Từ điển và các danh sách thông báo

Nguyễn Mạnh Hiển
hiennm@tlu.edu.vn

Các bước xây dựng chỉ mục ngược (nhắc lại)

Bước 1. Thu thập các văn bản cần lập chỉ mục:

Friends, Romans, countrymen. | So let it be with Caesar | …

Bước 2. Tách từ, biến mỗi văn bản thành một danh sách từ:

Friends | Romans | countrymen | So | …

Bước 3. Chuẩn hóa các từ dùng các phép xử lý ngôn ngữ:

friend | roman | countryman | so | …

Bước 4. Lập chỉ mục ngược gồm từ điển (dictionary) và các danh sách thông báo (postings list).

Xây dựng từ điển

Các thao tác xử lý:

- Tách từ (tokenization)
- Xóa bỏ các ***từ dừng*** (stop word)
- Chuẩn hóa từ (normalization)
- Tách gốc từ (stemming)

Tách từ

- Tách các từ khỏi mỗi văn bản dựa trên các dấu trắng (cách, tab, xuống dòng) và các dấu câu (phẩy, chấm, chấm hỏi…).

  → Sẽ không tách được cụm từ như "New York"!
- Không phải mọi dãy ký tự tách ra đều là từ:
  - Số điện thoại: (800) 234-2333
  - Địa chỉ email: someone@gmail.com
  - Địa chỉ IP: 142.32.48.231
  - Địa chỉ web (URL): https://www.google.com

  → Chính xác hơn thì dãy ký tự đó được gọi là ***thẻ*** (token).
- Cần cách xử lý riêng để tách được các thẻ và các cụm từ có chứa dấu cách hoặc dấu câu.

Xóa bỏ các từ dừng

- Một số từ cực kì phổ biến và mang rất ít thông tin.

  → Các ***từ dừng*** (stop word).
- Các từ phổ biến trong tập bài báo tin tức Reuters-RCV1:

  a an and are as at be by for from
  has he in is it its of on that the
  to was were will with
- Xóa bỏ các từ dừng giúp giảm số thông báo (posting) phải lưu trữ xuống một cách đáng kể.
- Thỉnh thoảng các từ dừng cũng có ích:
  - Truy vấn cụm từ "President of the United States" chính xác hơn truy vấn President AND "United States".

Chuẩn hóa từ

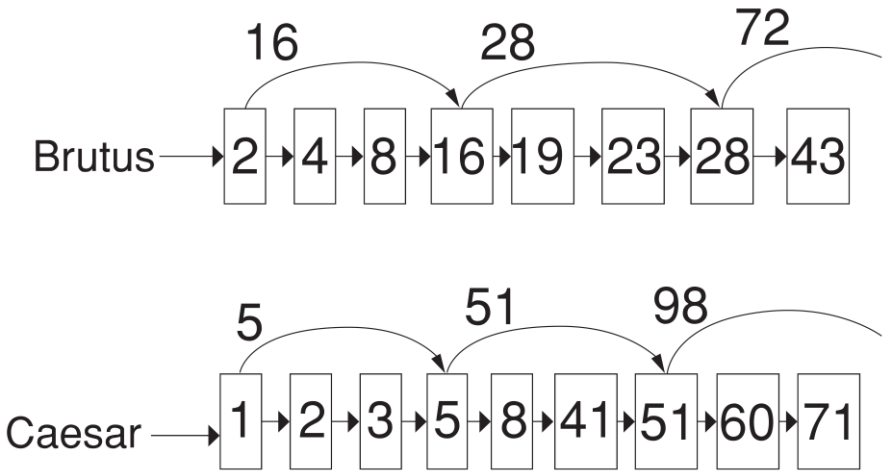
- Ánh xạ các từ khác nhau nhưng cùng nghĩa sang một từ duy nhất:
  - anti-discriminatory, antidiscriminatory $\rightarrow$ antidiscriminatory (xóa dấu gạch ngang)
  - car, automobile $\rightarrow$ car (từ đồng nghĩa)
  - window, Window $\rightarrow$ window (chuyển thành chữ thường)
- Khi người dùng truy vấn từ này (car) thì trả về cả những văn bản chứa từ tương đương (automobile).
- Chuẩn hóa từ có thể gây ra vấn đề:
  - Từ “Windows” mang nghĩa hệ điều hành trong khi từ “windows” mang nghĩa cửa sổ.

Tách gốc từ

- Quy các dạng khác nhau của cùng một gốc từ thành gốc từ:
  - organize, organizes, organizing → organiz
  - compute, computing, computation → comput
- Thuật toán Porter rất phổ biến cho việc tách gốc từ trong tiếng Anh:
  - Gồm 5 pha rút gọn từ, được áp dụng tuần tự.
  - Trong pha đầu tiên, xác định hậu tố dài nhất rồi áp dụng luật tương ứng:

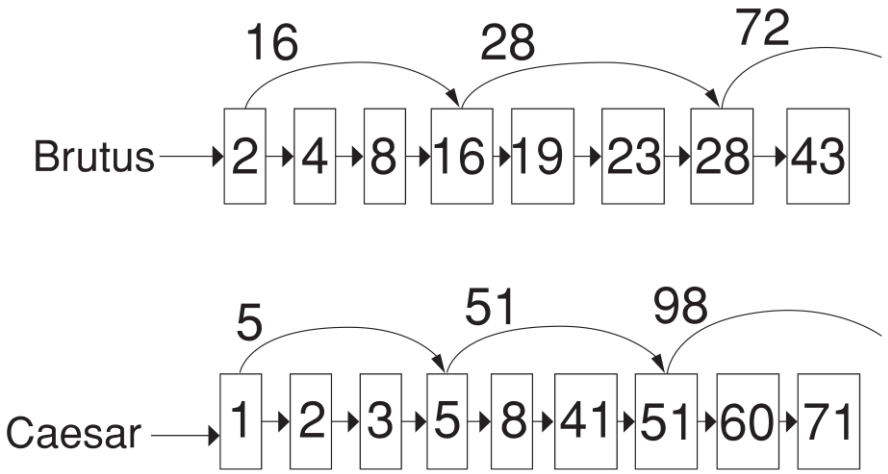
Rule			Example		
SSES	→	SS	caresses	→	caress
IES	→	I	ponies	→	poni
SS	→	SS	caress	→	caress
S	→		cats	→	cat

Lấy giao hai danh sách thông báo dùng con trỏ nhảy (skip pointer)



- Trên danh sách thông báo, một số phần tử chứa ***con trỏ nhảy*** trỏ tới một phần tử khác nằm ở phía sau:
  - 2 trỏ tới 16
  - 16 trỏ tới 28
- Nếu dùng con trỏ nhảy, ta có thể thực hiện phép giao hai danh sách thông báo nhanh hơn.

Minh họa phép giao dùng con trỏ nhảy



- Giả sử ta vừa khớp được hai phần tử 8 trên hai danh sách.
- Tiến sang 16 trên danh sách thứ nhất và tiến sang 41 trên danh sách thứ hai.
- Có con trỏ nhảy ở 16, thấy nó trỏ tới 28 trong khi 28 lại nhỏ hơn 41, nên ta nhảy tới 28 (tức là bỏ qua 19 và 23).

Thuật toán lấy giao dùng con trỏ nhảy

```
IntersectWithSkips(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then Add(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then if hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
 9                        then while hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
10                                do p1 ← skip(p1)
11                        else p1 ← next(p1)
12                else if hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
13                        then while hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
14                                do p2 ← skip(p2)
15                        else p2 ← next(p2)
16  return answer
```

Xây dựng con trỏ nhảy

- Con trỏ nhảy được tạo ra ở thời gian xây dựng chỉ mục ngược.
- Các danh sách thông báo trung gian (sinh ra khi xử lý câu truy vấn) sẽ không có con trỏ nhảy.
- Dùng con trỏ nhảy không tốt khi các danh sách thông báo phải cập nhật thường xuyên.
- Các con trỏ nhảy mau hơn $\rightarrow$ kiểm tra con trỏ nhảy thường xuyên hơn và tốn nhiều không gian lưu trữ con trỏ nhảy hơn.
- Các con trỏ nhảy thưa hơn $\rightarrow$ ít phải kiểm tra con trỏ nhảy hơn $\rightarrow$ ít cơ hội nhảy hơn.
- Một kinh nghiệm:
  - Gọi P là chiều dài danh sách thông báo.
  - Đặt $\sqrt{P}$ con trỏ nhảy cách đều nhau trên danh sách thông báo.

Chỉ mục hai từ (biword index)

- Để hỗ trợ ***truy vấn cụm từ***, ta tách hai từ liên tiếp trong một văn bản ra và xem đó như một từ thông thường trong từ điển.
  - Ví dụ, văn bản "Friends, Romans, Countrymen" sẽ sinh ra:
    - friend roman
    - roman countryman
- Các câu truy vấn ***cụm hai từ***: Có thể xử lý trực tiếp trên chỉ mục.
- Các câu truy vấn ***cụm ba từ trở lên***: Phải chia nhỏ ra.
  - Ví dụ, câu truy vấn "stanford university palo alto" phải tách thành các cụm hai từ rồi kết nối với nhau bằng phép AND:

    "stanford university" AND "university palo" AND "palo alto"

    → Cách xử lý này không đảm bảo chỉ tìm ra những văn bản chứa chính xác cụm bốn từ ban đầu.

Chỉ mục hai từ (biword index)

Đôi câu hỏi:

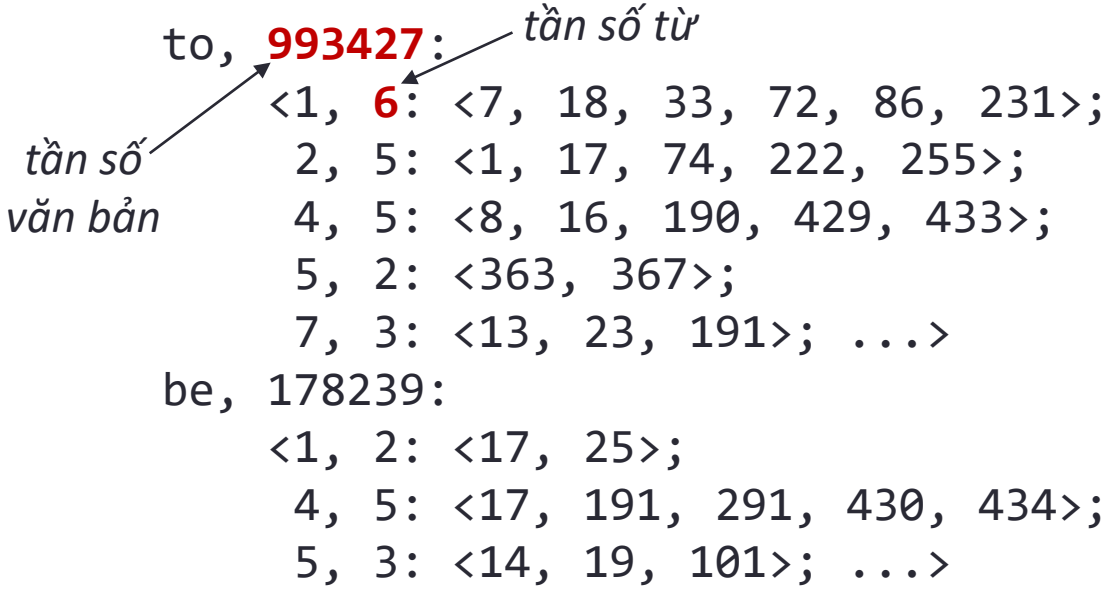
- Xử lý câu truy vấn chỉ gồm một từ trên chỉ mục hai từ như thế nào?
- Có thể mở rộng ý tưởng chỉ mục hai từ thành ***chỉ mục cụm từ*** (phrase index) được hay không, trong đó chiều dài của các cụm từ trong từ điển biến thiên (một, hai, ba…)?

Chỉ mục vị trí (positional index)

- Cách phổ biến hơn để hỗ trợ truy vấn cụm từ là dùng ***chỉ mục vị trí***.
- Mỗi thông báo (posting) có dạng:

docID: <position1, position2, …>

  - docID là mã văn bản.
  - position1, position2, … là các vị trí khác nhau của từ trong văn bản.



Chỉ mục vị trí (positional index)

Xử lý câu truy vấn cụm từ:

- Định vị các từ của câu truy vấn trên chỉ mục ngược để lấy ra các danh sách thông báo.
- Khi lấy giao hai danh sách thông báo, ngoài việc phát hiện mã văn bản giống nhau, phải kiểm tra vị trí của hai từ trong văn bản có tương thích với vị trí của chúng trong câu truy vấn hay không.

Chỉ mục vị trí (positional index)

Xét câu truy vấn cụm từ "to be or not to be":

- Giả sử hai danh sách thông báo của to và be là:

  to: < … ; 4: < … , 429, 433>; … >

  be: < … ; 4: < … , 430, 434>; … >

- Ta thấy văn bản 4 chứa cả to và be, vị trí của to (429) nhỏ hơn vị trí của be (430) một đơn vị; sau đó lại thấy một lần xuất hiện khác kiểu như vậy của cặp to-be (433-434) nhưng cách lần xuất hiện trước 4 vị trí.

Chỉ mục vị trí (positional index)

- Chỉ mục vị trí cho phép truy vấn lân cận (nhưng chỉ mục hai từ thì không).
  - Ví dụ, câu truy vấn employment /3 place nghĩa là tìm các văn bản chứa cả hai từ employment và place nhưng hai từ đó phải nằm trong phạm vi 3 từ của nhau (về cả hai phía).

- Có thể kết hợp chỉ mục cụm từ và chỉ mục vị trí được hay không?

Bài tập

1. Các phát biểu sau đây là đúng hay sai?

a. Trong một hệ truy hồi Boole, tách gốc từ không bao giờ làm giảm độ chính xác (precision).

b. Trong một hệ truy hồi Boole, tách gốc từ không bao giờ làm giảm độ thu hồi (recall).

c. Tách gốc từ làm tăng kích thước từ điển.

d. Tách gốc từ được thực hiện ở thời gian xây dựng chỉ mục, nhưng không cần tách gốc từ đối với các từ trong câu truy vấn.

Bài tập

2. Các cặp từ sau đây được quy về cùng một gốc từ bằng thuật toán Porter. Theo các bạn, những cặp từ nào không nên hợp nhất? Giải thích.

a. abandon / abandonment
b. absorbency / absorbent
c. marketing / markets
d. university / universe
e. volume / volumes

Bài tập

3. Vì sao con trỏ nhảy không hữu ích cho câu truy vấn dạng x OR y?

4. Xét một câu truy vấn gồm hai từ. Một từ có danh sách thông báo gồm 16 phần tử:

[4, 6, 10, 12, 14, 16, 18, 20, 22, 32, 47, 81, 120, 122, 157, 180]

và từ kia có danh sách thông báo chỉ gồm một phần tử:

[47].

Hỏi có bao nhiêu lần so sánh khi lấy giao hai danh sách với hai chiến lược sau đây?

a. Dùng các danh sách thông báo chuẩn.

b. Dùng các danh sách thông báo có con trỏ nhảy, với bước nhảy gợi ý là $\sqrt{P}$, trong đó P là chiều dài danh sách.

Bài tập

5. Xét việc lấy giao một danh sách thông báo có con trỏ nhảy sau đây:

với một danh sách thông báo trung gian (và vì vậy không có con trỏ nhảy) sau đây:

3 5 89 95 97 99 100 101.

Giả sử ta đang thực hiện thuật toán lấy giao ở slide 10:

a. Hỏi có bao nhiêu lần nhảy theo các con trỏ nhảy?
b. Hỏi có bao nhiêu lần so sánh các thông báo?
c. Hỏi có bao nhiêu lần so sánh các thông báo khi không dùng con trỏ nhảy?

Bài tập

6. Cho một phần của chỉ mục vị trí như sau:

angels: 2: <36, 174, 252, 651>; 4: <12, 22, 102, 432>; 7: <17>

fools: 2: <1, 17, 74, 222>; 4: <8, 78, 108, 458>; 7: <3, 13, 23, 193>

fear: 2: <87, 704, 722, 901>; 4: <13, 43, 113, 433>; 7: <18, 328, 528>

in: 2: <3, 37, 76, 444, 851>; 4: <10, 20, 110, 470, 500>; 7: <5, 15, 25, 195>

rush: 2: <2, 66, 194, 321, 702>; 4: <9, 69, 149, 429, 569>; 7: <4, 14, 404>

to: 2: <47, 86, 234, 999>; 4: <14, 24, 774, 944>; 7: <199, 319, 599, 709>

tread: 2: <57, 94, 333>; 4: <15, 35, 155>; 7: <20, 320>

where: 2: <67, 124, 393, 1001>; 4: <11, 41, 101, 421, 431>; 7: <16, 36, 736>

Xác định các văn bản thỏa mãn các câu truy vấn bên dưới (trong đó cặp dấu nháy kép chỉ truy vấn cụm từ):

a. “fools rush in”

b. “fools rush in” AND “angels fear to tread”

Bài tập

7. Cho một phần của chỉ mục vị trí như sau:

Gates: 1: <3>; 2: <6>; 3: <2, 17>; 4: <1>

IBM: 4: <3>; 7: <14>

Microsoft: 1: <1>; 2: <1, 21>; 3: <3>; 5: <16, 22, 51>

Xác định các văn bản thỏa mãn câu truy vấn lân cận Gates /k Microsoft, trong đó:

a. k = 1

b. k = 3

c. k = 5