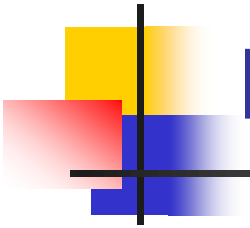
IT4853
Tìm kiếm và trình diễn thông tin

Bài 3. Xử lý từ truy vấn

IIR.C3. Dictionaries and tolerant retrieval

TS. Nguyễn Bá Ngọc, *Bộ môn Hệ thống thông tin, Viện CNTT & TT*
ngocnb@soict.hust.edu.vn

Hà Nội, 2016



Nội dung chính

- 1. Bộ từ vựng
- 2. Kiểu truy vấn
  - Truy vấn Boolean
  - Truy vấn mẫu từ
  - Truy vấn trích đoạn
- 3. Khoảng cách soạn thảo

Dữ liệu từ vựng

Từ	Số lượng văn bản	Con trỏ tới danh sách thẻ định vị
a	3212	--->
b	35	--->
c	128	--->
…	…	…
tn	620	--->

char[20] int Postings *

20 bytes 4/8 bytes 4/8 bytes

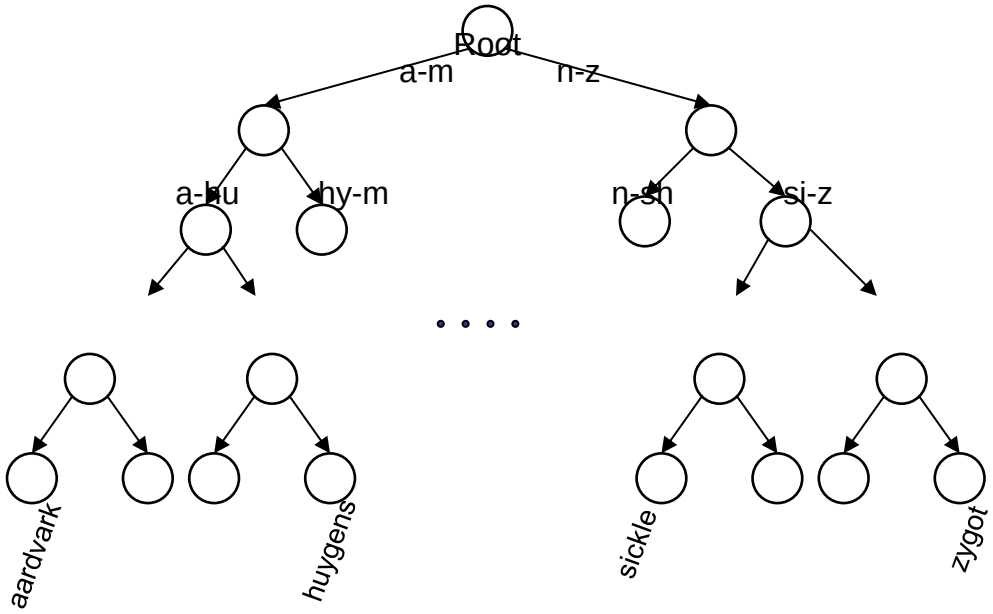
Tối ưu hóa bộ từ vựng:

Giảm kích thước (nén);

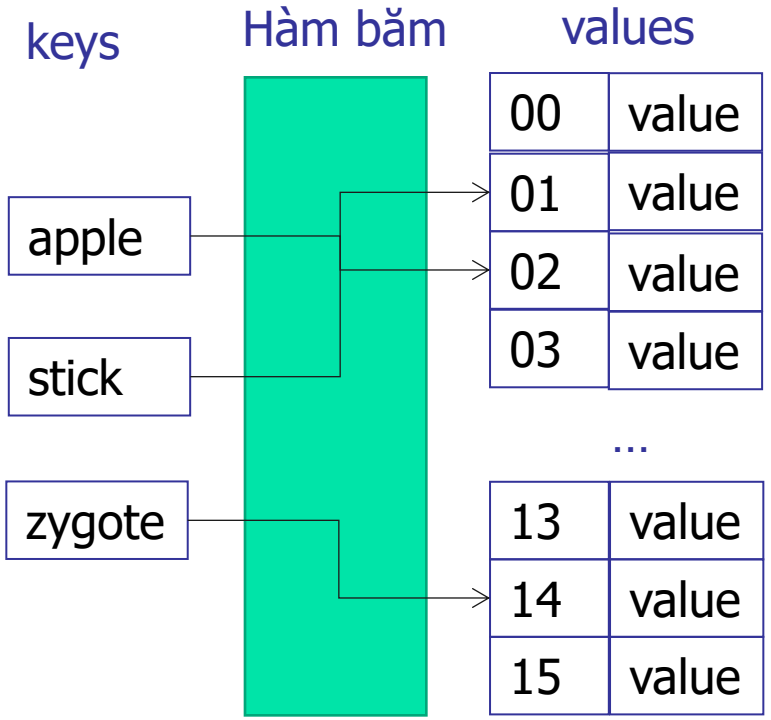
Tăng tốc độ tìm từ (cấu trúc dữ liệu trong bộ nhớ).

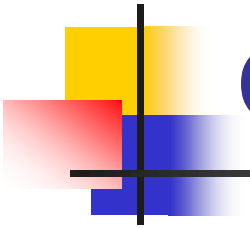
Tối ưu hóa tốc độ tìm từ

Cây nhị phân tìm kiếm



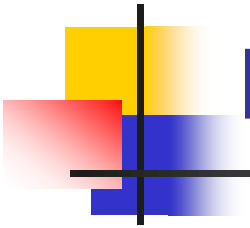
Bảng băm





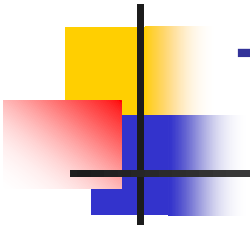
Cây nhị phân tìm kiếm vs bảng băm

- Trong cây nhị phân tìm kiếm từ khóa được sắp xếp theo thứ tự, vì vậy cho phép thực hiện nhiều dạng truy vấn hơn so với bảng băm, vd, truy vấn mẫu từ;
- Trong bảng băm từ khóa không được sắp xếp theo thứ tự, tuy nhiên tốc độ tìm từ nhanh hơn so với cây nhị phân tìm kiếm.



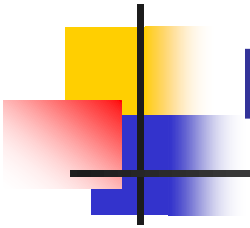
Nội dung chính

- 1. Bộ từ vựng
- 2. Kiểu truy vấn
  - Truy vấn Boolean
  - Truy vấn mẫu từ
  - Truy vấn trích đoạn
- 3. Khoảng cách soạn thảo



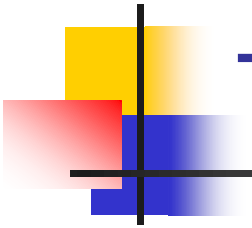
Truy vấn Boolean

- Mô tả luật xuất hiện của từ trong văn bản
- Các liên kết cơ bản:
  - OR
  - AND
  - AND NOT (hoặc BUT)



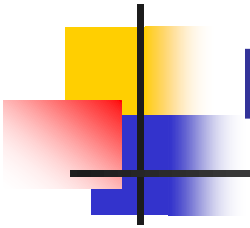
Nội dung chính

- 1. Bộ từ vựng
- 2. Kiểu truy vấn
  - Truy vấn Boolean
  - Truy vấn mẫu từ
  - Truy vấn trích đoạn
- 3. Khoảng cách soạn thảo



Truy vấn mẫu từ

- *IIR Slides*



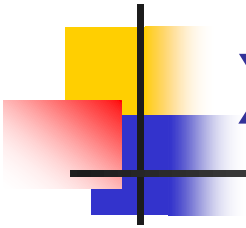
Mẫu từ: *

- ***a*:** Tất cả từ bắt đầu với a
- ****a*:** Tất cả từ kết thúc với a
- ***a *b*:** Tất cả từ bắt đầu với a, kết thúc với b

Cần thiết lập chỉ mục chuyên biệt để xử lý mẫu từ

Truy vấn mẫu từ: *

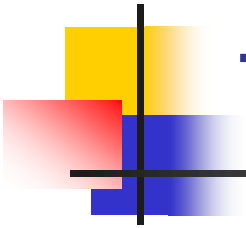
- mon*: Tìm tài liệu chứa từ bất kỳ bắt đầu bằng mon
  - B-tree: lấy toàn bộ từ thỏa mãn: mon ≤ t < moo
- *mon: tìm tài liệu chứa từ bất kỳ kết thúc bằng mon
  - Xử dụng thêm một cây chứa từ theo trình tự ngược backwards
  - Lấy tất cả từ trong khoảng: nom ≤ t < non
- Giải thuật tìm kiếm:
  - Tìm tập từ khóa khớp với mẫu từ, sau đó …;
  - … Trả về văn bản chứa bất kỳ từ nào trong tập từ khớp mẫu.



Xử lý mẫu a*b

- Ví dụ: m*nchen
  - Có thể tìm m* và *nchen sau đó lấy giao của hai tập hợp;
  - Tuy nhiên chi phí lớn.
- Giải pháp: chỉ mục từ xoay
  - Ý tưởng: Xoay mẫu từ sao cho dấu * xuất hiện ở cuối.
  - Lưu kết quả xoay từ trong từ điển

Permuterm index: Chỉ mục từ xoay



Từ xoay

- Các từ xoay cho từ HELLO là: hello$, ello$h, llohe, lohel, o$hell, và $hello, trong đó $ là ký tự đặc biệt không xuất hiện trong từ bất kỳ.

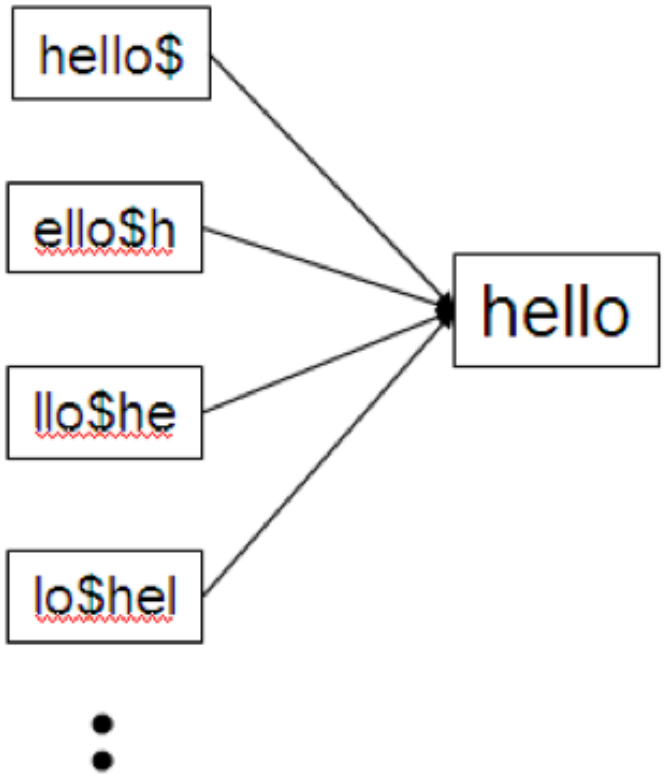
Chỉ mục từ xoay

- Lưu tất cả từ xoay cho từ chỉ mục bất kỳ;
- Xử lý mẫu từ:
  - X, tìm X$
  - X*, tìm $X*
  - *X, tìm X$*
  - *X*, tìm X*
  - X*Y, tìm Y$X*

Ví dụ: cho từ hel*o, tìm o$hel*

Chỉ mục từ xoay (2)

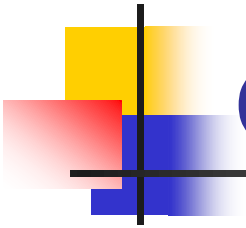
- Sử dụng B-Tree để lưu kết quả xoay từ;
- Liên kết từ xoay với từ gốc.



Chỉ mục từ xoay (3)

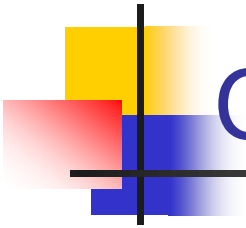
- Chỉ mục từ xoay hay là cây từ xoay?
  - Thuật ngữ chỉ mục từ xoay được sử dụng phổ biến hơn;
  - Cây từ xoay có thể sát nghĩa hơn (bài tập 3.2).

*Vấn đề: Chỉ mục từ xoay có kích thước lớn hơn nhiều lần so với chỉ mục thông thường.



Chỉ mục k-gram

- Lưu toàn bộ k-grams theo ký tự (chuỗi k ký tự liên tiếp) xuất hiện trong từ
  - 2-grams được gọi là bigrams.
- Ví dụ, các bigram đối với chuỗi April is the cruelest là: $a ap pr ri il l$ $i is s$ $t th he e$ $c cr ru ue el le es st t$ $m mo on nt h$
  - $ là ký tự đặc biệt, có vai trò tương tự như trong chỉ mục từ xoay.

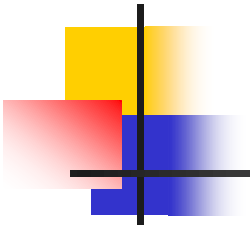


Chỉ mục k-gram (2)

- Sử dụng chỉ mục ngược trên bigrams tới từ
- Chúng ta có hai lớp chỉ mục ngược
  - Chỉ mục k-gram để tìm từ theo truy vấn chứa k-grams
  - Chỉ mục ngược từ-văn bản để tìm văn bản theo từ truy vấn

Xử lý mẫu từ bằng chỉ mục bigram

- Truy vấn mon* sẽ được xử lý như: $m AND mo AND on
  - Mục đích là tìm tất cả từ bắt đầu với mon . . .
  - . . . có thể có nhiều lỗi "false positives", v.d., MOON.
- Để có kết quả chính xác cần lọc các từ tìm được bằng cách so sánh với mẫu từ.
  - Chỉ sử dụng các từ khớp mẫu để tiếp tục tìm văn bản.



Chỉ mục k-grams vs. chỉ mục xoay từ

- Chỉ mục k-gram chiếm ít bộ nhớ hơn so với chỉ mục từ xoay;
- Xử lý mẫu từ trên chỉ mục từ xoay nhanh hơn và không cần lọc từ tìm được.

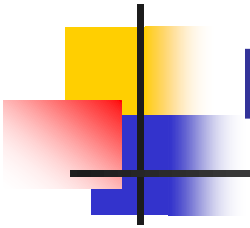
Truy vấn mẫu từ trong Google

- Google chỉ hỗ trợ truy vấn mẫu từ ở mức độ hạn chế.
- Ví dụ, Google sẽ xử lý không tốt truy vấn [gen* universit*]
  - Tình huống: Tìm the University of Geneva, nhưng không biết cách viết chính xác các từ university và Geneva.
- Theo thông tin chính thức từ Google, 2010-04-29: "Dấu * chỉ có thể thay thế cho một từ hoàn chỉnh, không phải một phần của từ."
  - Nhưng điều này không hoàn toàn đúng. Vi dụ, [pythag*] và [m*nchen]

Thử giải thích vì sao Google hạn chế truy vấn mẫu từ?

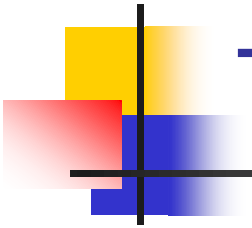
Truy vấn mẫu từ trong Google

- Vấn đề 1: Có thể phải thực hiện một lượng lớn truy vấn Boolean.
  - Kết hợp kết quả so khớp mẫu bằng liên kết OR;
  - Đối với [gen* universit*] sẽ tìm geneva university OR geneva université OR genève university OR genève université OR general universities OR . . .
  - Khối lượng tính toán rất lớn
- Vấn đề 2: Người dùng ưa chuộng cách viết tắt
  - Nếu mẫu [pyth* theo*] là hợp lệ cho [pythagoras' theorem] người dùng sẽ sử dụng thường xuyên.
  - Chi phí thực hiện truy vấn có thể tăng đáng kể.
- Những vấn đề liên quan tới viết những truy vấn phức tạp đã được giải quyết một phần qua gợi ý truy vấn.



Nội dung chính

- 1. Bộ từ vựng
- 2. Kiểu truy vấn
  - Truy vấn Boolean
  - Truy vấn mẫu từ
  - Truy vấn trích đoạn
- 3. Khoảng cách soạn thảo



Truy vấn trích đoạn

- Thường được sử dụng trong trường hợp cần tìm văn bản khi đã biết một phần nội dung của văn bản
- Truy vấn trích đoạn thường được đặt trong dấu nháy kép:
  - "Công nghệ thông tin"

Truy vấn với giới hạn khoảng cách

- Giới hạn về khoảng cách giữa các từ truy vấn trong văn bản
  - Ví dụ: việc làm **/4** lĩnh vực
  - Tìm tất cả văn bản chứa việc làm và lĩnh vực trong giới hạn khoảng cách 4 từ.
    - Position(lĩnh vực) – position(việc làm) <= 4
- Có thể coi giới hạn khoảng cách là trường hợp khái quát của truy vấn trích đoạn
  - Khoảng cách giữa các từ truy vấn bằng 1.

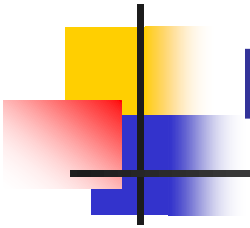
Truy vấn với giới hạn khoảng cách (2)

- Sử dụng chỉ mục ngược có vị trí:
  - Từ: mã_văn_bản: <danh_sách_vị_trí>; mã_văn_bản: <danh_sách_vị_trí>; ...

Ví dụ chỉ mục có vị trí:

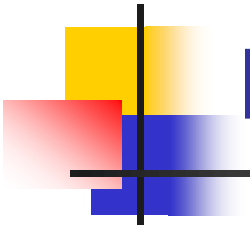
tìm_kiếm, 5: 1: <1>; 2: <6>; 3: <2, 15>; 4: <1>, 8:<2>.
dữ_liệu, 5: 1: <3>; 3: <5, 16>; 4: <6>; 7: <14>, 8:<5>.
thông_tin, 5: 1: <2>; 2: <12, 16, 21>; 3: <18>; 5: <21, 25>, 8:<3>

Truy vấn: tìm_kiếm /2 dữ_liệu
Kết quả: {1, 3}



Nội dung chính

- 1. Bộ từ vựng
- 2. Kiểu truy vấn
  - Truy vấn Boolean
  - Truy vấn mẫu từ
  - Truy vấn trích đoạn
- 3. Khoảng cách soạn thảo



Khoảng cách soạn thảo

- Khoảng cách soạn thảo giữa chuỗi ký tự s_1 và s_2 là số thao tác soạn thảo cơ bản để biến s_1 thành s_2.
- Ứng dụng:
  - Gợi ý từ truy vấn;
  - Sửa lỗi cú pháp.

Khoảng cách soạn thảo (2)

- Các thao tác cơ bản trong khoảng cách **Levenshtein** là: *chèn* (*insert*), *xóa* (*delete*), *và thay thế* (*replace*)
- Khoảng cách **Damerau-Levenshtein:** bổ xung thao tác *hoán vị* (*transposition*).

Giải thuật quy hoạch động tính khoảng cách Levenshtein

```
LevenshteinDistance(s1, s2)
 1  for i ← 0 to |s1|
 2  do m[i, 0] = i
 3  for j ← 0 to |s2|
 4  do m[0, j] = j
 5  for i ← 1 to |s1|
 6  do for j ← 1 to |s2|
 7       do if s1[i] = s2[j]
 8            then m[i, j] = min{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]}
 9            else  m[i, j] = min{m[i-1, j]+1, m[i, j-1]+1, m[i-1, j-1]+1}
10  return m[|s1|, |s2|]
```

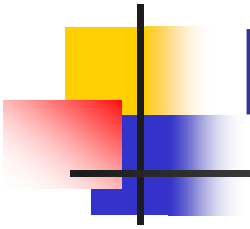
Operations: insert (cost 1), delete (cost 1), replace (cost 1), copy (cost 0)

Ví dụ tính khoảng cách Levenshtein

		f		a		s		t	
	0	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**
c	**1** **1**	*1* *2*	*2* *1*	**2** **2**	3 **2**	**3** **3**	4 **3**	**4** **4**	5 **4**
a	**2** **2**	**2** 3	**2** **2**	*1* *3*	*3* *1*	*3* *2*	*4* *2*	4 **3**	5 **3**
t	**3** **3**	**3** 4	**3** **3**	3 4	**2** **2**	*2* *3*	*3* *2*	*2* *3*	*4* *2*
s	**4** **4**	**4** 5	**4** **4**	4 5	**3** **3**	**2** 4	3 **2**	*3* *3*	*3* *3*

Các giá trị trong ma trận Levenshtein

s1[i] == s2[j] ? m[i – 1, j–1]: m[i – 1, j – 1] + 1	*copy* *replace*	m[i – 1, j] + 1	*delete*(s_1[i])
m[i, j – 1] + 1	insert (s_2[j])	min	



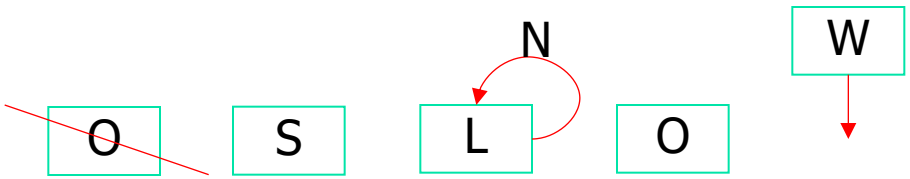
Bài tập 3.1

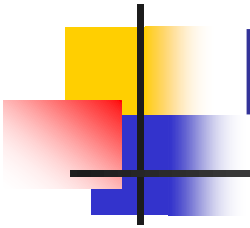
- Tính ma trận khoảng cách *Levenshtein* cho *OSLO* –> *SNOW*;
- Xác định các thao tác soạn thảo.

		s	n	o	w
	0	1 1	2 2	3 3	4 4
o	1 / 1				
s	2 / 2				
l	3 / 3				
o	4 / 4				

		s	n	o	w
	0	**1** **1**	**2** **2**	**3** **3**	**4** **4**
o	**1** / **1**	**1** 2 / 2 **1**	**2** 3 / **2** **2**	**2** 4 / 3 **2**	4 5 / **3** **3**
s	**2** / **2**	**1** 2 / 3 **1**	**2** 3 / **2** **2**	**3** **3** / **3** **3**	**3** 4 / 4 **3**
l	**3** / **3**	3 **2** / 4 **2**	**2** 3 / 3 **2**	**3** 4 / **3** **3**	**4** **4** / **4** **4**
o	**4** / **4**	4 **3** / 5 **3**	**3** **3** / 4 **3**	**2** 4 / 4 **2**	4 5 / **3** **3**

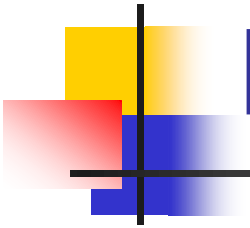
cost	operation	input	output
1	delete	o	*
0	(copy)	s	s
1	replace	l	n
0	(copy)	o	o
1	insert	*	w





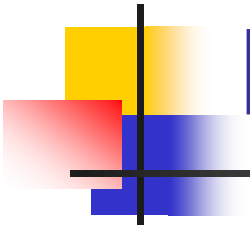
Bài tập 3.2

Trong chỉ mục từ xoay, mỗi từ xoay chỉ đến danh sách từ gốc của từ đó. Có bao nhiêu từ gốc trong danh sách thẻ định vị của mỗi từ xoay?



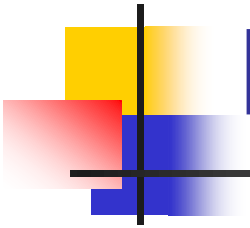
Bài tập 3.3

Cho truy vấn fi*mo*er. Có thể sử dụng truy vấn Boolean nào trên bigram cho truy vấn này? Hãy thử lấy ví dụ một từ thỏa mãn mẫu er$fi* nhưng không thỏa mãn truy vấn bigram này ?



Bài tập 3.4

Hãy thử lấy ví dụ một câu không thỏa mãn mẫu mon*h nhưng được trả về nếu chỉ sử dụng biểu thức AND trên bigram?



Bài tập 3.5

Tính khoảng cách soạn thảo giữa hai từ COLD và SLOW và các thao tác soạn thảo để biến COLD thành SLOW.