

Tìm kiếm thông tin

Chương 6. Các kỹ thuật nâng cao chất lượng tìm kiếm

Soạn bởi: TS. Nguyễn Bá Ngọc

Nâng cao chất lượng tìm kiếm là gì?


- Chất lượng tìm kiếm có thể được đánh giá thông qua nhiều chỉ số khác nhau
 - *(Nội dung Chương 5. Đánh giá kết quả tìm kiếm).*
- Trong phạm vi bài giảng này chúng ta chủ yếu quan tâm đến số lượng kết quả phù hợp được trả về ở những vị trí đầu danh sách:
 - Tăng độ chính xác và độ đầy đủ trong phạm vi $\text{top}@K$,
 - nhưng có thể giảm nếu mở rộng phạm vi đánh giá ngoài $\text{top}@K$

Nâng cao chất lượng tìm kiếm bằng cách nào?

- Xét một truy vấn q : [phi cơ] ... và văn bản d chứa từ “máy bay”, nhưng không chứa từ phi cơ:
 - Nếu chỉ dựa trên dấu hiệu từ truy vấn xuất hiện trong văn bản thì hệ thống TKTT có thể không trả về d , dù d có thể là văn bản phù hợp nhất với q .
 - *Tất nhiên chúng ta muốn hệ thống trả về các văn bản phù hợp với truy vấn dù không chứa từ truy vấn.*
- Viết lại câu truy vấn, ví dụ bổ xung thêm từ máy bay (từ đồng nghĩa), là 1 cách tiếp cận tiêu biểu để giải quyết vấn đề trong ví dụ này.

Chúng ta sẽ tiếp tục phân tích một số phương pháp nhằm nâng cao chất lượng tìm kiếm

Nội dung

- 
1. Phản hồi kết quả phù hợp và giải thuật Rocchio
 2. Dữ liệu hành vi người dùng
 3. Đánh giá kết quả có phản hồi
 4. Mở rộng truy vấn dựa trên từ điển
 5. Tự động xác định các từ liên quan
 6. Phân tích ngữ nghĩa ẩn

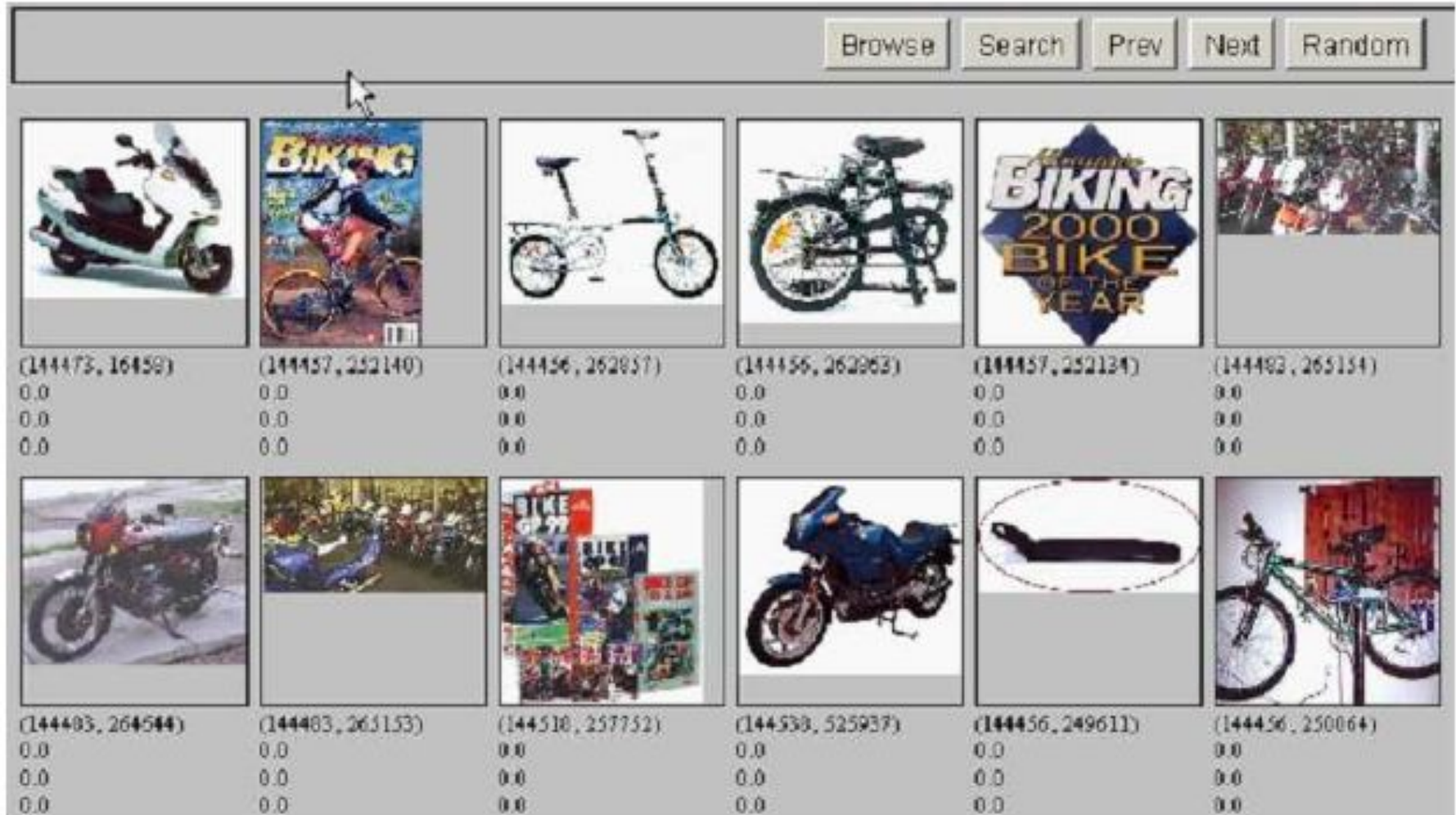
Cơ chế phản hồi kết quả phù hợp

- Người dùng gửi một truy vấn (ngắn, đơn giản)
- Máy tìm kiếm trả về một danh sách kết quả
- Người dùng đánh dấu văn bản phù hợp hoặc không
- Máy tìm kiếm tạo một truy vấn mới dựa trên phản hồi.
- Máy tìm kiếm thực hiện truy vấn mới và trả về các kết quả
 - Các kết quả mới (được kỳ vọng) có chất lượng tốt hơn

(Chúng ta có thể thực hiện nhiều vòng phản hồi)

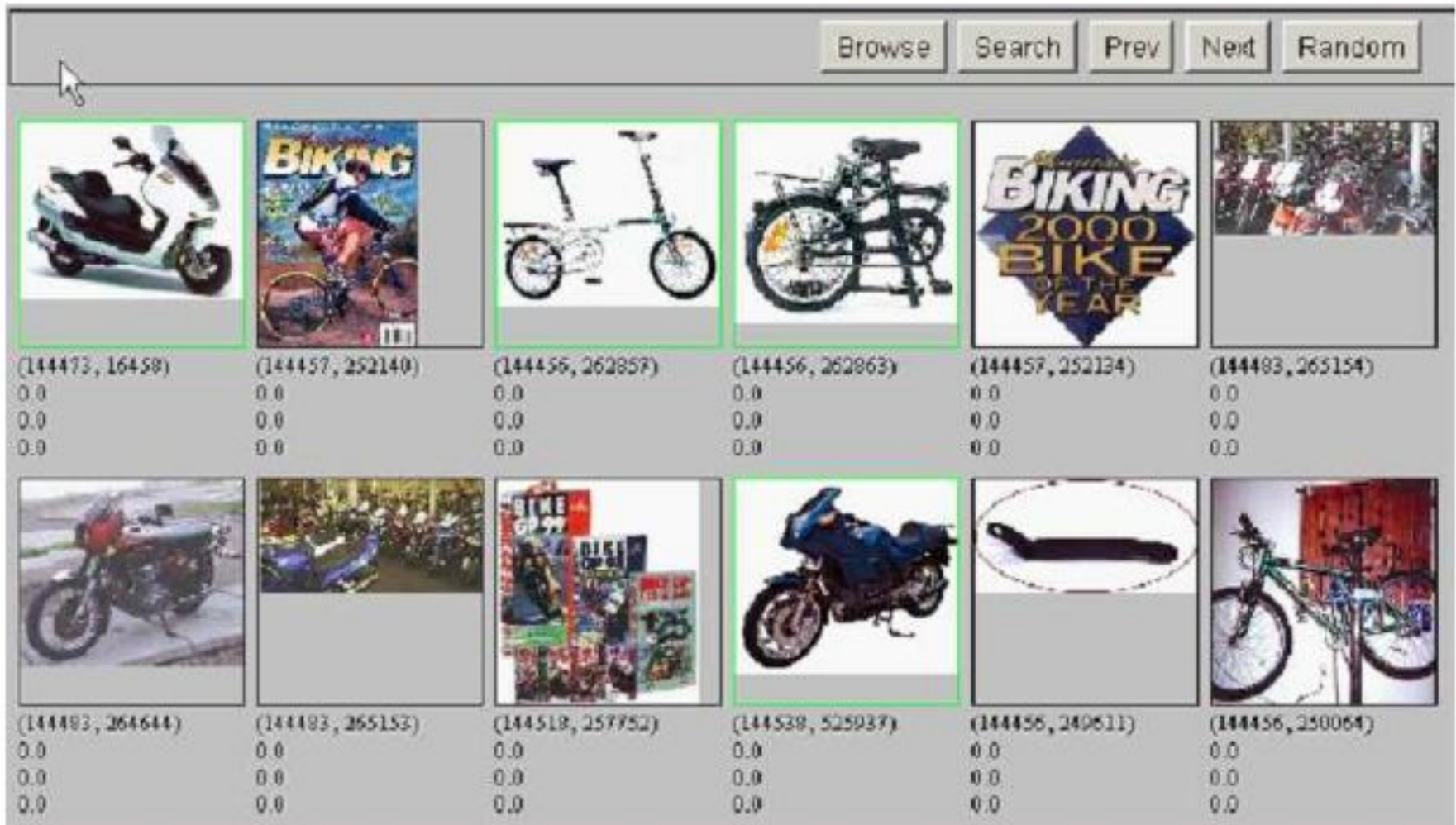
Ví dụ 6.1. Phản hồi trong tìm kiếm ảnh

Kết quả truy vấn ban đầu: Bike



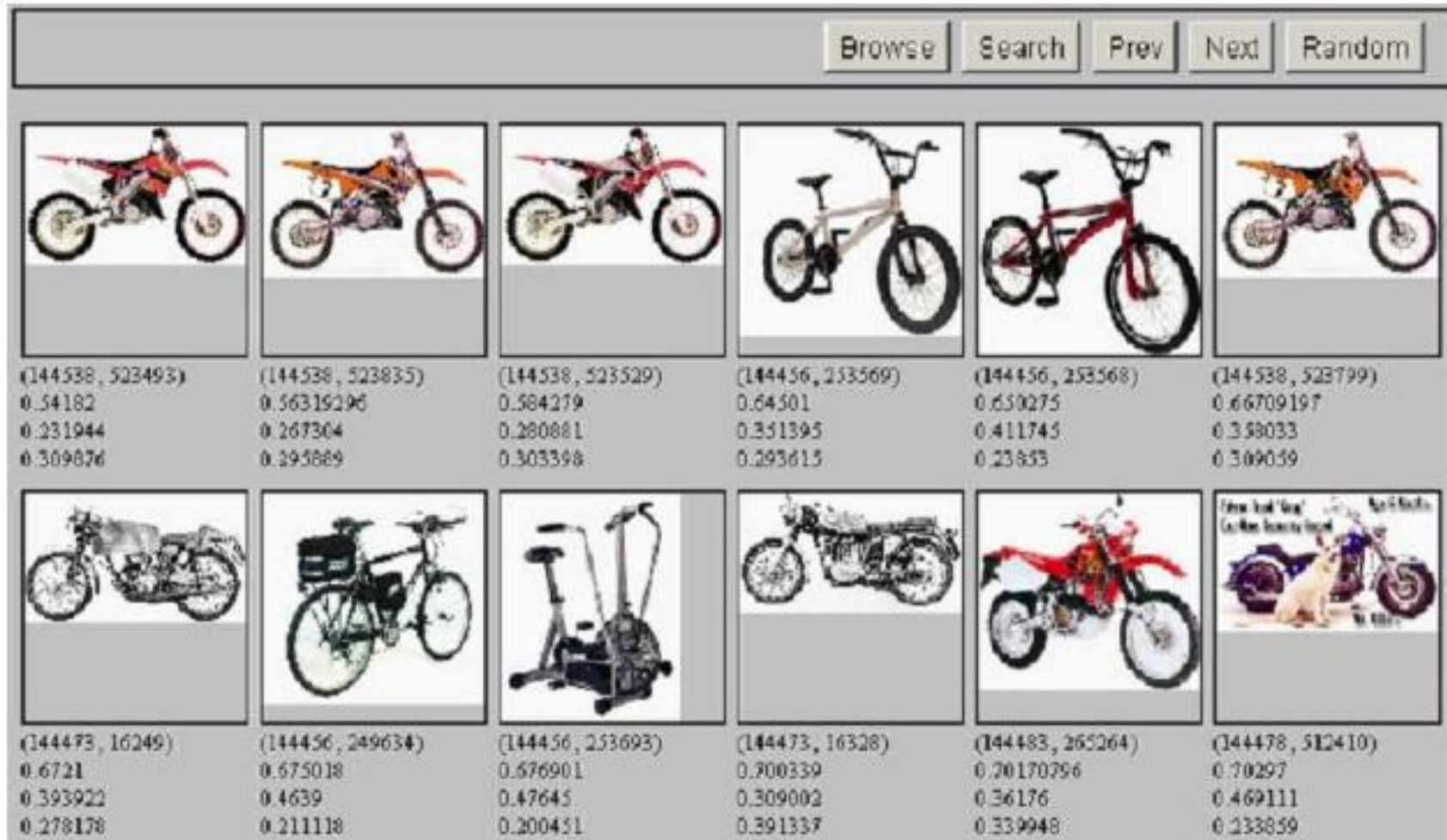
Ví dụ 6.1. Phản hồi trong tìm kiếm ảnh⁽²⁾

Người dùng lựa chọn các kết quả phù hợp



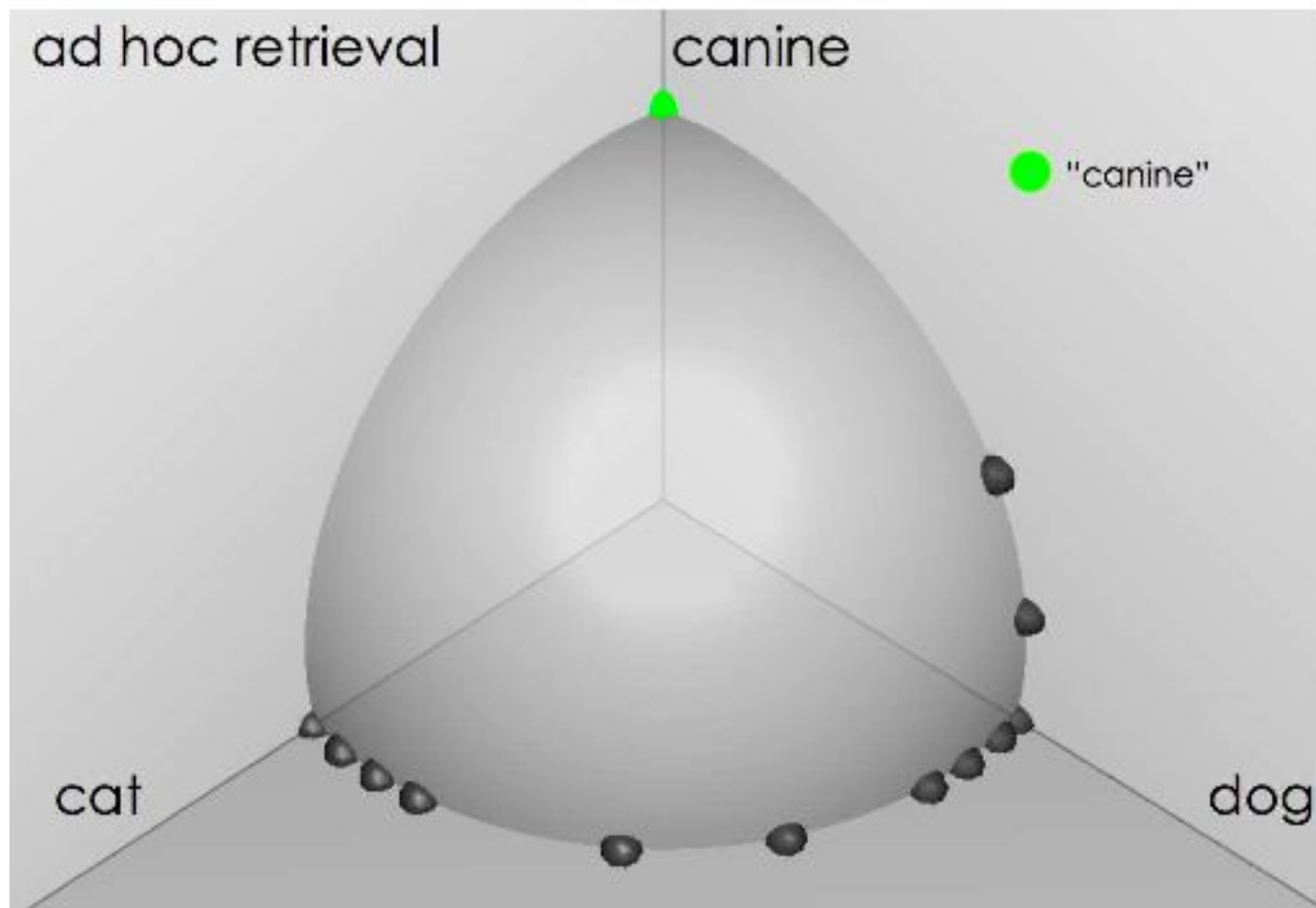
Ví dụ 6.1. Phản hồi trong tìm kiếm ảnh⁽³⁾

Kết quả tìm kiếm sau khi xử lý các phản hồi



Ví dụ 6.2. Phản hồi trong không gian vec-tơ

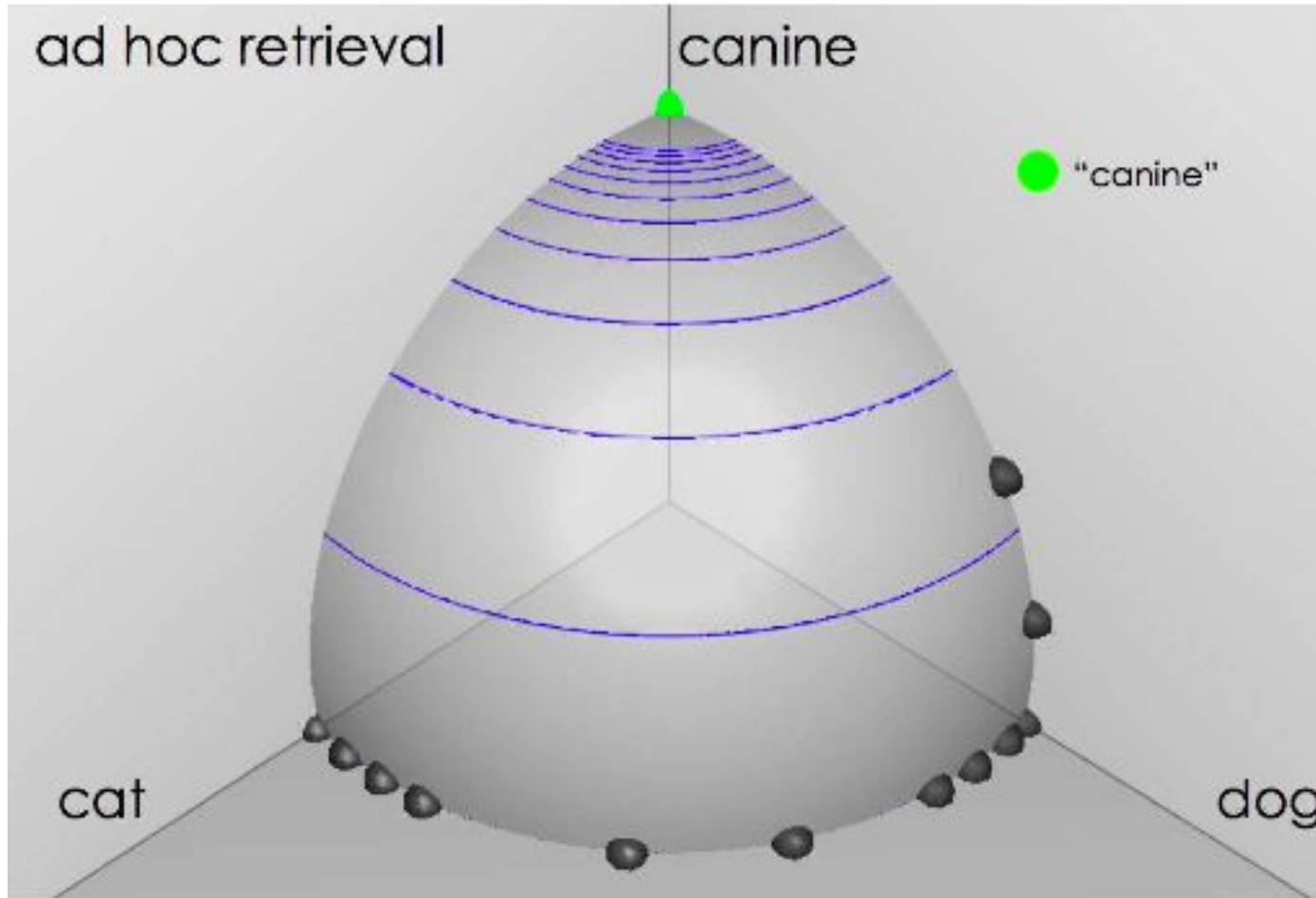
Truy vấn “canine”



Nguồn: [Fernando Díaz]

Ví dụ 6.2. Phản hồi trong không gian vec-tơ

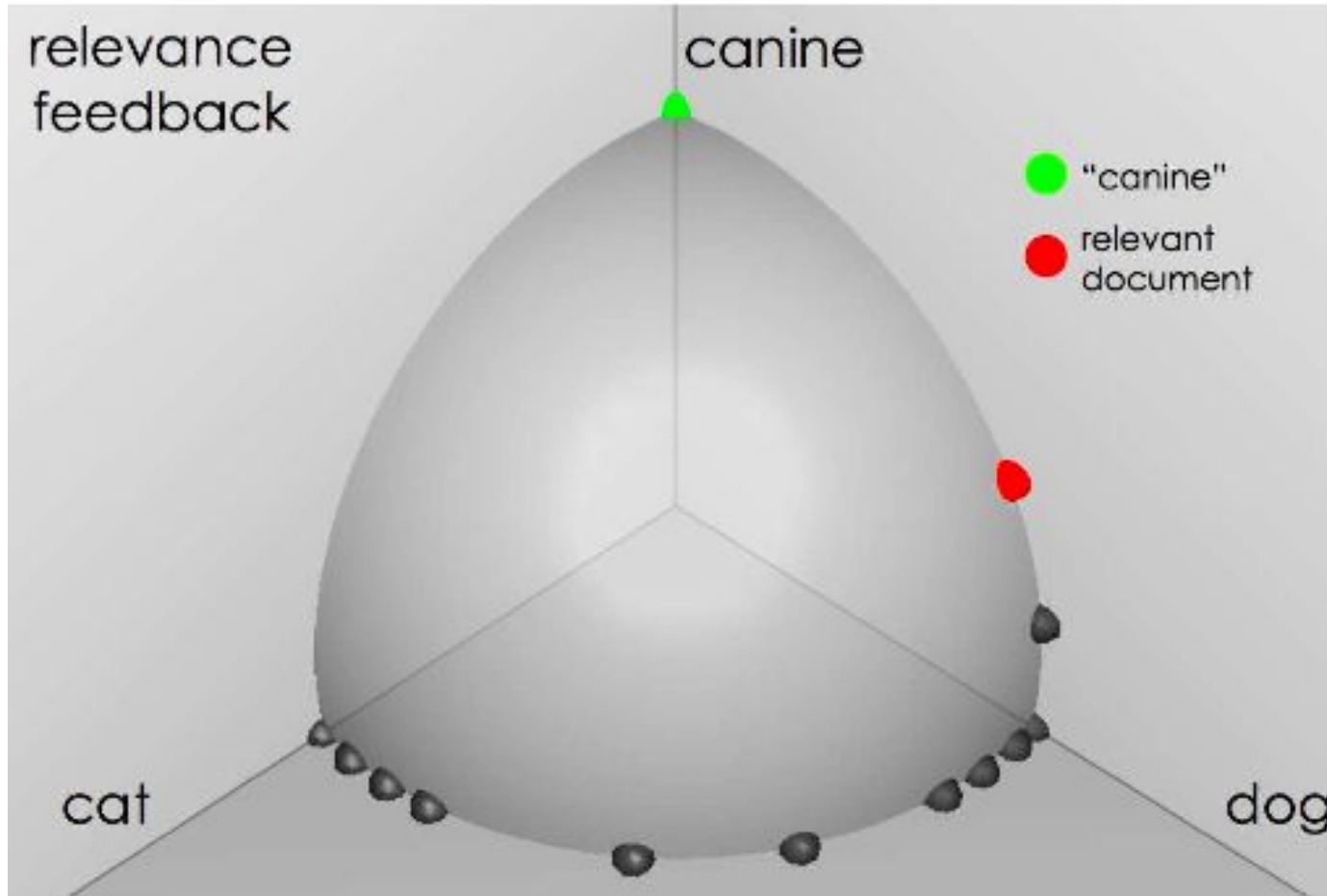
(Khoảng cách góc của văn bản và truy vấn)



Nguồn: [Fernando Díaz]

Ví dụ 6.2. Phản hồi trong không gian vec-tơ⁽³⁾

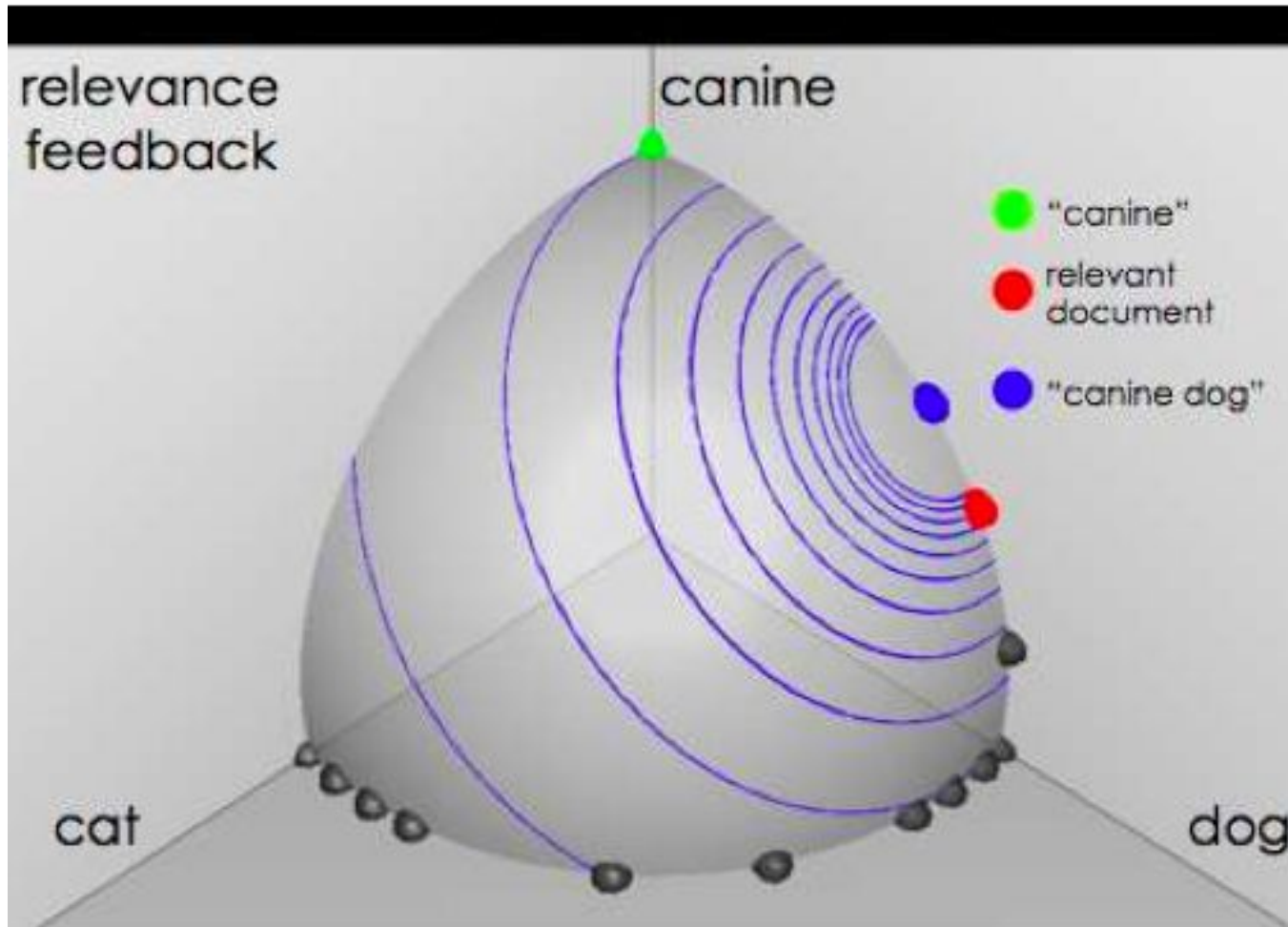
Người dùng chọn văn bản phù hợp



Nguồn: [Fernando Díaz]

Ví dụ 6.2. Phản hồi trong không gian vec-to₍₄₎

Kết quả sau khi tiếp nhận phản hồi



Nguồn: [Fernando Díaz]

Ví dụ 6.3. Phản hồi với dữ liệu văn bản

Truy vấn ban đầu: [new space satellite applications]

Các kết quả cho truy vấn ban đầu: (r = rank)

	r		
+	1	0.539	NASA Hasn't Scrapped Imaging Spectrometer
+	2	0.533	NASA Scratches Environment Gear From Satellite Plan
	3	0.528	Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes
	4	0.526	A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget
	5	0.525	Scientist Who Exposed Global Warming Proposes Satellites for Climate Research
	6	0.524	Report Provides Support for the Critics Of Using Big Satellites to Study Climate
	7	0.516	Arianespace Receives Satellite Launch Pact From Telesat Canada
+	8	0.509	Telecommunications Tale of Two Companies

Sau đó người dùng đánh dấu các văn bản phù hợp (VB có dấu “+”)

Ví dụ 6.3. Phản hồi với dữ liệu văn bản₍₂₎

Câu truy vấn sau khi tiếp nhận phản hồi

2.074 new 30.816 satellite 5.991 nasa 4.196 launch
3.516 instrument 3.004 bundespost 2.790 rocket
2.003 broadcast 0.836 oil 15.106 space 5.660 application
5.196 eos 3.972 aster 3.446 arianespace 2.806 ss
2.053 scientist 1.172 earth 0.646 measure

So với truy vấn ban đầu:

query: [new space satellite applications]

Ví dụ 6.3. Phản hồi với dữ liệu văn bản⁽³⁾

Các kết quả cho truy vấn mới

	r	
*	1	0.513 NASA Scratches Environment Gear From Satellite Plan
*	2	0.500 NASA Hasn't Scrapped Imaging Spectrometer
	3	0.493 When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
	4	0.493 NASA Uses 'Warm' Superconductors For Fast Circuit
*	5	0.492 Telecommunications Tale of Two Companies
	6	0.491 Soviets May Adapt Parts of SS-20 Missile For Commercial Use
	7	0.490 Gaping Gap: Pentagon Lags in Race To Match the Soviets In Rocket Launchers
	8	0.490 Rescue of Satellite By Space Agency To Cost \$90 Million

Cơ sở giải thuật Rocchio

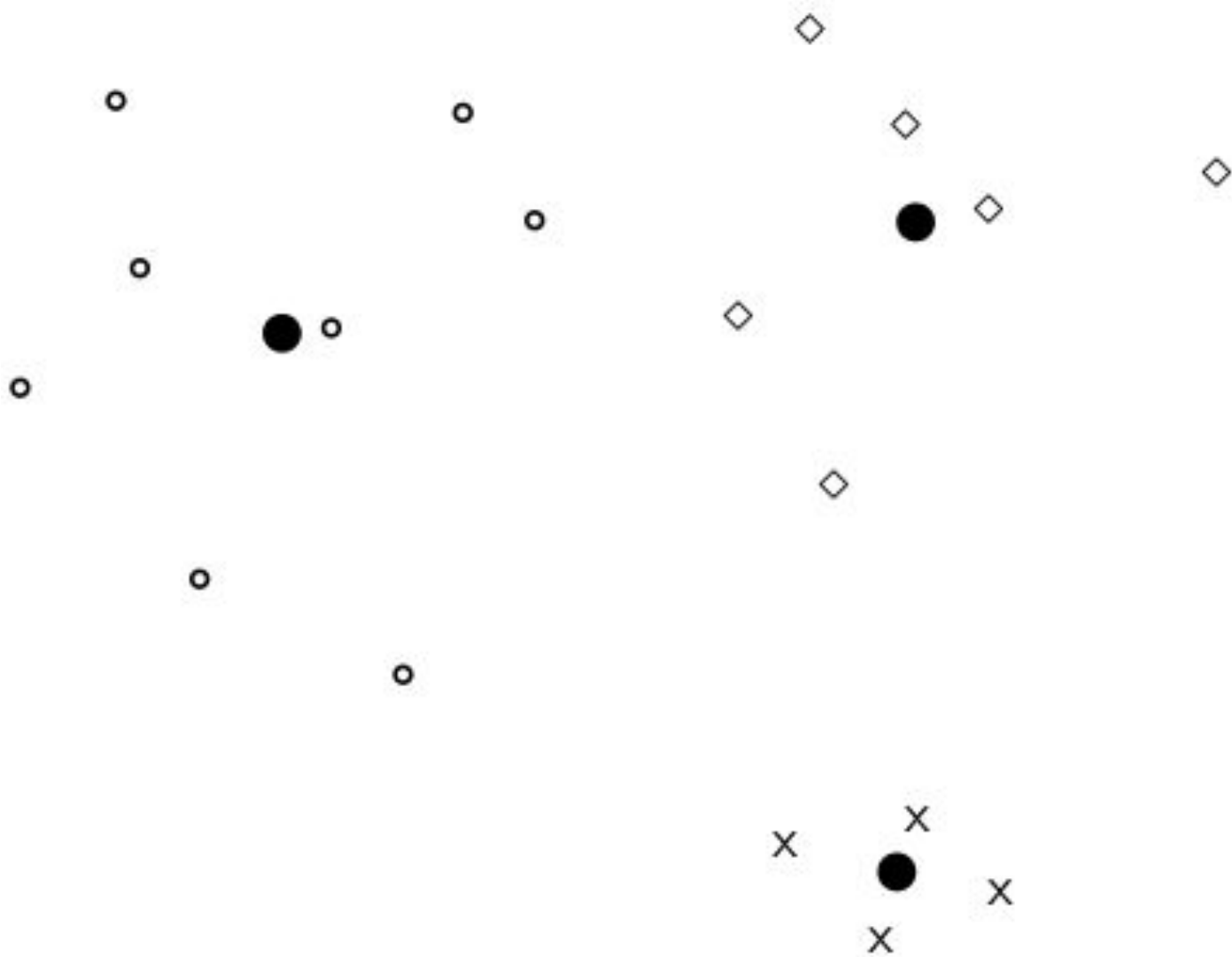
Khái niệm vec-tơ trọng tâm:

- Như đã học trong mô hình không gian vec-tơ chúng ta biểu diễn văn bản như các điểm trong một không gian đa chiều
- Trọng tâm của 1 tập vec-tơ biểu diễn văn bản tương tự như tâm trọng lực của một tập điểm
- Vec-tơ trọng tâm có thể được tính theo công thức:

$$\vec{\mu}(D) = \frac{1}{|D|} \sum_{d \in D} \vec{v}(d)$$

Trong đó D là một tập văn bản; $\vec{v}(d)/\vec{d}$ là biểu diễn vec-tơ của văn bản d .

Ví dụ 6.4. Trọng tâm



Cơ sở giải thuật Rocchio₍₂₎

Vec-tơ truy vấn tối ưu: Với 1 truy vấn q giả sử chúng ta đã biết tập văn bản phù hợp với truy vấn (C_R) và tập văn bản không phù hợp (C_{NR}). Chúng ta muốn tìm vec-tơ \vec{q}_{opt} thỏa mãn:

$$\vec{q}_{opt} = \operatorname{argmax}_{\vec{q}} [\operatorname{sim}(\vec{q}, C_R) - \operatorname{sim}(\vec{q}, C_{NR})]$$

$$\vec{q}_{opt} = \operatorname{argmax}_{\vec{q}} \left[\frac{1}{|C_R|} \sum_{d \in C_R} \operatorname{sim}(\vec{q}, d) - \frac{1}{|C_{NR}|} \sum_{d \in C_{NR}} \operatorname{sim}(\vec{q}, d) \right]$$

- Vec-tơ \vec{q}_{opt} phân tách tối đa các văn bản phù hợp và không phù hợp và được gọi là vec-tơ truy vấn tối ưu. Với độ tương đồng cosine chúng ta có:

$$\vec{q}_{opt} = \frac{1}{|C_R|} \sum_{d_j \in C_R} d_j - \frac{1}{|C_{NR}|} \sum_{d_j \in C_{NR}} d_j$$

$$\vec{q}_{opt} = \mu^{\rightarrow}(C_R) - \mu^{\rightarrow}(C_{NR})$$

Giải thuật Rocchio 1971 (SMART)

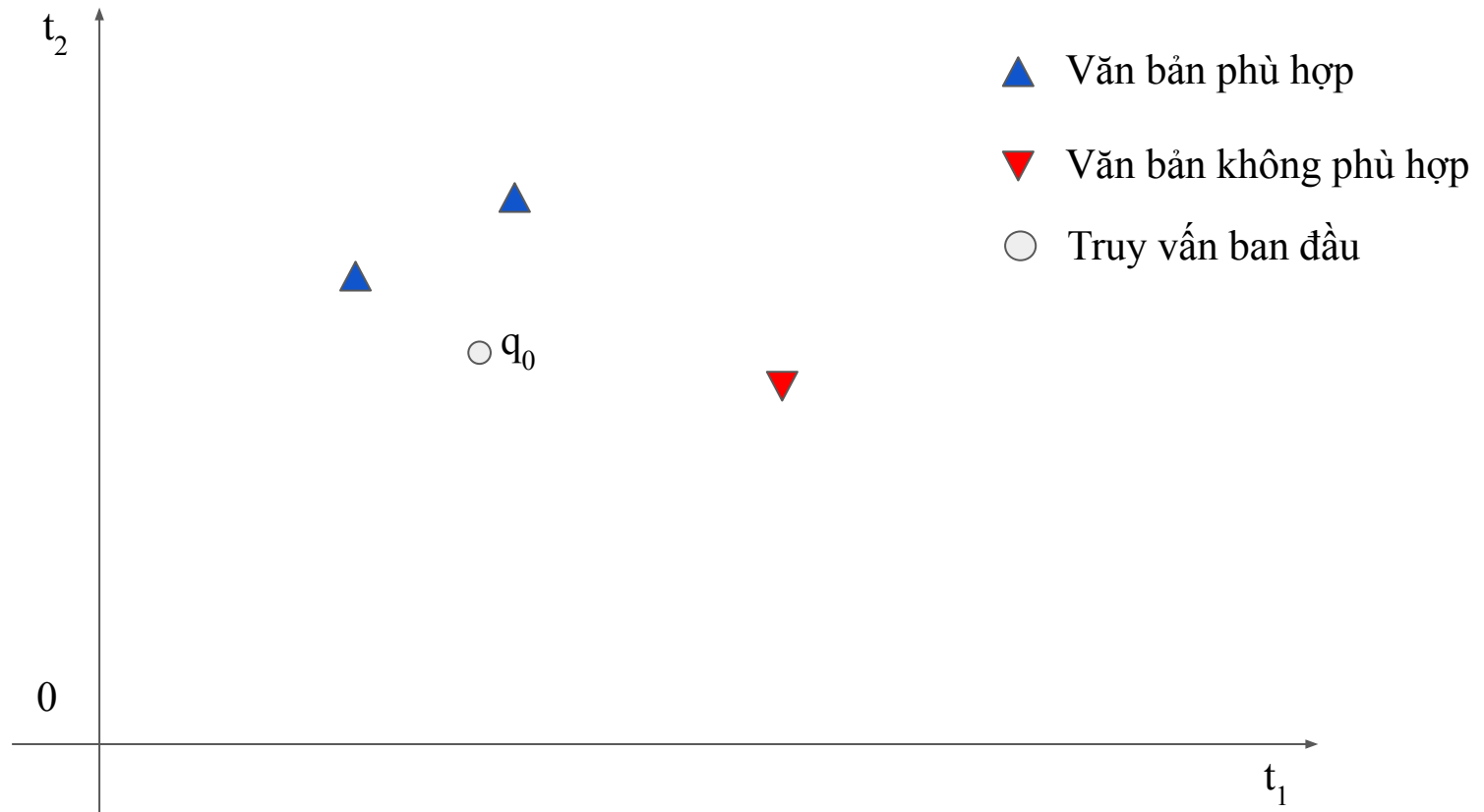
Trong thực tế chúng ta không biết các tập C_R và C_{NR} . Giải thuật Rocchio (trong hệ thống SMART) xác định lại truy vấn như sau:

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_R|} \sum_{\vec{d}_j \in D_R} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{NR}} \vec{d}_j$$

Trong đó \vec{q}_m là vec-tơ truy vấn sau hiệu chỉnh; \vec{q}_0 - vec-tơ truy vấn ban đầu; D_R và D_{NR} - Tập văn bản phù hợp và tập văn bản không phù hợp đã biết; α, β, γ là các tham số tùy chỉnh.

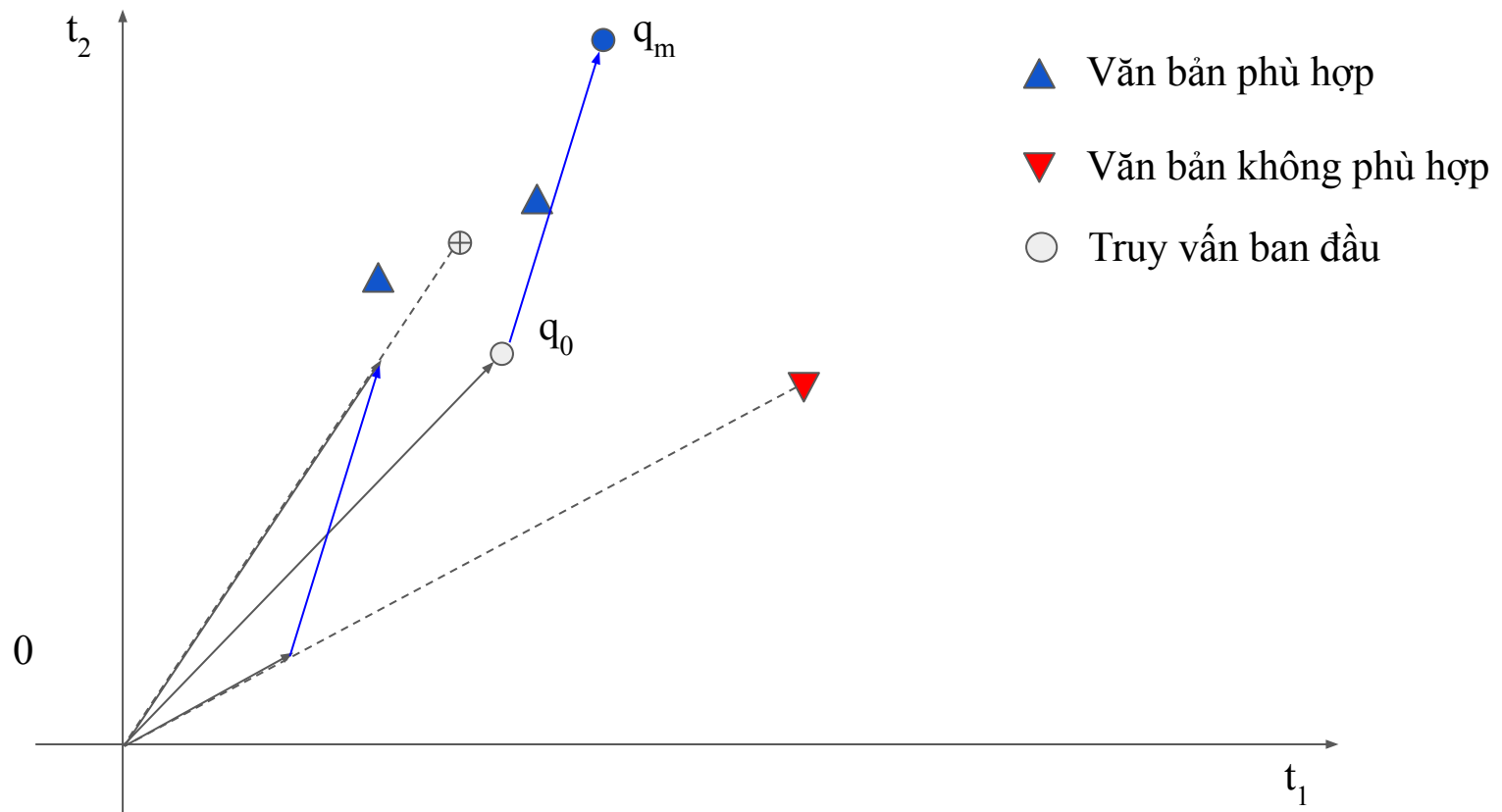
- Truy vấn gốc được di chuyển về phía trọng tâm tập văn bản phù hợp và rời xa trọng tâm tập văn bản không phù hợp.
- Trọng số âm được gán lại bằng 0 (vô nghĩa trong VSM).
- Các kết quả phù hợp thường có trọng số cao hơn ($\beta > \gamma$).

Ví dụ 6.5. Minh họa giải thuật Rocchio



Yêu cầu: Tìm câu truy vấn sau hiệu chỉnh theo giải thuật Rocchio, sử dụng các tham số $\alpha = 1$, $\beta = 0.75$, $\gamma = 0.25$?

Ví dụ 6.5. Minh họa giải thuật Rocchio₍₂₎



Yêu cầu: Tìm câu truy vấn sau hiệu chỉnh theo giải thuật Rocchio, sử dụng các tham số $\alpha = 1$, $\beta = 0.75$, $\gamma = 0.25$?

Phản hồi kết quả phù hợp trên Web

- Một số máy tìm kiếm cung cấp tính năng tìm kết quả tương tự/liên quan (một dạng phản hồi tối giản)
- Excite ban đầu có phản hồi phù hợp đầy đủ, nhưng sau đó loại bỏ do ít được sử dụng
 - Chỉ có khoảng 4% phiên tìm kiếm có sử dụng tính năng phản hồi kết quả phù hợp
 - Được cung cấp trong liên kết “**More like this**” (Thêm kết quả tương tự) cho mỗi kết quả tìm kiếm.
 - Tỷ lệ trường hợp kết quả tìm kiếm được đánh giá tốt hơn sau khi xử lý phản hồi $\approx 2/3$.
 - [Spink et al. 2000]

Người dùng có thể ưa thích viết câu truy vấn mới hơn phản hồi các kết quả phù hợp. Ngoài ra hệ thống cũng có thể cung cấp các gợi ý.

Giả lập phản hồi

- Tiến trình ẩn đối với người dùng, không yêu cầu người dùng đánh giá tính phù hợp
- Được thực hiện trong tiến trình xử lý truy vấn:
 - Tìm danh sách kết quả cho truy vấn ban đầu từ người dùng
 - Coi top k văn bản như những kết quả phù hợp
 - Thực hiện xử lý phản hồi (ví dụ, Rocchio)
 - Trả về danh sách kết quả sau khi xử lý phản hồi cho người dùng
- Hoạt động tốt trong trường hợp tổng quát
- Nhưng có thể rất không tốt với một số truy vấn
- Truy vấn có thể bị biến dạng sau các vòng phản hồi

Giả lập phản hồi ở TREC4

Hệ thống SMART

- Số lượng văn bản phù hợp trong top 100 đối với 50 truy vấn (tổng số văn bản là 5000) được tổng hợp trong bản:

Phương pháp	#văn bản phù hợp	Phương pháp	#văn bản phù hợp
Inc.ltc	3210	Lnu.ltu	3709
Inc.ltc-PsRF	3634	Lnu.ltu-PsRF	4350

- Các kết quả so sánh hai sơ đồ tính điểm với 2 đại lượng chuẩn hóa độ dài khác nhau trong trường hợp có giả lập phản hồi (PsRF) và không có giả lập phản hồi.
- Phương pháp giả lập phản hồi trong thí nghiệm chỉ thêm 20 từ vào truy vấn (Rocchio sẽ thêm nhiều hơn).
- Kết quả thí nghiệm cho thấy mô hình có sử dụng giả lập phản hồi cho kết quả tốt hơn trong trường hợp tổng quát.

Nội dung

1. Phản hồi kết quả phù hợp và giải thuật Rocchio
2. Dữ liệu hành vi người dùng
3. Đánh giá kết quả có phản hồi
4. Mở rộng truy vấn dựa trên từ điển
5. Tự động xác định các từ liên quan
6. Phân tích ngữ nghĩa ẩn

Dữ liệu nhấn chuột

- Trong máy tìm kiếm Web danh sách kết quả thường chỉ chứa trích đoạn và các thông tin tổng quan về các tài liệu.
- Khi xem danh sách kết quả người dùng có thể mở kết quả tìm kiếm (nhấn chuột vào liên kết đến tài liệu gốc).
- Thao tác nhấn chuột thể hiện sự quan tâm của người dùng đối với kết quả tìm kiếm.
 - Có thể thu thập một lượng lớn dữ liệu nhấn chuột mà không làm ảnh hưởng tới người dùng.
 - *Máy tìm kiếm có thể khai thác dữ liệu thu thập được để cải tiến phương pháp tìm kiếm nhằm đưa ra những kết quả tốt hơn.*

Kết quả được mở xem có có phải là kết quả phù hợp hay không?

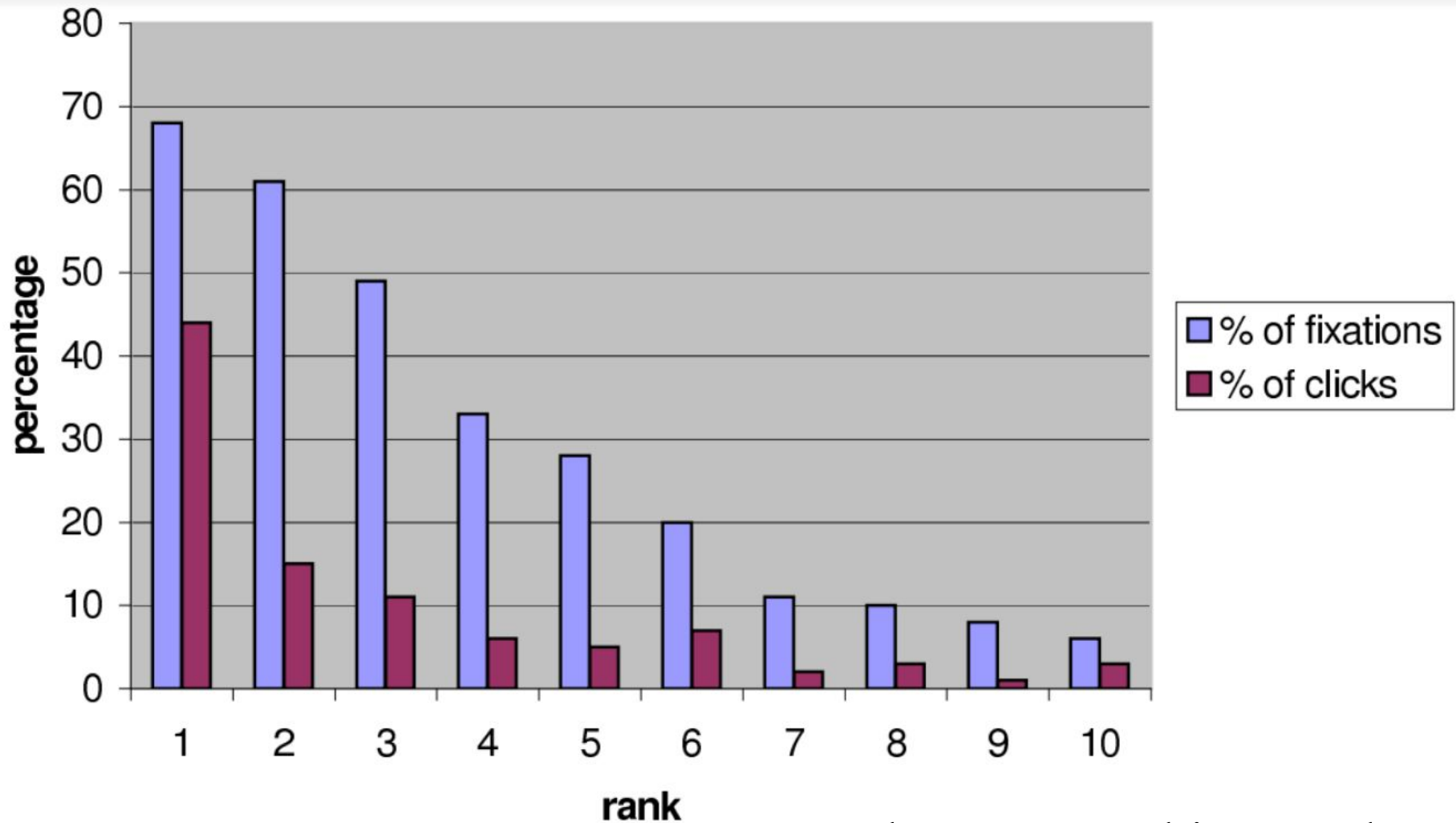
Hướng nhìn của người dùng

- Các thiết bị quan sát hướng nhìn của người dùng (quan sát hành vi thị giác) cho phép xác định vùng màn hình đang thu hút sự quan tâm của người dùng
- Các hành vi thị giác có thể được chia thành 4 loại: Nhìn chăm chú, chuyển tiêu điểm đột ngột, co giãn đồng tử, lướt/dò tìm
- Nhìn chăm chú (cố định) vào một vùng cụ thể trên màn hình trong khoảng 200-300 ms
 - Thường gắn liền với tiếp nhận và xử lý thông tin;
 - Được lưu ý nhất trong biểu diễn hành vi người dùng.

Hướng nhìn của người dùng₍₂₎

- Các thí nghiệm quan sát hướng nhìn đã cho thấy người dùng đọc danh sách kết quả theo thứ tự từ trên xuống dưới.
- Người dùng kiểm tra 2 kết quả đầu tiên ngay lập trong giới hạn 3 lần đọc kết quả (nhìn chăm chú) đầu tiên.
- Tiếp theo người dùng có xu hướng đọc các kết quả khác ngay trong trang đầu tiên trước khi sử dụng các thanh điều khiển để xem các kết quả khác.

Hướng nhìn của người dùng₍₃₎



[Thorsten Joachims et al]

Số lần nhấn chuột & sự quan tâm

- Nhấn chuột không trực tiếp thể hiện sự phù hợp của kết quả tìm kiếm.
- Nhấn chuột thể hiện **sự quan tâm** của người dùng là giả thuyết được sử dụng phổ biến hơn
 - Ví dụ, người dùng có thể xem một vài kết quả rồi quyết định nhấn chuột vào một kết quả trong số đó, bỏ qua các kết quả khác
 - Điều đó chứng tỏ người dùng quan tâm đến kết quả được chọn hơn các kết quả khác
- Sự quan tâm được đánh giá dựa trên 2 quan sát:
 - Kết quả được người dùng mở xem
 - Kết quả người dùng đã thấy nhưng không mở xem

Nhấn chuột trong phạm vi 1 truy vấn

- Để đánh giá sự quan tâm của người dùng dựa trên dữ liệu nhấn chuột chúng ta sử dụng các ký hiệu sau:
 - r_k là văn bản được xếp hạng thứ k theo giá trị trạng thái tìm kiếm $RSV(d, q)$ trong danh sách kết quả.
 - r_1, r_2, r_3, \dots tương ứng là những văn bản đầu tiên trong danh sách
 - $*r_k$ là ký hiệu người dùng đã nhấn chuột vào kết quả r_k .
 - $r_k > r_n$ thể hiện r_k được quan tâm hơn r_n .

Chúng ta cần so sánh mức độ được quan tâm tương đối giữa các kết quả.

Nhấn chuột trong phạm vi 1 truy vấn₍₂₎

- Chúng ta xét một danh sách kết quả như sau:

$$r_1 \ r_2 \ *r_3 \ r_4 \ *r_5 \ r_6 \ r_7 \ r_8 \ r_9 \ *r_{10}$$

- Dữ liệu này không cho chúng ta khẳng định chắc chắn về tính phù hợp của các kết quả r_3 , r_5 , và r_{10} , tuy nhiên chúng ta có thể đánh giá sự quan tâm tương đối của người dùng.
- Hai *luật quyết định* thường được sử dụng:
 - **Bỏ qua phần trên:** Nếu $*r_k$ với k nhỏ nhất thì $r_k > r_{k-}$ (người dùng quan tâm r_k hơn tất cả các kết quả đứng trước k).
 - **Bỏ qua kẻ trước:** Nếu $*r_k$ và r_{k-1} với k nhỏ nhất thì $r_k > r_{k-1}$

Nhấn chuột trong phạm vi 1 truy vấn₍₃₎

Với kết quả truy vấn đang được phân tích:

$$r_1 \ r_2 \ *r_3 \ r_4 \ *r_5 \ r_6 \ r_7 \ r_8 \ r_9 \ *r_{10}$$

- Nếu áp dụng luật bỏ qua phần trên, thì chúng ta có:
 - $r_3 > r_2; r_3 > r_1$
- và với luật bỏ qua kẻ trước chúng ta có:
 - $r_3 > r_2$
- *=> Có thể thấy luật bỏ qua phần trên cho nhiều quan hệ hơn luật bỏ qua kẻ trước*

Nhấn chuột trong phạm vi chuỗi truy vấn

- Trong tiến trình xử lý một vấn đề thông tin người dùng có thể gửi nhiều truy vấn nối tiếp nhau tạo thành chuỗi truy vấn.
- Để minh họa chúng ta xét 1 chuỗi 2 truy vấn với 2 danh sách kết quả tìm kiếm như sau:

$r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9 r_{10}$

$s_1 *s_2 s_3 s_4 *s_5 s_6 s_7 s_8 s_9 s_{10}$

- Trong ví dụ này người dùng chỉ nhấn chuột vào kết quả thứ 2 và thứ 5 trong danh sách kết quả thứ 2.

Nhấn chuột trong phạm vi chuỗi truy vấn₍₂₎

- Hai luật quyết định thường được sử dụng trong trường hợp này là:
 - **Top-1:** Nếu $\exists s_k \mid *s_k$ thì $s_j > r_1$, với $j \leq 10$
 - **Top-2:** Nếu $\exists s_k \mid *s_k$ thì $s_j > r_1$ và $s_j > r_2$, với $j \leq 10$
- Với dữ liệu minh họa đang được phân tích, chúng ta có thể suy ra các quan hệ sau:
 - Theo Top-1:
 - $s_1 > r_1; s_2 > r_1; s_3 > r_1; \dots; s_{10} > r_1$
 - Theo Top-2 thì:
 - $s_1 > r_1; s_2 > r_2, s_3 > r_1; \dots; s_{10} > r_1$
 - $s_1 > r_2; s_2 > r_2, s_3 > r_2; \dots; s_{10} > r_2$

Hành vi nhấn chuột và tính tương đồng

- Hành vi nhấn chuột cũng được coi như 1 hình thức phản hồi của người dùng, thể hiện sự quan tâm tương đối của người dùng đối với kết quả tìm kiếm.
- Có thể suy diễn quan hệ so sánh mức độ được quan tâm giữa các văn bản từ dữ liệu nhấn chuột
- Các nghiên cứu cho thấy xếp hạng văn bản theo mức độ quan tâm có tính nhất quán cao so với xếp hạng văn bản theo mức tương đồng
- Quan hệ so sánh mức độ quan tâm giữa các văn bản là một dạng dữ liệu nhiễu nhưng có thể được sử dụng để huấn luyện mô hình xếp hạng dựa trên học máy.

Nội dung

1. Phản hồi kết quả phù hợp và giải thuật Rocchio
2. Dữ liệu hành vi người dùng
3. Đánh giá kết quả có phản hồi
4. Mở rộng truy vấn dựa trên từ điển
5. Tự động xác định các từ liên quan
6. Phân tích ngữ nghĩa ẩn

Đánh giá kết quả có phản hồi

- Lựa chọn một đại lượng đo đã biết, ví dụ, độ chính xác ở top 10: $P@10$
- Tính $P@10$ cho truy vấn ban đầu q_0
- Tính $P@10$ cho truy vấn được hiệu chỉnh sau khi tiếp nhận phản hồi q_1
- Trong hầu hết các trường hợp q_1 cho kết quả tốt hơn đáng kể so với q_0

Cách đánh giá này có đủ tin cậy hay không?

Đánh giá kết quả có phản hồi₍₂₎

- Đánh giá phải được thực hiện trên bộ dữ liệu hoàn toàn mới với các văn bản chưa được đánh giá bởi người dùng
 - Các nghiên cứu cho thấy phản hồi vẫn có kết quả tốt nếu đánh giá theo cách này
- Đồng thời thực nghiệm cho thấy, một vòng phản hồi thường rất hữu ích, nhưng tăng số vòng phản hồi không làm tăng đáng kể chất lượng tìm kiếm.

Phương pháp đánh giá khác

- Cũng có thể đánh giá tính hiệu quả của việc áp dụng cơ chế phản hồi bằng cách so sánh với các phương pháp tương đương khác
 - Ví dụ: Gợi ý truy vấn hoặc người dùng tự viết lại câu truy vấn
 - Người dùng có thể thích hiệu chỉnh, viết lại câu truy vấn hơn so với đánh giá tính phù hợp của các văn bản

Các hạn chế của kỹ thuật phản hồi

- Xử lý phản hồi có chi phí cao
 - Phản hồi tạo ra các vec-tơ truy vấn dài
 - Tốn nhiều chi phí để xử lý các truy vấn dài
- Người dùng không sẵn sàng cung cấp các phản hồi
- Có thể khó hiểu vì sao một văn bản cụ thể được trả về sau khi áp dụng phản hồi
 - Xê dịch chủ đề tìm kiếm.

Nội dung

1. Phản hồi kết quả phù hợp và giải thuật Rocchio
2. Dữ liệu hành vi người dùng
3. Đánh giá kết quả có phản hồi
4. Mở rộng truy vấn dựa trên từ điển
5. Tự động xác định các từ liên quan
6. Phân tích ngữ nghĩa ẩn

Mở rộng truy vấn dựa trên từ điển

- Một kỹ thuật khác nhằm nâng cao chất lượng tìm kiếm
- Ý tưởng cơ bản:
 - Bổ xung thêm vào câu truy vấn các từ đồng nghĩa với từ truy vấn.
 - Ví dụ: Ô tô => Xe hơi
 - Thường làm tăng độ đầy đủ nhưng có thể làm giảm độ chính xác.
 - Các từ đồng nghĩa được xác định dựa trên từ điển đồng nghĩa.
- Chúng ta sẽ tìm hiểu 2 dạng từ điển đồng nghĩa
 - Được tạo thủ công và: Được duy trì bởi nhóm soạn thảo, ví dụ PubMed
 - Thường được sử dụng cho các máy tìm kiếm chuyên dụng trong các lĩnh vực khoa học và kỹ thuật
 - Chi phí tạo và duy trì từ điển theo thời gian rất cao
 - Được tạo tự động: Được tổng hợp từ bộ dữ liệu văn bản, ví dụ dựa trên thống kê đồng xuất hiện.

Ví dụ 6.6. Mở rộng truy vấn trong PubMed

NCBI Resources How To

MeSH MeSH flu Search

Limits Advanced

Search Details

Query Translation:

"influenza, human"[MeSH Terms] OR flu[Text Word]

Search URL

Nội dung

1. Phản hồi kết quả phù hợp và giải thuật Rocchio
2. Dữ liệu hành vi người dùng
3. Đánh giá kết quả có phản hồi
4. Mở rộng truy vấn dựa trên từ điển
5. Tự động xác định các từ liên quan
6. Phân tích ngữ nghĩa ẩn

Mở rộng dựa trên lịch sử truy vấn

- Máy tìm kiếm có một lượng lớn dữ liệu lịch sử truy vấn.
- *Mở rộng truy vấn dựa trên lịch sử truy vấn có hiệu ứng tương tự như sử dụng từ điển đồng nghĩa.*
- Ví dụ 1: Sau khi tìm [herbs], người dùng thường tiếp tục tìm [herbal remedies].
 - => có thể sử dụng “herbal remedies” để mở rộng “herb”.
- Ví dụ 2: Người dùng tìm [flower pix] thường xuyên mở liên kết photobucket.com/flower, đồng thời người dùng tìm [flower clipart] thường xuyên bấm vào cùng URL.
 - => có thể sử dụng “flower clipart” để mở rộng “flower pix” và ngược lại

Các cặp truy vấn có liên quan có thể được xác định dựa trên các đặc điểm trung có trong lịch sử tìm kiếm của người dùng.

Tổng hợp từ điển đồng nghĩa

- Mục đích: Tự động tạo từ điển dựa trên phân bố từ trong bộ dữ liệu văn bản.
- Khái niệm nền tảng: Độ tương đồng giữa hai từ
- **Định nghĩa 1:** Hai từ được coi là tương đồng (ý nghĩa) nếu cùng xuất hiện với những từ giống nhau
 - ô tô \approx xe máy, bởi vì cả hai cùng xuất hiện với từ con đường, xăng, và bằng lái.
- **Định nghĩa 2:** Hai từ được coi là tương đồng nếu cùng có mối quan hệ ngữ pháp giống nhau với cùng một nhóm từ.
 - Bạn có thể thu hoạch, gọt vỏ, ăn, ép nước, v.v.. táo và lê, vì vậy táo và lê là các từ tương tự.

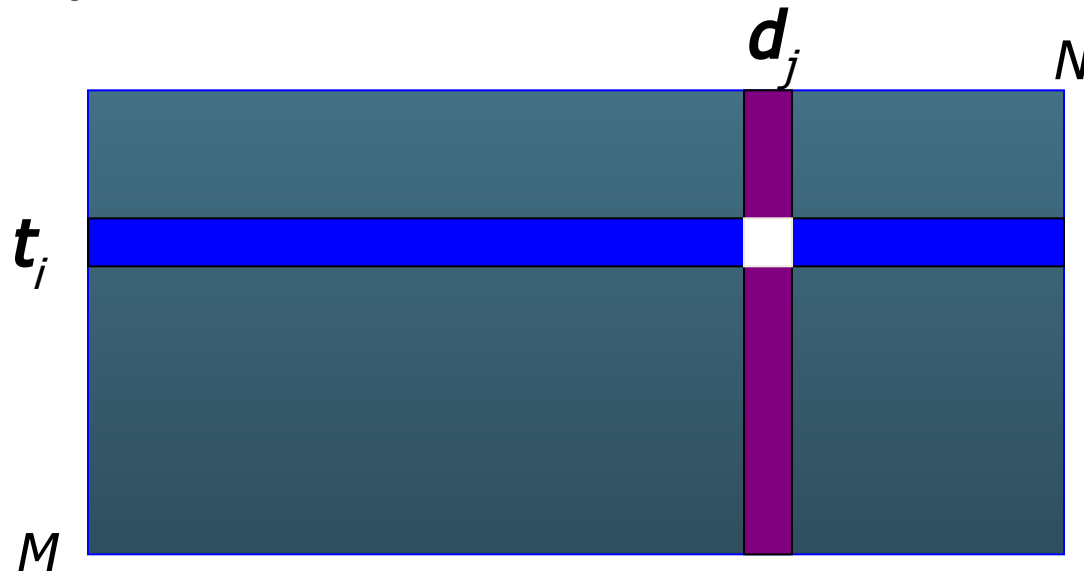
Xử lý đồng xuất hiện đơn giản hơn, quan hệ ngữ pháp yêu cầu các xử lý ngôn ngữ phức tạp hơn nhưng chính xác hơn.

Từ điển đồng nghĩa dựa trên đồng xuất hiện

- Cách đơn giản nhất để tổng hợp từ điển đồng nghĩa dựa trên độ tương đồng giữa các cặp từ

$C = AA^T$ trong đó A là ma trận từ-văn bản.

w_{ij} = trọng số (sau chuẩn hóa) cho (t_i, d_j)



Các giá trị của C là gì nếu A là ma trận đánh dấu (0/1)?

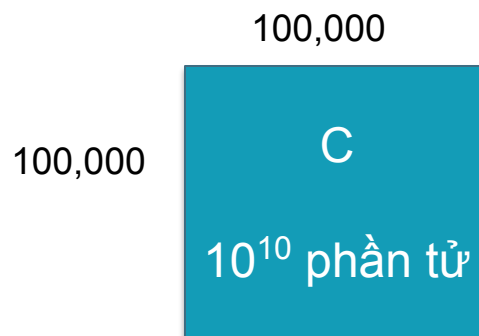
- Cho từ t_i , chọn các từ có giá trị lớn trong C

Ví dụ 6.7. Từ điển đồng nghĩa dựa trên đồng xuất hiện

Từ	Láng giềng gần nhất
absolutely bottomed captivating doghouse makeup mediating keeping lithographs pathogens senses	absurd whatsoever totally exactly nothing dip copper drops topped slide trimmed shimmer stunningly superbly plucky witty dog porch crawling beside downstairs repellent lotion glossy sunscreen skin gel reconciliation negotiate case conciliation hoping bring wiping could some would drawings Picasso Dali sculptures Gauguin toxins bacteria organisms bacterial parasite grasp psyche truly clumsy naive innate

Các vấn đề đối với tổng hợp từ điển

- Tính thừa của dữ liệu:



- Tính đa nghĩa của từ có thể dẫn tới các liên kết bất thường
 - “planet earth facts” -> “planet earth soil ground facts”
- Do từ có thể đồng xuất hiện vì bất kỳ lý do gì, truy vấn mở rộng có thể không trả về nhiều văn bản mới
 - => cần giới hạn khái niệm đồng xuất hiện: Từ xuất hiện trong cùng ngữ cảnh, ví dụ trong phạm vi một cửa sổ chữ không phải toàn bộ văn bản.

Kỹ thuật giảm chiều không gian vec-tơ

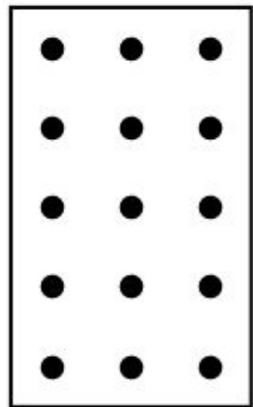
- Ma trận đồng xuất hiện có số chiều rất lớn và chứa rất nhiều phần tử, tuy nhiên có rất nhiều phần tử có giá trị = 0 (*ma trận thưa*).
- Giải pháp: Biến đổi không gian đồng xuất hiện rất lớn nhưng thưa về không gian với số chiều nhỏ nhưng vẫn bảo toàn tính chất tương đồng giữa các biểu diễn của những từ có ý nghĩa tương đồng (word embedding).

Qua đó giải quyết vấn đề từ đồng nghĩa, nâng cao độ đầy đủ và có thể cả độ chính xác trong 1 số trường hợp.

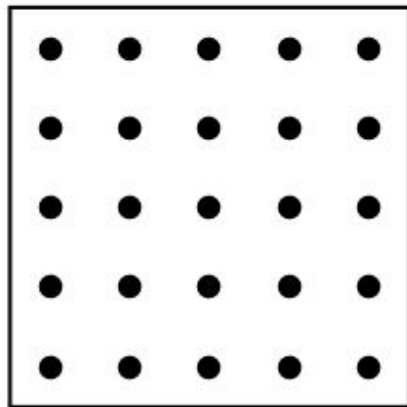
Nội dung

1. Phản hồi kết quả phù hợp và giải thuật Rocchio
2. Dữ liệu hành vi người dùng
3. Đánh giá kết quả có phản hồi
4. Từ điển đồng nghĩa được tạo thủ công
5. Từ điển đồng nghĩa được tổng hợp tự động
6. Phân tích ngữ nghĩa ẩn

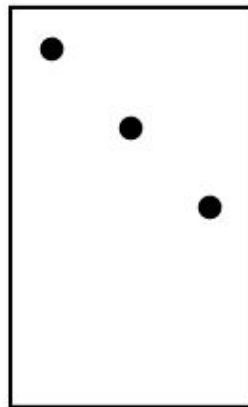
Phân tích đơn trị



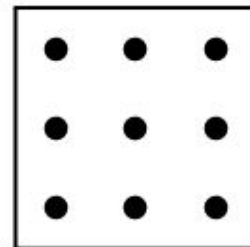
A
(M x N)



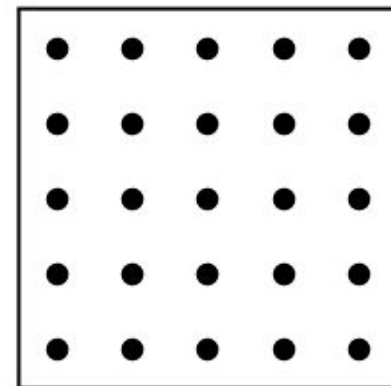
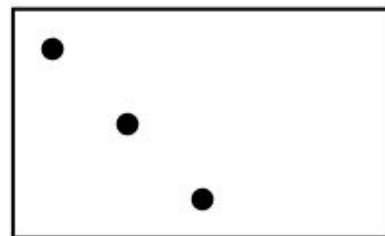
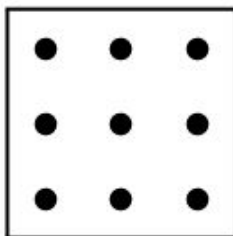
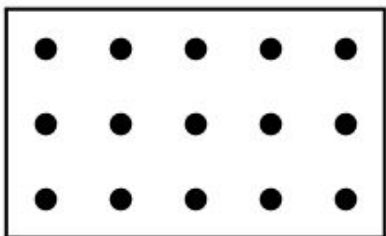
U
(M x M)



Σ
(M x N)



V^T
(N x N)



Ví dụ 6.8. Phân tích đơn trị trong R

A	d ₁	d ₂	d ₃	d ₄	d ₅	d ₆
t ₁	1	0	1	0	0	0
t ₂	0	1	0	0	0	0
t ₃	1	1	0	0	0	0
t ₄	1	0	0	1	1	0
t ₅	0	0	0	1	0	1

$M = 5, N = 6$

Cột thứ 6 của Σ và dòng thứ 6 của V^T chỉ chứa giá trị 0 và đã được lược bỏ (biểu diễn giản lược của SVD).

U
usv = svd(A)



```
> round(usv$u, 3)
      [,1] [,2] [,3] [,4] [,5]
[1,] 0.440 -0.296 -0.569 0.577 -0.246
[2,] 0.129 -0.331 0.587 0.000 -0.727
[3,] 0.476 -0.511 0.368 0.000 0.614
[4,] 0.703 0.351 -0.155 -0.577 -0.160
[5,] 0.263 0.647 0.415 0.577 0.087
```

Σ

```
> round(diag(usv$d), 3)
      [,1] [,2] [,3] [,4] [,5]
[1,] 2.163 0.000 0.000 0 0.000
[2,] 0.000 1.594 0.000 0 0.000
[3,] 0.000 0.000 1.275 0 0.000
[4,] 0.000 0.000 0.000 1 0.000
[5,] 0.000 0.000 0.000 0 0.394
```

V^T

```
> round(t(usv$v), 3)
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 0.749 0.280 0.204 0.447 0.325 0.121
[2,] -0.286 -0.528 -0.186 0.626 0.220 0.406
[3,] -0.280 0.749 -0.447 0.204 -0.121 0.325
[4,] 0.000 0.000 0.577 0.000 -0.577 0.577
[5,] 0.528 -0.286 -0.626 -0.186 -0.406 0.220
```

Các thành phần trong phân tích đơn trị₍₂₎

$$AA^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V\Sigma U^T) = U\Sigma^2 U^T$$

Các giá trị riêng $\lambda_1 \dots \lambda_r$ của AA^T là các giá trị riêng của $A^T A$

$\sigma_i = \sqrt{\lambda_i}$; $\lambda_i \geq \lambda_{i+1}$, $\Sigma_{ii} = \sigma_i$ với $1 \leq i \leq r$, và $= 0$ nếu ngược lại.

Các cột của U là các vec-tơ riêng vuông góc của AA^T

Tương tự, $A^T A = (U\Sigma V^T)^T(U\Sigma V^T) = (V\Sigma U^T)(U\Sigma V^T) = V\Sigma^2 V^T$

Các cột của V là các vec-tơ riêng vuông góc của $A^T A$

Ví dụ 6.9. Chi tiết các thành phần SVD

Sử dụng ma trận A như trong ví dụ 6.7.

```
> A %*% t(A)
      [,1] [,2] [,3] [,4] [,5]
[1,]  2    0    1    1    0
[2,]  0    1    1    0    0
[3,]  1    1    2    1    0
[4,]  1    0    1    3    1
[5,]  0    0    0    1    2
> ev <- eigen(A %*% t(A))
> values <- ev$values
> round(sqrt(values), 3)
[1] 2.163 1.594 1.275 1.000 0.394
> vectors = ev$vectors
> round(vectors, 3)
      [,1] [,2] [,3] [,4] [,5]
[1,] -0.440 0.296 -0.569 0.577 0.246
[2,] -0.129 0.331 0.587 0.000 0.727
[3,] -0.476 0.511 0.368 0.000 -0.614
[4,] -0.703 -0.351 -0.155 -0.577 0.160
[5,] -0.263 -0.647 0.415 0.577 -0.087
```

```
> t(A) %*% A
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]  3    1    1    1    1    0
[2,]  1    2    0    0    0    0
[3,]  1    0    1    0    0    0
[4,]  1    0    0    2    1    1
[5,]  1    0    0    1    1    0
[6,]  0    0    0    1    0    1
> ev <- eigen(t(A) %*% A)
> values <- ev$values
> round(sqrt(values), 3)
[1] 2.163 1.594 1.275 1.000 0.394 0.000
> vectors = ev$vectors
> round(vectors, 3)
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] -0.749 0.286 -0.280 0.000 0.528 0.000
[2,] -0.280 0.528 0.749 0.000 -0.286 0.000
[3,] -0.204 0.186 -0.447 0.577 -0.626 0.000
[4,] -0.447 -0.626 0.204 0.000 -0.186 -0.577
[5,] -0.325 -0.220 -0.121 -0.577 -0.406 0.577
[6,] -0.121 -0.406 0.325 0.577 0.220 0.577
```


Ví dụ 6.9. Chi tiết các thành phần $SVD_{(2)}$

Các giá trị riêng, vec-tơ riêng và các thành phần SVD

```
> ev <- eigen(A %*% t(A))
> values <- ev$values
> round(sqrt(values), 3)
[1] 2.163 1.594 1.275 1.000 0.394
> vectors = ev$vectors
> round(vectors, 3)
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.440	0.296	-0.569	0.577	0.246
[2,]	-0.129	0.331	0.587	0.000	0.727
[3,]	-0.476	0.511	0.368	0.000	-0.614
[4,]	-0.703	-0.351	-0.155	-0.577	0.160
[5,]	-0.263	-0.647	0.415	0.577	-0.087

```
> round(usv$u, 3)
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.440	-0.296	-0.569	0.577	-0.246
[2,]	0.129	-0.331	0.587	0.000	-0.727
[3,]	0.476	-0.511	0.368	0.000	0.614
[4,]	0.703	0.351	-0.155	-0.577	-0.160
[5,]	0.263	0.647	0.415	0.577	0.087

```
> ev <- eigen(t(A) %*% A)
> values <- ev$values
> round(sqrt(values), 3)
[1] 2.163 1.594 1.275 1.000 0.394 0.000
> vectors = ev$vectors
> round(vectors, 3)
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
[1,]	-0.749	0.286	-0.280	0.000	0.528	0.000
[2,]	-0.280	0.528	0.749	0.000	-0.286	0.000
[3,]	-0.204	0.186	-0.447	0.577	-0.626	0.000
[4,]	-0.447	-0.626	0.204	0.000	-0.186	-0.577
[5,]	-0.325	-0.220	-0.121	-0.577	-0.406	0.577
[6,]	-0.121	-0.406	0.325	0.577	0.220	0.577

```
> round(usv$v, 3)
```

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	0.749	-0.286	-0.280	0.000	0.528
[2,]	0.280	-0.528	0.749	0.000	-0.286
[3,]	0.204	-0.186	-0.447	0.577	-0.626
[4,]	0.447	0.626	0.204	0.000	-0.186
[5,]	0.325	0.220	-0.121	-0.577	-0.406
[6,]	0.121	0.406	0.325	0.577	0.220

Được gần lược

Đảo dấu các cột tương ứng trong U và V không làm thay đổi kết quả phân tích.

Xấp xỉ hạng nhỏ

- Vấn đề xấp xỉ: Cho ma trận A và số nguyên dương k , chúng ta cần tìm A_k có hạng không lớn hơn k sao cho

$\|A - A_k\|_F$ đạt giá trị cực tiểu.

Đặt $X = A - A_k$, $\|X\|_F = \sqrt{\sum_{ij} X_{ij}^2}$ là Frobenius norm của X

- Chúng ta có thể tìm A_k dựa trên SVD.

Thông thường chúng ta muốn có $k \ll r$

Giản lược SVD

- Nếu chúng ta chỉ giữ k đơn trị có giá trị lớn nhất trong Σ và thiết lập phần còn lại bằng 0, thì các dòng và cột tương ứng trong U và V^T không còn ảnh hưởng đến tích các ma trận.
- Vì vậy có thể giản lược các dòng và cột 0 trong Σ và các dòng và cột tương ứng trong U và V^T . Tương tự như chúng ta không có thể lược bỏ phần ma trận được tô màu.
 - Kích thước các ma trận đã giản lược: $\Sigma (k \times k)$, $U (M \times k)$, $V^T (k \times N)$
 - Giản lược biểu diễn SVD làm giảm dung lượng bộ nhớ cần sử dụng để lưu các thành phần \Rightarrow Thuận tiện hơn để triển khai ứng dụng.

$$\underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{A_k} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_U \underbrace{\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{V^T}$$

Đánh chỉ mục ngữ nghĩa ẩn

Latent Semantic Indexing (LSI). Các bước:

- Tính xấp xỉ hạng nhỏ của ma trận từ-văn bản (k thường được lựa chọn trong khoảng 100-300).
- Ánh xạ các văn bản và truy vấn vào một không gian với số chiều nhỏ (không gian ngữ nghĩa ẩn).
- Tính độ tương đồng dựa trên các biểu diễn trong không gian ngữ nghĩa ẩn.

Mong đợi: Độ đầy đủ và độ chính xác của kết quả được tăng lên.

Ví dụ 6.10. Giảm lược biểu diễn với $k = 2$

Sử dụng các kết quả phân tích ở Ví dụ 6.7

```
> U = usv$u
> round(U[,1:2], 3)
      [,1] [,2]
[1,] 0.440 -0.296
[2,] 0.129 -0.331
[3,] 0.476 -0.511
[4,] 0.703  0.351
[5,] 0.263  0.647
```

```
> sigma = diag(usv$d)
> round(sigma[1:2, 1:2], 3)
      [,1] [,2]
[1,] 2.163 0.000
[2,] 0.000 1.594
```

```
> round(VT[1:2,], 3)
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 0.749 0.280 0.204 0.447 0.325 0.121
[2,] -0.286 -0.528 -0.186 0.626 0.220 0.406
```

A_k

A	d ₁	d ₂	d ₃	d ₄	d ₅	d ₆
t ₁	1	0	1	0	0	0
t ₂	0	1	0	0	0	0
t ₃	1	1	0	0	0	0
t ₄	1	0	0	1	1	0
t ₅	0	0	0	1	0	1

```
> round(U[,1:2] %% sigma[1:2,1:2] %% VT[1:2,], 3)
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 0.848 0.516 0.282 0.130 0.206 -0.076
[2,] 0.361 0.358 0.155 -0.206 -0.025 -0.180
[3,] 1.003 0.718 0.361 -0.051 0.155 -0.206
[4,] 0.978 0.130 0.206 1.029 0.617 0.411
[5,] 0.130 -0.386 -0.076 0.899 0.411 0.487
```

*A là ma trận thưa còn A_k
là ma trận đầy đủ*

Thực hiện truy vấn

Thực hiện truy vấn:

- Vec-tơ truy vấn được ánh xạ vào không gian giảm lược theo công thức:

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

- Tính độ tương đồng của các biểu diễn vec-tơ trong không gian giảm lược.

Ví dụ 6.11. Biến đổi biểu diễn truy vấn

```
> sigkinv = solve(sigma[1:2, 1:2])
> sigkinv
      [,1]      [,2]
[1,] 0.4624275 0.0000000
[2,] 0.0000000 0.6272021
```

```
> Ukt = t(U[, 1:2])
> Ukt
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.4403475 0.1293463 0.4755303 0.7030203 0.2626728
[2,] -0.2961744 -0.3314507 -0.5111152 0.3505724 0.6467468
```

```
> q <- matrix(c(1, 0, 1, 0, 1), 5, 1, byrow=TRUE)
> q
      [,1]
[1,] 1
[2,] 0
[3,] 1
[4,] 0
[5,] 1
```

```
> qk = sigkinv %*% Ukt %*% q
> qk
      [,1]
[1,] 0.5449942
[2,] -0.1006928
```

Bài tập 6.1. Giải thuật Rocchio

Giả sử hệ thống triển khai cơ chế phản hồi dựa trên giải thuật Rocchio và chỉ tìm kiếm theo tiêu đề của các văn bản. Chúng ta xét một trường hợp truy vấn ban đầu của người dùng gồm 2 từ ký hiệu là t_1 và t_2 :

$$q_0: t_1 t_2$$

Trong danh sách kết quả đầu tiên người dùng đánh dấu 2 văn bản phù hợp được ký hiệu là r_1 và r_2 , các tiêu đề lần lượt là “ $t_1 t_2 t_3 t_5$ ” và “ $t_1 t_2 t_3 t_6$ ”. Đồng thời người dùng cũng đánh dấu 1 văn bản là không phù hợp, văn bản này có tiêu đề là “ $t_2 t_6 t_8$ ”.

Giả sử hệ thống chỉ sử dụng tần suất từ, không sử dụng tần suất văn bản và chuẩn hóa độ dài trong các biểu diễn vec-tơ. Các tham số cho giải thuật Rocchio: $\alpha = 1$, $\beta = 0.75$, $\gamma = 0.25$.

Yêu cầu: Hãy tính vec-tơ truy vấn \vec{q}_m sau hiệu chỉnh.

Bài tập 6.2. Phân tích đơn trị

Cho ma trận

$$C = \begin{pmatrix} 6 & -2 \\ 3 & 0 \end{pmatrix}$$

Yêu cầu: Tìm phân tích đơn trị của C.

Bài tập 6.3. Xấp xỉ hạng thấp

- a) Tính xấp xỉ hạng $k = 1$ (C_1) cho ma trận C trong bài tập 6.2.
- b) Tính giá trị lỗi Frobenius norm: $\|C_1 - C\|_F$

