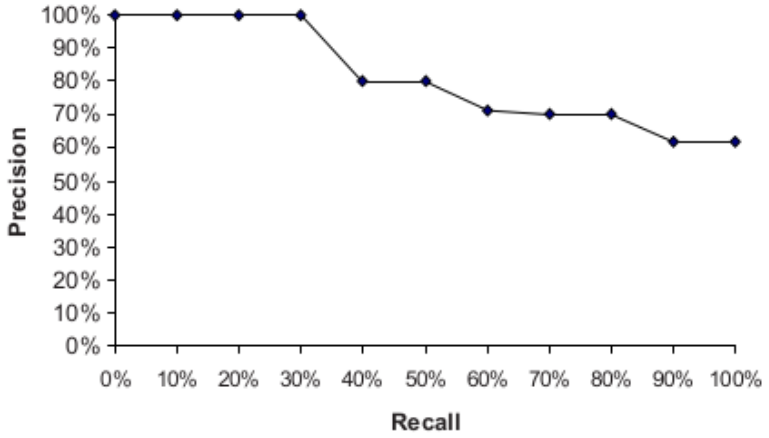
Vị trí	Liên quan	p(i)	r(i)	Vị trí	Liên quan	p(i)	r(i)
1	v	1/1	1/8	11		7/11	7/8
2	v	2/2	2/8	12		7/12	7/8
3	v	3/3	3/8	13	v	8/13	8/8
4		3/4	3/8	14		8/14	8/8
5	v	4/5	4/8	15		8/15	8/8
6		4/6	4/8	16		8/16	8/8
7	v	5/7	5/8	17		8/17	8/8
8		5/8	5/8	18		8/18	8/8
9	v	6/9	6/8	19		8/19	8/8
10	v	7/10	7/8	20		8/20	8/8

$$p_{avg} = \frac{1/1+2/2+3/3+4/5+5/7+6/9+7/10+8/13}{8} = 0.81$$

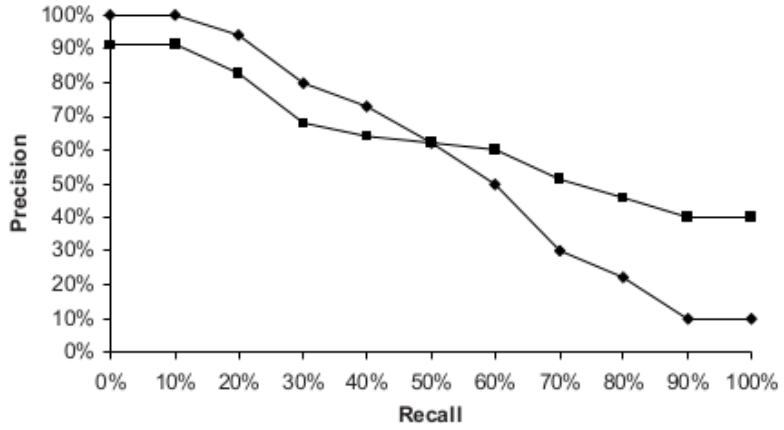
Phương pháp đánh giá (tiếp)

i	$p(r_i)$	r_i
0	1	0
1	1	0.10
2	1	0.20
3	1	0.30
4	0.80	0.40
5	0.80	0.50
6	0.71	0.60
7	0.70	0.70
8	0.70	0.80
9	0.62	0.90
10	0.62	1

$$p(r_i) = \max_{r_i \le r \le r_{10}} p(r) .$$



Đường cong độ chính xác - độ phủ



Phương pháp đánh giá (tiếp)

- Đánh giá trên nhiều câu truy vấn

$$\overline{p}(r_i) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} p_j(r_i),$$

trong đó Q là tập tất cả các câu truy vấn và $p_j(r_i)$ là độ chính xác trên câu truy vấn j tại mức phủ r_i

- Trên lý thuyết, độ chính xác và độ phủ không phụ thuộc lẫn nhau. Tuy nhiên, trong thực tế độ chính xác cao thường đi kèm độ phủ thấp và ngược lại
- Một vấn đề với độ chính xác và độ phủ là trong nhiều trường hợp không thể xác định D_q. Vd trong tìm kiếm web số lượng văn bản là quá lớn và người dùng hiếm khi xem các văn bản ngoài top 30
- Độ chính xác xếp hạng: *P(5)*, *P(10)*, *P(15)*, *P(20)*, *P(25)*, *P(30)*
- F-score

$$F(i) = \frac{2p(i)r(i)}{p(i)+r(i)}$$

5. Tiền xử lý văn bản

- Tiền xử lý văn bản:
  - Loại bỏ từ dừng
  - Stemming
  - Xử lý chữ số, dấu nối, dấu câu và viết hoa
- Tiền xử lý siêu văn bản
  - Loại bỏ thẻ HTML
  - Xác định phần nội dung chính

5.1 Loại bỏ từ dừng

- Từ dừng là các từ thường xuyên xuất hiện trong một ngôn ngữ nhưng không đáng chú ý, thường có vai trò xây dựng câu nhưng không mang nội dung
- Mạo từ, giới từ, một số đại từ là các từ dừng mặc định

a, about, an, are, as, at, be, by, for, from, how, in, is, of, on, or, that, the, these, this, to, was, what, when, where, who, will, with,...

- Từ dừng nên được loại bỏ trước khi đánh chỉ mục và lưu trữ văn bản
- Từ dừng trong câu truy vấn cũng nên được loại bỏ

5.2 Stemming

- Trong nhiều ngôn ngữ, một từ có thể có nhiều dạng cú pháp khác nhau phụ thuộc vào ngữ cảnh cụ thể (vd, trong tiếng Anh có danh từ số ít và số nhiều, động từ dạng V-ing, quá khứ, và hiện tại). Những từ này được gọi là các biến thể cú pháp của cùng một dạng gốc.
- Hiện tượng này dẫn đến độ bao phủ thấp của máy tìm kiếm khi văn bản chứa một biến thể khác với từ trong câu truy vấn (đồng thời làm tăng kích thước từ vựng)
- Stemming: chuyển từ về dạng stem tương ứng. Trong tiếng Anh, stem thu được bằng cách loại bỏ tiền tố hoặc hậu tố của từ. Vd 'computer', 'computing', và 'compute' có chung stem là 'comput'; 'walks', 'walking', và 'walker' có chung stem là 'walk'
- Thuật toán stemming của Martin Porter
- Nhược điểm: có thể làm giảm độ chính xác do trả về các văn bản không liên quan. Vd 'cop' (cảnh sát) và 'cope' (đối phó) đều có chung stem là 'cop'

5.3 Các thao tác xử lý văn bản khác

- Chữ số: Trong các hệ thống IR truyền thống, chữ số thường được loại bỏ ngoài trừ các dạng đặc biệt như thời gian, ngày tháng. Tuy nhiên, trong các máy tìm kiếm ngày nay, các chữ số được giữ lại
- Dấu nối: Dấu nối thường được xóa đi kèm theo một số ngoại lệ
  - Xóa dấu nối để lại khoảng trống. Vd 'pre-processing' → 'pre processing'
  - Xóa dấu nối 'state-of-the-art'→ 'stateoftheart'
- Dấu câu: Xử lý tương tự như dấu nối
- Viết hoa: Tất cả kí tự được chuyển về một dạng (viết hoa hoặc viết thường)

5.4 Tiền xử lý trang web

- Nhận diện cấu trúc văn bản: Văn bản HTML thường bao gồm các phần khác nhau như tiêu đề, sapo và thân bài. Phần tiêu đề tóm tắt nội dung trang web nên có trọng số lớn hơn. Trong phần thân bài, những phần được nhấn mạnh (thường có các thẻ <h1>, <h2>, <b>) cũng được đánh trọng số cao hơn.
- Nhận diện liên kết: Cụm từ xuất hiện kèm với các siêu liên kết có vai trò quan trọng với trang liên kết tới (nhất là các trang ở tên miền khác) vì nó miêu tả ngắn gọn và khách quan nội dung được liên kết
- Xóa các thẻ HTML: Xóa các thẻ HTML không đúng ảnh hưởng tới câu truy vấn lân cận và truy vấn cụm
- Nhận diện các khối nội dung chính: Các trang web thương mại thường chứa các nội dung như quảng cáo, thanh điều hướng, bản quyền.
  - Nhận diện dựa trên thông tin trực quan (vd tọa độ của các khối nội dung)
  - So khớp cây: Các trang web thương mại thường được tùy chỉnh từ một số mẫu nhất định. Dựa trên cấu trúc cây tạo ra từ các trang HTML của cùng một site, việc so khớp cây có thể xác định các mẫu ẩn này. Bên cạnh đó, nội dung của các phần không phải nội dung chính thường tương tự nhau ở các trang khác nhau.

Tiền xử lý trang web (tiếp)

Gõ tiếng Việt
Trợ giúp
Tự động [F9]
Telex (?)
VNI (?)
VIQR (?)
VIQR*
Tắt [F12]
Bỏ dấu kiểu cũ [F7]
Đúng chính tả [F8]
Công cụ
Các liên kết đến đây
Thay đổi liên quan
Các trang đặc biệt
Liên kết thường trực
Thông tin trang
Khoản mục Wikidata
In/xuất ra
Tạo một quyển sách
Tải về dưới dạng PDF
Bản để in ra
Tại dự án khác
Wikimedia Commons
MediaWiki
Meta-Wiki
Wikispecies
Wikibooks
Wikidata
Wikiquote

Bài viết chọn lọc

Sao Kim hay **Kim tinh**, còn gọi là **sao Thái Bạch**, là hành tinh thứ hai trong hệ Mặt Trời, tự quay quanh nó với chu kỳ 224,7 ngày Trái Đất. Xếp sau Mặt Trăng, nó là thiên thể tự nhiên sáng nhất trong bầu trời tối, với cấp sao biểu kiến bằng −4.6, đủ sáng để tạo nên bóng trên mặt nước. Bởi vì Sao Kim là hành tinh phía trong tính từ Trái Đất, nó không bao giờ xuất hiện trên bầu trời mà quá xa Mặt Trời: góc ly giác đạt cực đại bằng 47,8°. Sao Kim đạt độ sáng lớn nhất ngay sát thời điểm hoàng hôn hoặc bình minh, do vậy mà dân gian còn gọi là **sao Hôm**, khi hành tinh này mọc lên lúc hoàng hôn, và **sao Mai**, khi hành tinh này mọc lên lúc bình minh. Sao Kim được xếp vào nhóm hành tinh đất đá và đôi khi người ta còn coi nó là "hành tinh chị em" với Trái Đất do kích cỡ, gia tốc hấp dẫn, tham số quỹ đạo gần giống với Trái Đất. Tuy nhiên, người ta đã chỉ ra rằng nó rất khác Trái Đất trên những mặt khác. Mật độ không khí trong khí quyển của nó lớn nhất trong số bốn hành tinh đất đá, thành phần chủ yếu là cacbon điôxít. Áp suất khí quyển tại bề mặt hành tinh cao gấp 92 lần so với của Trái Đất. Với nhiệt độ bề mặt trung bình bằng 735 K (462 °C), Sao Kim là hành tinh nóng nhất trong Hệ Mặt Trời. Toàn bộ bề mặt của Sao Kim là một hoang mạc khô cằn với đá và bụi và có lẽ vẫn còn núi lửa hoạt động trên hành tinh này. (**xem tiếp...**)

Mới chọn: Stephen Hawking • Trần Thánh Tông • Imagine (bài hát của John Lennon)

Lưu trữ • **Thêm bài viết chọn lọc** • **Ứng cử viên**

Hình ảnh chọn lọc

Bạn có biết...

- ...**Albert xứ Saxe-Coburg và Gotha** và hôn thê tương lai của ông khi chào đời được trợ sinh bởi cùng một bà mụ?
- ...câu thành ngữ ***Ichi-go ichi-e*** khuyên mọi người trân quý những cuộc hội ngộ nguyên là lời cảm thán của bậc thầy trà đạo Sen no Rikyū?
- ...nhà văn **Hugo Gernsback** được mệnh danh là cha đẻ của khoa học viễn tưởng hiện đại?
- ...nhân vật chính Debbie Reynolds và con gái bà trong phim tài liệu ***Bright Lights: Starring Carrie Fisher and Debbie Reynolds*** đều qua đời trước khi phim lên sóng?

Từ những bài viết mới của Wikipedia

Lưu trữ • **Bắt đầu viết bài mới** • **Cập nhật**

Tin tức

- ***Người đẹp và thủy quái*** giành chiến thắng cả bốn đề cử, trong đó có phim hay nhất tại lễ trao **giải BFCA lần thứ 23**
- **Một vụ cháy xe buýt** xảy ra ở khu vực tỉnh Aktobe, Kazakhstan, thiêu chết 52 người.
- Phim ***Three Billboards Outside Ebbing, Missouri*** giành nhiều chiến thắng nhất tại lễ trao **giải Quả cầu vàng lần thứ 75**.
- **George Weah** (*hình*) đắc cử Tổng thống Liberia.

Cập nhật

Ngày này năm xưa

25 tháng 1: Ngày Tatiana tại Nga, ngày cử tri quốc gia tại Ấn Độ.

5.5 Phát hiện trùng lặp

- Trùng lặp là hiện tượng phổ biến trên Web do nhu cầu duyệt web, tải tệp với giới hạn địa lý và hiệu năng của Internet hoặc do đạo văn có chủ ý. Phát hiện trùng lặp có thể giảm khối lượng CSDL và tăng hiệu quả tìm kiếm
- Trùng lặp có thể x/h ở một vài trang hoặc cả website (mirroring)
- Phương pháp hashing hoặc checksum có thể giúp phát hiện trùng lặp hoàn toàn
- Để phát hiện trùng lặp một phần, sử dụng biểu diễn dựa trên n-gram (vd "data mining and web mining" có thể biểu diễn bởi tập các bi-gram S_n={'data mining', 'mining and', 'and web', 'web mining'}) kết hợp với xác lập ngưỡng tối đa theo độ đo tương tự (vd Jaccard)

$$sim(d_1,d_2) = \frac{|S_n(d_1) \cap S_n(d_2)|}{|S_n(d_1) \cup S_n(d_2)|}$$

6. Chỉ mục ngược

- Baseline: Với mỗi câu truy vấn, quét từng văn bản trong CSDL để tìm các từ khóa trong câu truy vấn → không thực tế trong điều kiện tìm kiếm web
- Cấu trúc Chỉ mục ngược làm tăng tốc độ tìm kiếm cũng như tốc độ xây dựng CSDL

6.1 Chỉ mục ngược

- Chỉ mục ngược của một tập văn bản bao gồm danh sách các từ khóa trong đó mỗi từ khóa đi kèm với một danh sách các văn bản chứa nó. Thời gian tìm kiếm từ khóa trong văn bản là hằng số đ/v số lượng văn bản
- Cho một tập văn bản $D=\{d_1,d_2,...,d_N\}$ trong đó mỗi văn bản có một định danh (*id*) duy nhất; chỉ mục ngược của D bao gồm:
  - Từ vựng V bao gồm tất cả các từ trong tập văn bản
  - Mỗi từ t_i có một danh sách ngược trong đó mỗi nút chứa các thông tin $<id_j, f_{ij}, [o_1,o_2,...,o_{|fij|}]>$
    - id_j là định danh của văn bản d_j
    - f_{ij} là tần xuất x/h của t_i trong d_j
    - o_k là độ lệch (vị trí) của t_i trong d_j
  - Các danh sách ngược (và danh sách độ lệch) được sắp xếp dựa trên *id* (và độ lệch)

Chỉ mục ngược (tiếp)

id_1: Web mining is useful.
1 2 3 4
id_2: Usage mining applications.
1 2 3
id_3: Web structure mining studies the Web hyperlink structure.
1 2 3 4 5 6 7 8

V={web,mining,useful,applications,usage,structure,studies,hyperlink}

applications:	*id_2*	*applications:*	*<id_2,1,[3]>*
hyperlink:	*id_3*	*hyperlink:*	*<id_3,1,[7]>*
mining:	*id_1, id_2, id_3*	*mining:*	*<id_1,1,[2]>, < id_2,1,[2]>, <id_3,1[2]>*
structure:	*id_3*	*structure:*	*<id_3,2,[2,8]>*
studies:	*id_3*	*studies:*	*<id_3,1,[4]>*
usage:	*id_2*	*usage:*	*<id_2,1,[1]>*
useful:	*id_1*	*useful:*	*<id_1,1,[4]>*
web:	*id_1, id_3*	*web:*	*<id_1,1,[1]>, <id_3,2,[1,6]>*

a) b)

6.2 Tìm kiếm trong chỉ mục ngược

- Các từ khóa trong câu truy vấn được tìm kiếm trong chỉ mục ngược theo các bước:

1. **Tìm kiếm từ vựng**: Mỗi từ khóa được tìm trong từ vựng để trả về danh sách ngược. Từ vựng được lưu dưới dạng cấu trúc hash, tries, hoặc B-tree theo thứ tự từ điển để tăng tốc độ tìm kiếm. Độ phức tạp tính toán: $O(log|V|)$ trong đó $|V|$ là kích thước từ vựng
2. **Kết hợp kết quả**: Tìm các văn bản chứa toàn bộ các từ khóa. Dựa trên danh sách ngược ngắn nhất, với mỗi văn bản trong danh sách đó, tìm kiếm nhị phân trên các danh sách còn lại (thực hiện tương tự để tìm các văn bản chứa một số từ khóa). Các từ khóa x/h phổ biến trong câu truy vấn (dựa trên phân tích log) được cache trong bộ nhớ để tăng tốc độ xử lý
3. **Tính điểm liên quan**: Tính điểm liên quan của các văn bản dựa trên hàm liên quan (vd: *okapi*, *cosine*) có thể kết hợp truy vấn lân cận hoặc truy vấn cụm và trả về kết quả

Tìm kiếm trong chỉ mục ngược (tiếp)

- Ví dụ với câu truy vấn *q* "web mining":
  - Bước 1: tìm thấy hai danh sách

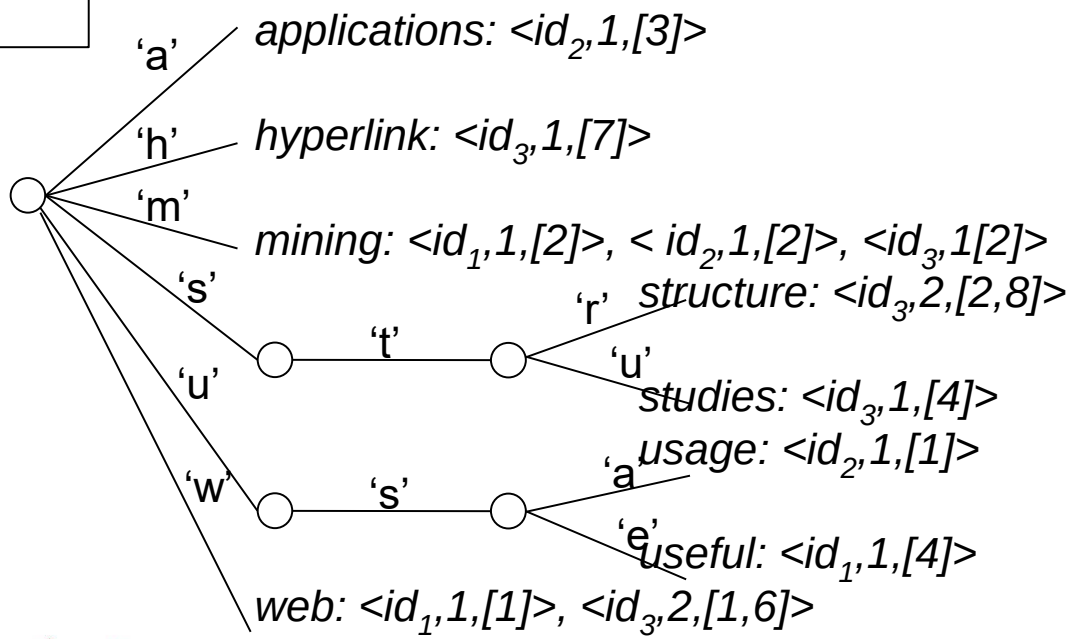
    mining: <id_1,1,[2]>, < id_2,1,[2]>, <id_3,1[2]>

    web: <id_1,1,[1]>, <id_3,2,[1,6]>
  - Bước 2: Tìm thấy hai văn bản id_1 và id_3 chứa cả hai từ khóa
  - Bước 3: id_1 có điểm liên quan cao hơn id_3 vì 'web' và 'mining' đứng cạnh nhau trong id_1 và có cùng thứ tự x/h như trong *q*

6.3 Xây dựng chỉ mục

- Xử dụng cấu trúc trie, độ phức tạp tính toán $O(T)$ trong đó T là kích thước từ vựng

```
for each doc do
    for each term in doc do
        if term in trie update inverted list
        else add term into trie
```



Xây dựng chỉ mục (tiếp)

- Vấn đề: Không đủ bộ nhớ để lưu toàn bộ chỉ mục → Giải pháp:
  - Xây dựng chỉ mục từng phần trên bộ nhớ trong và lưu vào bộ nhớ ngoài $I_1, I_2, \ldots, I_k$
  - Kết hợp đôi một I_1, I_2 thành $I_{1\text{-}2}$, I_3, I_4 thành $I_{3\text{-}4}$ theo từng mức tới khi chỉ còn một chỉ mục I duy nhất
- Vấn đề: Các trang web thường xuyên được cập nhật và xóa đi → Giải pháp:
  - Xây dựng một chỉ mục D_+ cho các trang được cập nhật và D_- cho các trang bị xóa đi
  - Thực hiện truy vấn trên cả ba chỉ mục, kết quả cuối cùng: $D \cup D_+ \setminus D_-$

6.4 Nén chỉ mục

- Nén chỉ mục nhằm giảm khối lượng lưu trữ trong khi không làm mất mát thông tin → áp dụng kĩ thuật nén không mất mát
- Các giá trị đều mang giá trị nguyên dương → áp dụng kĩ thuật nén số nguyên
- Phương pháp variable-bit (bitwise) biểu diễn số nguyên trên một số bit: đơn phân, Elias gamma, delta, và Golomb
- Phương pháp variable-byte sử dụng 7 bit để biểu diễn giá trị và 1 bit phải nhất = 0 nếu là byte cuối cùng và = 1 nếu ngược lại
- Trong danh sách ngược, do id được sắp xếp tăng dần, chỉ cần lưu id bé nhất, sau đó lưu các khoảng cách giữa hai id liên tiếp → giá trị cần lưu bé hơn (tương tự với danh sách vị trí). Vd:

<4,10,300,305> → <4,6,290,5>

Nén chỉ mục (tiếp)

- **Mã hóa đơn phân**: Biểu diễn x bằng x-1 bit 0 và một bit 1. Vd 5: 00001
- **Mã hóa Elias gamma**: Biểu diễn $1+[\log_2 x]$ bằng mã đơn phân theo sau bởi biểu diễn nhị phân của x ngoại trừ bit trái nhất. Phù hợp với các số nhỏ.

Vd: 9: 0001001 do $1+[\log_2 9]=4$ và $9=1001_{(2)}$

  - Giải mã: **1.** Đọc K bit 0 cho tới khi gặp bit 1; **2.** Coi bit 1 là bit đầu tiên của số nguyên (2^K), đọc K bit tiếp theo.

  Vd: Để giải mã 0001001, ta có K=3 bit 0, số nguyên có biểu diễn nhị phân là $1001_{(2)}=9$

- **Mã hóa Elias delta**: Biểu diễn $1+[\log_2 x]$ bằng mã gamma theo sau bởi biểu diễn nhị phân của x ngoại trừ bit trái nhất. Phù hợp với các số lớn.

Vd: 9=00100001 do $1+[\log_2 9]=4$ có biểu diễn gamma 00100

  - Giải mã: **1.** Đọc L bit 0 cho tới khi gặp bit 1 đầu tiên. **2.** Coi bit *1* là bit đầu tiên của số nguyên (2^L), đọc *L* bit tiếp theo có được số nguyên M. **3.** Đặt bit 1 đầu tiên (2^M) và đọc tiếp M-1 bit tiếp theo.

  Vd: Giải mã 00100001, **1.** L=2; **2.** M=4; **3.** $1001_{(2)}=9$

Nén chỉ mục (tiếp)

- **Mã hóa Golomb**:

1. Phần đầu là biểu diễn đơn phân của q+1 với q=[(x/b)]

2. Phần tiếp theo là biểu diễn nhị phân của số dư r=x-qb. Xét i = $[\log_2 b]$, $d=2^{i+1}-b$ số dư đầu tiên biểu diễn bằng i bit; các số dư tiếp theo biểu diễn bằng $[\log_2 b]$ (i+1) bit (fixed prefix coding)

Vd: x=9, b=3; q=[(9/3)]=3, i=$[\log_2 3]$=1,

$d=2^{1+1}-3=1$, r=9-3x3=0 → 9: 00010

- Lựa chọn b:

$$b \approx 0.69 \left(\frac{N}{n_t} \right)$$

trong đó N là tổng số văn bản, n_t là số văn bản chứa t

Nén chỉ mục (tiếp)

- **Giải mã**:

1. Giãi mã q

2. $i = \lfloor \log_2 b \rfloor$, $d = 2^{i+1} - b$

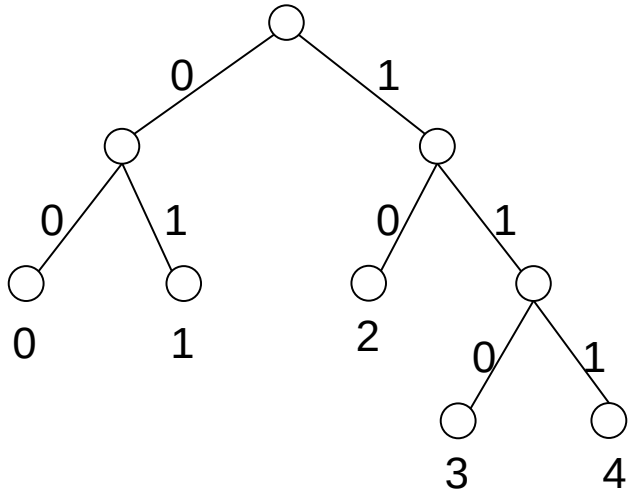
3. Lấy i bit tiếp theo đưa vào r

4. Nếu $r \geq d$

lấy thêm 1 bit chèn vào cuối r

r = r - d

5. x = qb + r



Cây mã hóa với b=5

- Vd: Giải mã 11111 với b = 10

1. q = 0 **2.** $i = \lfloor \log_2 10 \rfloor = 3$, $d = 2^{3+1} - 10 = 6$ **3.** $r = 111_{(2)} = 7$ **4.** $7 > 6 \rightarrow r = 1111_{(2)} = 15$; $r = 15 - 6 = 9$ **5.** x = 0x10+9=9

Nén chỉ mục (tiếp)

- **Mã hóa variable-byte**: 7 bit để biểu diễn giá trị, bit phải nhất = 0 nếu là byte cuối cùng, = 1 nếu ngược lại. Phù hợp biểu diễn các số nhỏ. Vd: 135: 00000011 00001110
- **Giải mã**:

1. Đọc lần lượt các byte tới khi gặp byte có bit phải nhất = 0
2. Xóa các bit phải nhất của các byte và ghép nối các bit còn lại với nhau theo đúng thứ tự đọc

Vd: 00000011 00001110 được giải mã thành $00000010000111_{(2)} = 135$

- Nhận xét:
  - Mã hóa tham số hóa Golomb có tỉ lệ nén tốt hơn và tốc độ truy vấn nhanh hơn các phương pháp không tham số hóa
  - Mã hóa theo byte có tốc độ truy vấn nhanh hơn mặc dù sử dụng nhiều bộ nhớ hơn
  - Trung bình, mã hóa giúp tốc độ truy vấn tăng gấp đôi và dung lượng chỉ mục giảm bốn đến năm lần

7. Đánh chỉ mục ngữ nghĩa ẩn

- Hiện tượng đồng nghĩa dẫn đến câu truy vấn và văn bản sử dụng các từ khác nhau để diễn đạt cùng một khái niệm. Vd: với câu truy vấn chứa từ ‘picture’, các văn bản chứa từ ‘photo’ và ‘image’ sẽ không được trả về.
- Đánh chỉ mục ngữ nghĩa - LSI (Latent Semantic Indexing) nhằm tới giải quyết vấn đề này dựa trên khai thác tương quan thông kê của các từ trong tập văn bản.
- Dựa trên giả thuyết tồn tại một cấu trúc ngữ nghĩa ẩn đăng sau sự x/h “ngẫu nhiên” của các từ trong văn bản, LSI sử dụng kĩ thuật Singular Value Decomposition (SVD) để ước lượng cấu trúc ẩn này (không gian khái niệm ẩn) và loại bỏ nhiễu.
- Các từ, văn bản và câu truy vấn đều được chuyển đổi về không gian khái niệm ẩn

7.1 SVD

- SVD thực hiện phân rã ma trận m x n **A** thành tích của 3 ma trận con

$$\mathbf{A}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

trong đó

U là ma trận m x r trong đó các cột (véc-tơ singular trái) là các véc-tơ riêng ứng với r trị riêng của $\mathbf{A}\mathbf{A}^T$; $\mathbf{U}^T\mathbf{U}=\mathbf{I}$

V là ma trận n x r trong đó các cột (véc-tơ singular phải) là các véc-tơ riêng ứng với r trị riêng của $\mathbf{A}^T\mathbf{A}$; $\mathbf{V}^T\mathbf{V}=\mathbf{I}$

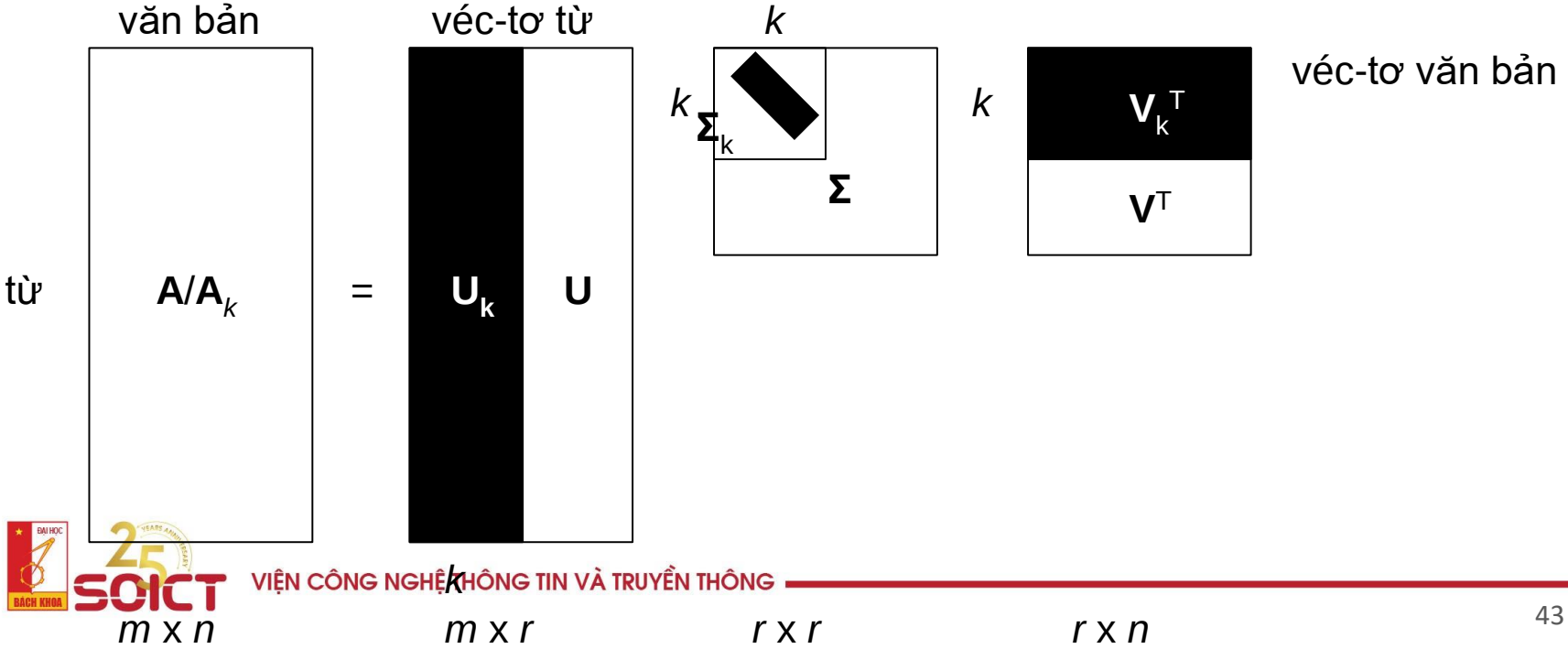
Σ là ma trận đường chéo r x r, $\mathbf{\Sigma}=\text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$, $\sigma_i > 0$. $\sigma_1, \sigma_2, \ldots, \sigma_r$ (các giá trị singular) là căn bậc hai không âm của r trị riêng (khác không) của $\mathbf{A}\mathbf{A}^T$. $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$

- m: số từ trong từ vựng; n: tổng số văn bản; r là bậc của **A**, $r \leq \min(m, n)$

SVD (tiếp)

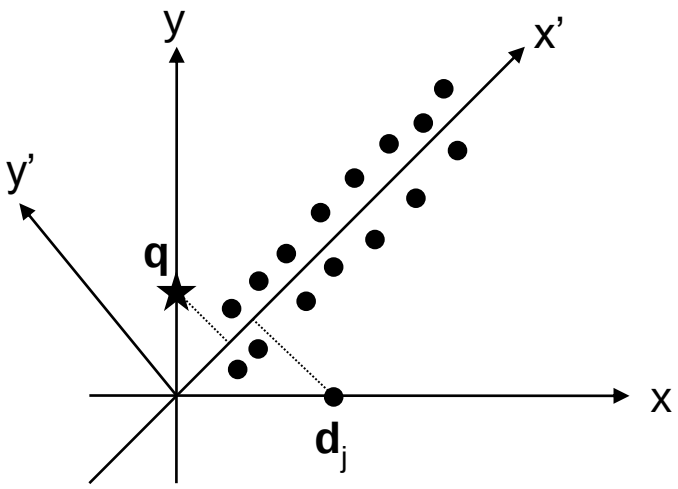
$$\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$$

- Giữ lại k thành phần lớn nhất để khôi phục xấp xỉ của $\mathbf{A}$ ($\mathbf{A}_k$), tương ứng với cấu trúc ẩn quan trọng trong không gian khái niệm, loại bỏ các thành phần nhiễu



SVD (tiếp)

- SVD xoay không gian *m*-chiều của A sao cho trục thứ nhất, thứ hai, ... ứng với hướng dữ liệu biến đổi nhiều nhất, nhiều thứ hai, ...
- Giả sử trục gốc là *x*-*y* và trục do SVD sinh ra là *x'*-*y'*. *x* và *y* có mối tương quan rõ ràng. Ta có thể loại bỏ trục *y'* do nó không có vai trò đáng kể. Một văn bản $\boldsymbol{d}_j$ chỉ chứa *x* và câu truy vấn ***q*** chỉ chứa *y* sau khi được chiếu lên *x'* trở nên tương đồng



Giả thiết của LSI

7.2 Truy vấn và truy hồi

$$A_k = U_k \Sigma_k V_k^T$$

- Biểu diễn câu truy vấn $\boldsymbol{q}$ trong không gian gốc, ta có:

$$q = U_k \Sigma_k q_k^T$$

- Do $\boldsymbol{U}_k$ gồm các véc-tơ đơn vị vuông góc, $\boldsymbol{U}_k^T \boldsymbol{U}_k = \boldsymbol{I}$

$$U_k^T q = \Sigma_k q_k^T$$

$$\Sigma_k^{-1} U_k^T q = q_k^T$$

- Cuối cùng, ta có:

$$q_k = \Sigma_k^{-1} U_k q^T$$

- So sánh $\boldsymbol{q}_k$ với các văn bản trong không gian k-chiều sử dụng độ đo tương đồng (vd cosine)

7.3 Ví dụ

human-computer interaction

graphs

c_1: Human machine interface for Lab ABC computer applications
c_2: A survey of user opinion of computer system response time
c_3: The EPS user interface management system
c_4: System and human system engineering testing of EPS
c_5: Relation of user-perceived response time to error measurement
m_1: The generation of random, binary, unordered trees
m_2: The intersection graph of paths in trees
m_3: Graph minors IV: Widths of trees and well-quasi-ordering
m_4: Graph minors: A survey

A=

c_1	c_2	c_3	c_4	c_5	m_1	m_2	m_3	m_4	
1	0	0	1	0	0	0	0	0	*human*
1	0	1	0	0	0	0	0	0	*interface*
1	1	0	0	0	0	0	0	0	*computer*
0	1	1	0	1	0	0	0	0	*user*
0	1	1	2	0	0	0	0	0	*system*
0	1	0	0	1	0	0	0	0	*response*
0	1	0	0	1	0	0	0	0	*time*
0	0	1	1	0	0	0	0	0	*EPS*
0	1	0	0	0	0	0	0	1	*survey*
0	0	0	0	0	1	1	1	0	*trees*
0	0	0	0	0	0	1	1	1	*graph*
0	0	0	0	0	0	0	1	1	*minors*

$$\mathbf{U} = \begin{pmatrix} 0.22 & -0.11 & 0.29 & -0.41 & -0.11 & -0.34 & 0.52 & -0.06 & -0.41 \\ 0.20 & -0.07 & 0.14 & -0.55 & 0.28 & 0.50 & -0.07 & -0.01 & -0.11 \\ 0.24 & 0.04 & -0.16 & -0.59 & -0.11 & -0.25 & -0.30 & 0.06 & 0.49 \\ 0.40 & 0.06 & -0.34 & 0.10 & 0.33 & 0.38 & 0.00 & 0.00 & 0.01 \\ 0.64 & -0.17 & 0.36 & 0.33 & -0.16 & -0.21 & -0.17 & 0.03 & 0.27 \\ 0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\ 0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\ 0.30 & -0.14 & 0.33 & 0.19 & 0.11 & 0.27 & 0.03 & -0.02 & -0.17 \\ 0.21 & 0.27 & -0.18 & -0.03 & -0.54 & 0.08 & -0.47 & -0.04 & -0.58 \\ 0.01 & 0.49 & 0.23 & 0.03 & 0.59 & -0.39 & -0.29 & 0.25 & -0.23 \\ 0.04 & 0.62 & 0.22 & 0.00 & -0.07 & 0.11 & 0.16 & -0.68 & 0.23 \\ 0.03 & 0.45 & 0.14 & -0.01 & -0.30 & 0.28 & 0.34 & 0.68 & 0.18 \end{pmatrix}$$

$$\mathbf{\Sigma} = \begin{pmatrix} 3.34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.54 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.35 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.64 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.31 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.85 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.56 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.36 \end{pmatrix}$$

$$V = \begin{pmatrix} 0.20 & -0.06 & 0.11 & -0.95 & 0.05 & -0.08 & 0.18 & -0.01 & -0.06 \\ 0.61 & 0.17 & -0.50 & -0.03 & -0.21 & -0.26 & -0.43 & 0.05 & 0.24 \\ 0.46 & -0.13 & 0.21 & 0.04 & 0.38 & 0.72 & -0.24 & 0.01 & 0.02 \\ 0.54 & -0.23 & 0.57 & 0.27 & -0.21 & -0.37 & 0.26 & -0.02 & -0.08 \\ 0.28 & 0.11 & -0.51 & 0.15 & 0.33 & 0.03 & 0.67 & -0.06 & -0.26 \\ 0.00 & 0.19 & 0.10 & 0.02 & 0.39 & -0.30 & -0.34 & 0.45 & -0.62 \\ 0.01 & 0.44 & 0.19 & 0.02 & 0.35 & -0.21 & -0.15 & -0.76 & 0.02 \\ 0.02 & 0.62 & 0.25 & 0.01 & 0.15 & 0.00 & 0.25 & 0.45 & 0.52 \\ 0.08 & 0.53 & 0.08 & -0.03 & -0.60 & 0.36 & 0.04 & -0.07 & -0.45 \end{pmatrix}$$

U_k Σ_k V_k

$$A_k = \begin{pmatrix} 0.22 & -0.11 \\ 0.20 & -0.07 \\ 0.24 & 0.04 \\ 0.40 & 0.06 \\ 0.64 & -0.17 \\ 0.27 & 0.11 \\ 0.27 & 0.11 \\ 0.30 & -0.14 \\ 0.21 & 0.27 \\ 0.01 & 0.49 \\ 0.04 & 0.62 \\ 0.03 & 0.45 \end{pmatrix} \begin{pmatrix} 3.34 & 0 \\ 0 & 2.54 \end{pmatrix} \begin{pmatrix} 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.02 & 0.02 & 0.08 \\ -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \end{pmatrix}$$

Câu truy vấn "*user interface*"

$$\mathbf{q}_k = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}^T \begin{pmatrix} 0.22 & -0.11 \\ 0.20 & -0.07 \\ 0.24 & 0.04 \\ 0.40 & 0.06 \\ 0.64 & -0.17 \\ 0.27 & 0.11 \\ 0.27 & 0.11 \\ 0.30 & -0.14 \\ 0.21 & 0.27 \\ 0.01 & 0.49 \\ 0.04 & 0.62 \\ 0.03 & 0.45 \end{pmatrix} \begin{bmatrix} 3.34 & 0 \\ 0 & 2.54 \end{bmatrix}^{-1} = (0.179\text{-}0.004)$$

cosine($\mathbf{q}$,$\mathbf{d}_j$):

c_1: 0.964
c_2: 0.957
c_3: 0.968
c_4: 0.928
c_5: 0.922
m_1: 0.022
m_2: 0.023
m_3: 0.010
m_4: 0.127

Xếp hạng: (c_3,c_1,c_2,c_4,c_5,m_4,m_3,m_2,m_1)

7.3 Thảo luận

- Ưu điểm: LSI cho kết quả truy vấn tốt hơn mô hình IR truyền thống
- Nhược điểm:
  - Độ phức tạp tính toán O(m^2n), không phù hợp với tìm kiếm Web
  - Tính diễn giải của không gian khái niệm kém
  - Việc xác định k dựa trên thực nghiệm (vd 50-350)
- Hướng phát triển: Áp dụng luật kết hợp để tìm các chuỗi từ (2-3 từ) phổ biến trong tập văn bản

8. Tìm kiếm Web

- Các máy tìm kiếm thương mại (vd Google, Bing, Baidu, Yandex)
- Quy trình: crawl trang web, parsing, đánh chỉ mục, lưu trữ; truy vấn, truy hồi
- Parsing: Đọc tệp HTML và sử dụng các bộ phân tích từ vựng (vd các công cụ mã nguồn mở YACC và Flex)
- Đánh chỉ mục: Kết hợp bộ chỉ mục của tiêu đề và các cụm từ x/h trong siêu liên kết
- Tìm kiếm và xếp hạng:
  - Chỉ sử dụng mức độ liên quan của nội dung có thể trả về các trang web chất lượng không cao (do tác giả không phải là chuyên gia trong lĩnh vực; thông tin không chính xác; thông tin bị thiên vị…).
  - Số lượng trang web rất lớn (Câu truy vấn "*web mining*" cho hơn 1 tỉ kết quả từ Gooogle), cần trả về các kết quả tốt nhất ở trang đầu

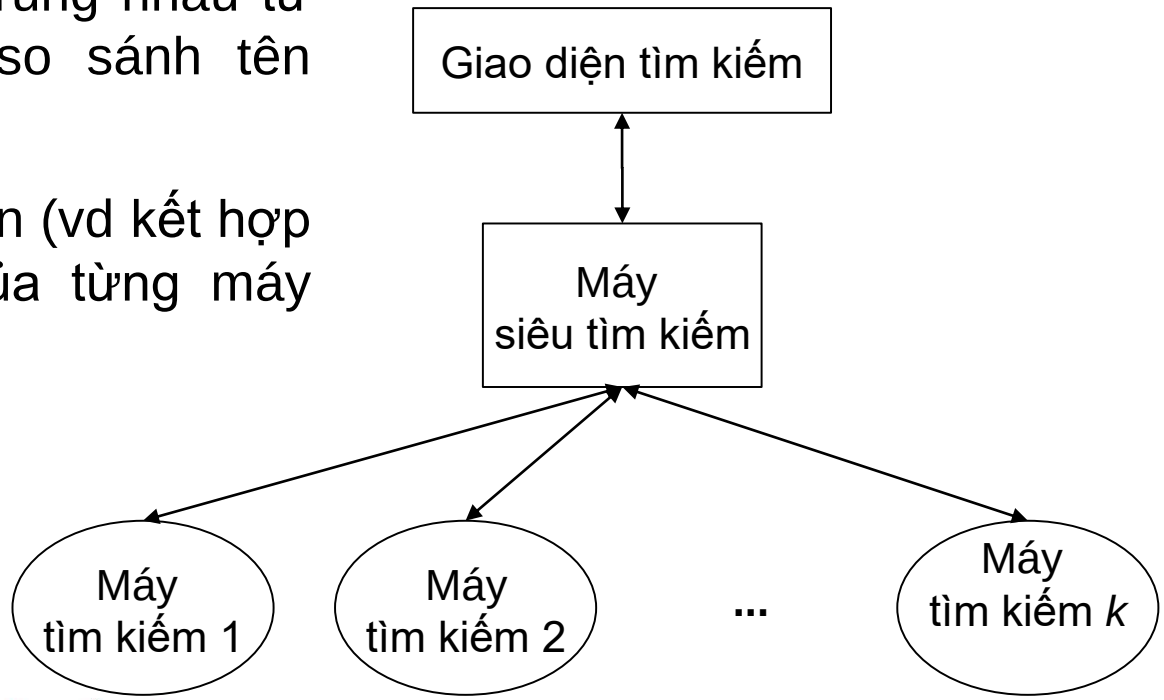
Tìm kiếm web (tiếp)

- Đánh giá chất lượng trang web thông qua khai phá cấu trúc web dựa trên siêu liên kết: Một siêu liên kết từ trang web A tới trang web B thể hiện tác giả của trang web A tin tưởng vào chất lượng và/hoặc uy tín của tác giả trang web B. Có thể ước lượng dựa trên thông tin cục bộ (in-degree) hoặc thông tin toàn cục (*Pagerank*)
- Đánh giá nội dung:
  - Vị trí từ khóa x/h: Tiêu đề, cụm từ trong siêu liên kết, URL, thân bài (các thẻ đánh dấu như in nghiêng, in đậm, đề mục...)
  - Tần xuất xuất hiện của từ khóa
  - Vị trí tương đối của các từ khóa: Từ khóa x/h gần nhau, có trật tự giống như trong câu truy vấn có độ liên quan cao hơn

9. Siêu tìm kiếm

- Máy siêu tìm kiếm kết hợp kết quả của nhiều máy tìm kiếm khác nhau
- Cải thiện mức độ bao phủ
- Cải thiện chất lượng tìm kiếm

- Loại bỏ các kết quả trùng nhau từ các máy tìm kiếm (so sánh tên miền, URL, tiêu đề...)
- Kết hợp điểm liên quan (vd kết hợp điểm tương đồng) của từng máy tìm kiếm



9.1 Kết hợp điểm tương đồng

- Tập các văn bản trả về $D = \{d_1, d_2, \ldots, d_N\}$, máy tìm kiếm i trả về s_{ij} là độ tương đồng của câu truy vấn với văn bản d_j

$\text{CombMIN}(d_j) = \min(s_{1j}, s_{2j}, \ldots, s_{kj})$

$\text{CombMAX}(d_j) = \max_k(s_{1j}, s_{2j}, \ldots, s_{kj})$

$\text{CombSUM}(d_j) = \Sigma_{i=1} s_{ij}$

$$\text{CombANZ}(d_j) = \frac{\text{CombSUM}(d_j)}{r_j}$$

trong đó r_j là số điểm tương đồng khác không

$\text{CombMNZ}(d_j) = \text{CombSUM}(d_j) \times r_j$

9.2 Kết hợp thứ hạng

- Xếp hạng *Borda*: Với mỗi máy tìm kiếm, văn bản đầu tiên được n điểm (n là tổng số văn bản), văn bản thứ hai được n-1 điểm, các văn bản không được máy tìm kiếm trả về được chia đều số điểm còn lại. Điểm cuối cùng của văn bản là tổng điểm trên các máy tìm kiếm.
- Xếp hạng *Condorcet*: Văn bản A xếp trên văn bản B nếu A xếp trên B theo nhiều máy tìm kiếm hơn. Trong cùng máy tìm kiếm, văn bản được xếp hạng xếp trên văn bản không được xếp hạng. Các văn bản không được xếp hạng không so sánh được.
- *Reciprocal ranking*: Với mỗi máy tìm kiếm, văn bản đứng đầu được 1 điểm, thứ hai được ½ điểm, không xếp hạng không được điểm. Điểm cuối cùng của văn bản là tổng điểm trên các máy tìm kiếm

Kết hợp thứ hạng (tiếp)

Borda:
Điểm(a) = 4 + 3 + 2 + 1 + 1.5 = 11.5
Điểm(b) = 3 + 4 + 3 + 3 + 3 = 16
Điểm(c) = 2 + 1 + 4 + 4 + 4 = 15
Điểm(d) = 1 + 2 + 1 + 2 + 1.5 = 7.5
Thứ hạng: *b, c, a, d*

Reciprocal Ranking:
Điểm(a) = 1 + 1/2 + 1/3 = 1.83
Điểm(b) = 1/2 + 1 + 1/2 + 1/2 + 1/2 = 3
Điểm(c) = 1/3 + 1/4 + 1 + 1 + 1 = 3.55
Điểm(d) = 1/4 + 1/3 + 1/4 + 1/3= 1.17
Thứ hạng: c, b, a, d.

Kết quả máy tìm kiếm:
Máy tìm kiếm 1: a , b , c , d
Máy tìm kiếm 2: b , a , d , c
Máy tìm kiếm 3: c , b , a , d
Máy tìm kiếm 4: c , b , d
Máy tìm kiếm 5: c , b

Condorcet:

	a	*b*	*c*	*d*
a	-	1:4:0	2:3:0	3:1:1
b	4:1:0	-	2:3:0	5:0:0
c	3:2:0	3:2:0	-	4:1:0
d	1:3:1	0:5:0	1:4:0	-

	Thắng	Thua	Hòa
a	1	2	0
b	2	1	0
c	3	0	0
d	0	3	0

Thứ hạng: *c, b, a, d*

10. Web spam

- Gia tăng nội dung và sự ảnh hưởng trên web sẽ đem lại sự nổi tiếng và lợi ích tài chính cho tổ chức và cá nhân
- Khi người dùng tìm kiếm thông tin qua câu truy vấn trên máy tìm kiếm, các trang web được xếp hạng cao mang lại lợi ích cho doanh nghiệp, tổ chức, và cá nhân đăng tải trang web đó
- Với một câu truy vấn, giả sử các trang web có mang nhiều giá trị thông tin được xếp hạng cao hơn. Tuy nhiên, máy tìm kiếm không "hiểu" được thông tin và đưa ra xếp hạng dựa trên các đặc trưng cú pháp và các đặc trưng bề nổi khác để đánh giá thông tin. Spammer lợi dụng việc hiểu cơ chế xếp hạng của máy tìm kiếm để xây dựng nội dung trang web sao cho mặc dù không mang giá trị thông tin cao nhưng vẫn được xếp hạng cao (SEO - Search Engine Optimization)
- Web spam làm ảnh hưởng người dùng khiến họ khó tìm thông tin thực sự và làm giảm trải nghiệm người dùng; lãng phí tài nguyên crawl và lưu trữ của máy tìm kiếm và làm cho xếp hạng của máy tìm kiếm bị giảm chất lượng

10.1 Spam nội dung

- Spam nội dung xây dựng nội dung trang web (tiêu đề, các thẻ meta, thân bài, cụm từ trong siêu liên kết, URL) liên quan đến một số câu truy vấn nhất định
- Hai kĩ thuật spam chính:
  - Lặp lại một số từ quan trọng: Mục đích làm tăng điểm *tf*. Các từ quan trọng thường được chèn ngẫu nhiên vào các câu khác nhau (vd "the picture *mining* quality of this camera *mining* is amazing") để tránh bị phát hiện
  - Thêm nhiều từ khóa không liên quan: Mục đích làm cho trang web liên quan đến nhiều câu truy vấn khác nhau. Sao chép các câu từ các văn bản liên quan. Thêm các từ khóa được tìm kiếm phổ biến (vd: thêm từ khóa "Tom Cruise" vào trang web cung cấp các chuyến du lịch trên tàu (cruise liner, cruise holiday packages))

10.2 Spam liên kết

- Liên kết-ra: Tạo ra siêu liên kết đến các trang có uy tín bằng cách clone từ các thư mục web (vd Yahoo!)
- Liên kết-vào:
  1. Tạo ra *hũ mật ong*: Tạo ra các trang web chứa các thông tin hữu ích (vd các bài viết chuyên sâu về khai phá Web) nhằm thu hút người dùng tạo ra các liên kết từ trang web của họ tới đó. Các hũ mật ong này chứa các liên kết (ẩn) tới những trang cần tăng điểm
  2. Thêm các liên kết vào các thư mục Web mở
  3. Đăng bài viết (kèm liên kết) lên các trang nội dung người dùng như forum, blog, wiki
  4. Tham gia vào các nhóm trao đổi liên kết khiến cho các site liên kết đến lẫn nhau và cùng tăng điểm
  5. Nắm giữ một số lượng lớn site và tạo ra các chiến dịch để tăng điểm cho trang web

10.3 Các kĩ thuật che dấu

- Mục đích che dấu các nội dung spam khỏi người dùng
- Che dấu nội dung: Sử dụng font hoặc siêu liên kết cùng màu nền, sử dụng script để thiết lập nội dung spam ở trạng thái ẩn

<body background = white>

* spam items*

*<img src="blank.gif"> *

- Đưa các thông tin spam tới crawler của máy tìm kiếm:
  - Dựa trên danh sách IP của máy tìm kiếm và so sánh với IP của crawler
  - Dựa trên thuộc tính user-agent của request

  User–Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
- Điều hướng: Điều hướng người dùng khỏi trang spam ngay khi trang được nạp. Trang spam vẫn được máy tìm kiếm crawl và đánh chỉ mục

10.4 Các kĩ thuật đối phó

- Xác lập crawler như trình duyệt web
- Đánh trọng số cao cho các từ khóa trong cụm từ x/h trong siêu liên kết
- Áp dụng các kĩ thuật phân tích liên kết: thuật toán PageRank, điểm authority/hub
- Áp dụng các kĩ thuật phân loại. Cho độ chính xác khá cao (>80%) nhưng đòi hỏi dữ liệu huấn luyện. Các đặc trưng quan trọng bao gồm:
  - Độ dài của tiêu đề, nội dung: Tiêu đề và nội dung của trang spam thường dài hơn do chứa thêm các từ khóa phổ biến
  - Độ dài trung bình của từ: Các nội dung tổng hợp (synthetic) thường có độ dài trung bình của từ khác biệt (vd độ dài trung bình của từ trong ngôn ngữ nói và viết tiếng Anh là 5)
  - Tỉ lệ nội dung hiện thị: Trang spam thường chứa các thành phần nội dung ẩn để che dấu người dùng
- Phân vùng các khối nội dung khác nhau: Các liên kết spam thường được đặt trong các khối ít quan trọng. Loại bỏ các liên kết này trước khi áp dụng thuật toán PageRank

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

soict.hust.edu.vn/ fb.com/groups/soict