

Tìm kiếm và trình diễn thông tin

Nén chỉ mục ngược
Index Compression

Nội dung chính

1. Nén từ điển
2. Nén danh sách mã văn bản

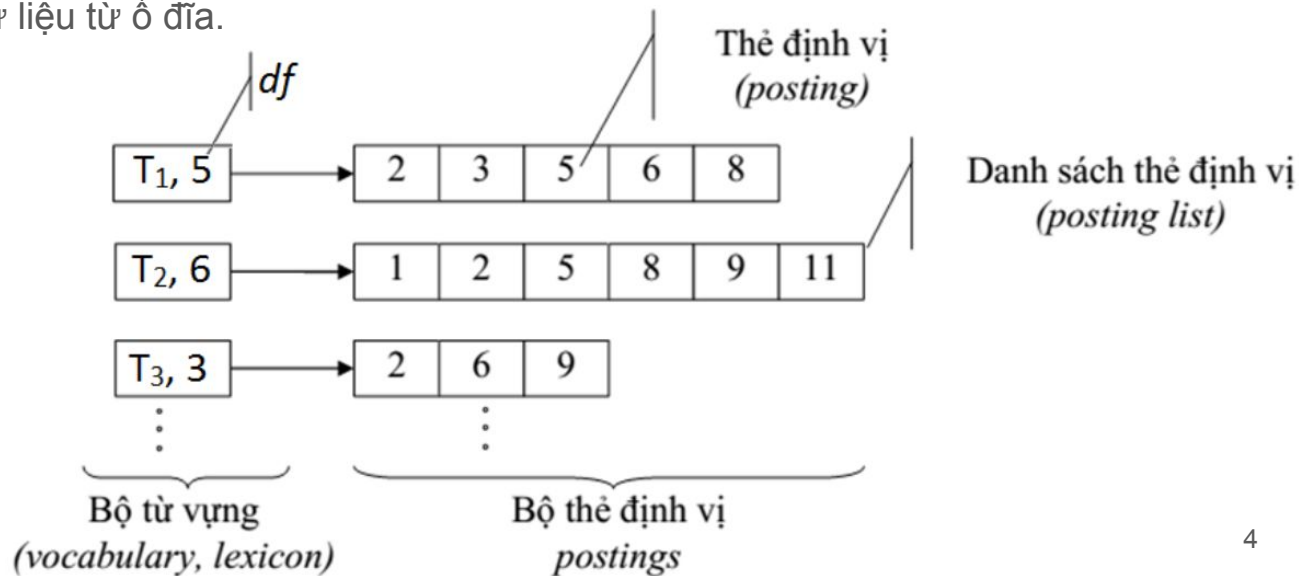
Nén bảo toàn vs. không bảo toàn

- Nén bảo toàn vs. không bảo toàn

- Nén bảo toàn: Dữ liệu được bảo toàn sau khi giải nén; Phổ biến nhất trong tìm kiếm.
- Nén không bảo toàn:
 - Loại bỏ một phần dữ liệu, tỉ lệ nén thường cao hơn phương pháp bảo toàn;
 - Có thể coi các phép lọc trong quá trình tách từ (chuẩn hóa cách viết, loại từ dừng, v.v.) là những phương pháp nén không bảo toàn.

Lý do nén từ điển

- Lý do nén từ điển
 - Thực hiện truy vấn luôn bắt đầu với tìm kiếm từ trong từ điển:
 - Cần sử dụng cấu trúc dữ liệu trong bộ nhớ để tìm kiếm nhanh;
- Áp dụng phương pháp nén giúp:
 - Lưu từ điển kích thước lớn trong bộ nhớ;
 - Giảm thời gian tải dữ liệu từ ổ đĩa.



Mảng phần tử kích thước tĩnh

An array of fixed-width entries

term	document frequency	pointer to postings list
a	656,265	→
aachen	65	→
...
zulu	221	→

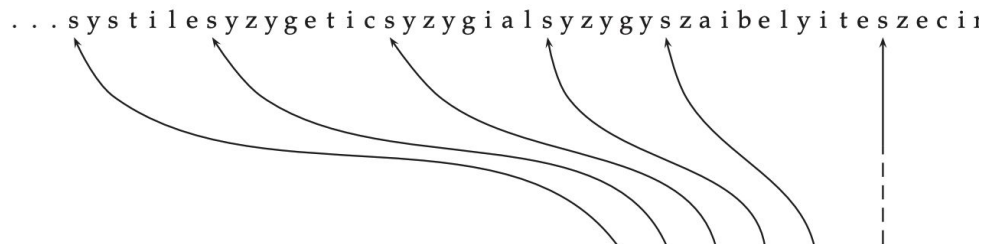
space needed: 20 bytes 4 bytes 4 bytes

For Reuters-RCV1, we need $M \times (20 + 4 + 4) = 400,000 \times 28 = 11.2\text{megabytes (MB)}$ for storing the dictionary in this scheme.

Chuỗi ký tự dài

- Dictionary as a string
- Lưu bộ từ vựng như một chuỗi ký tự dài:
 - Con trỏ tới từ tiếp theo là dấu hiệu kết thúc từ hiện tại

... systilesyzygeticsyzygialsyzygyszaibelyiteszec i



$$400,000 \times (4 + 4 + 3 + 8) = 7.6 \text{ MB}$$

- 4 bytes each for frequency and postings pointer
- 3 bytes for the term pointer
- 8 bytes on average for the term

freq.	postings ptr.	term ptr.
9	→	
92	→	
5	→	
71	→	
12	→	
...
4 bytes	4 bytes	3 bytes

Phân đoạn chuỗi ký tự dài

- Blocked storage
- Lưu con trỏ tới từ đầu tiên trong khối k từ.
- Bổ sung 1 byte để lưu độ dài từ

$k = 4$

3 bytes for the term pointer

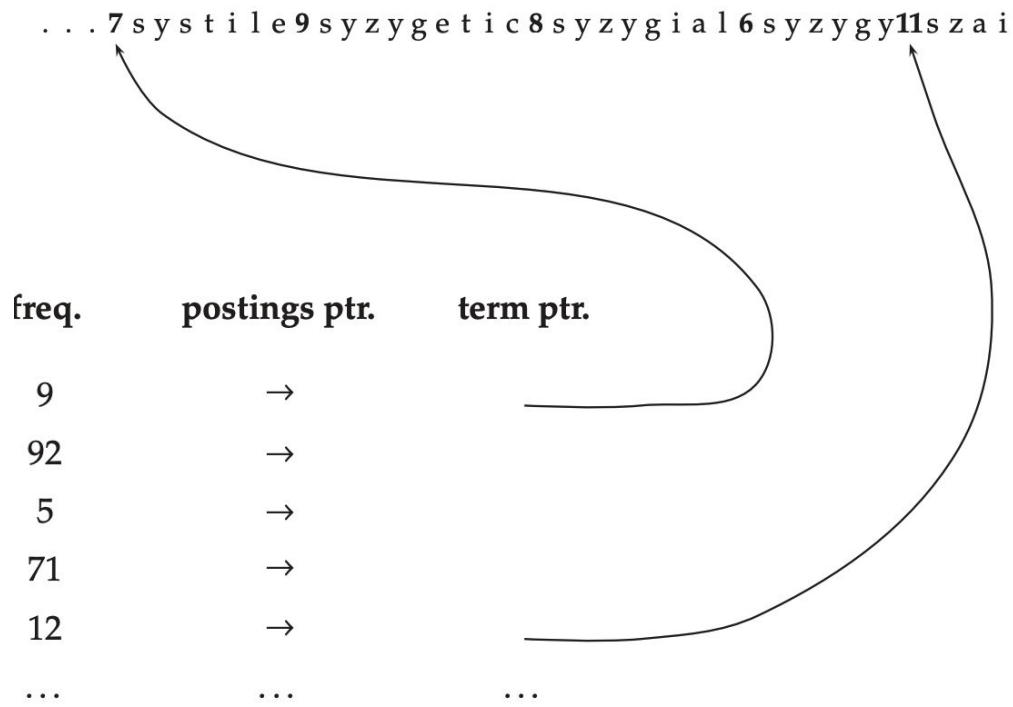
=> save $(k - 1) \times 3 = 9$ bytes for term pointers,

but need an additional $k = 4$ bytes for term lengths

=> reduce by 5 bytes per four-term block

or a total of $400,000 \times 1/4 \times 5 = 0.5$ MB

7.6 MB -> 7.1 MB



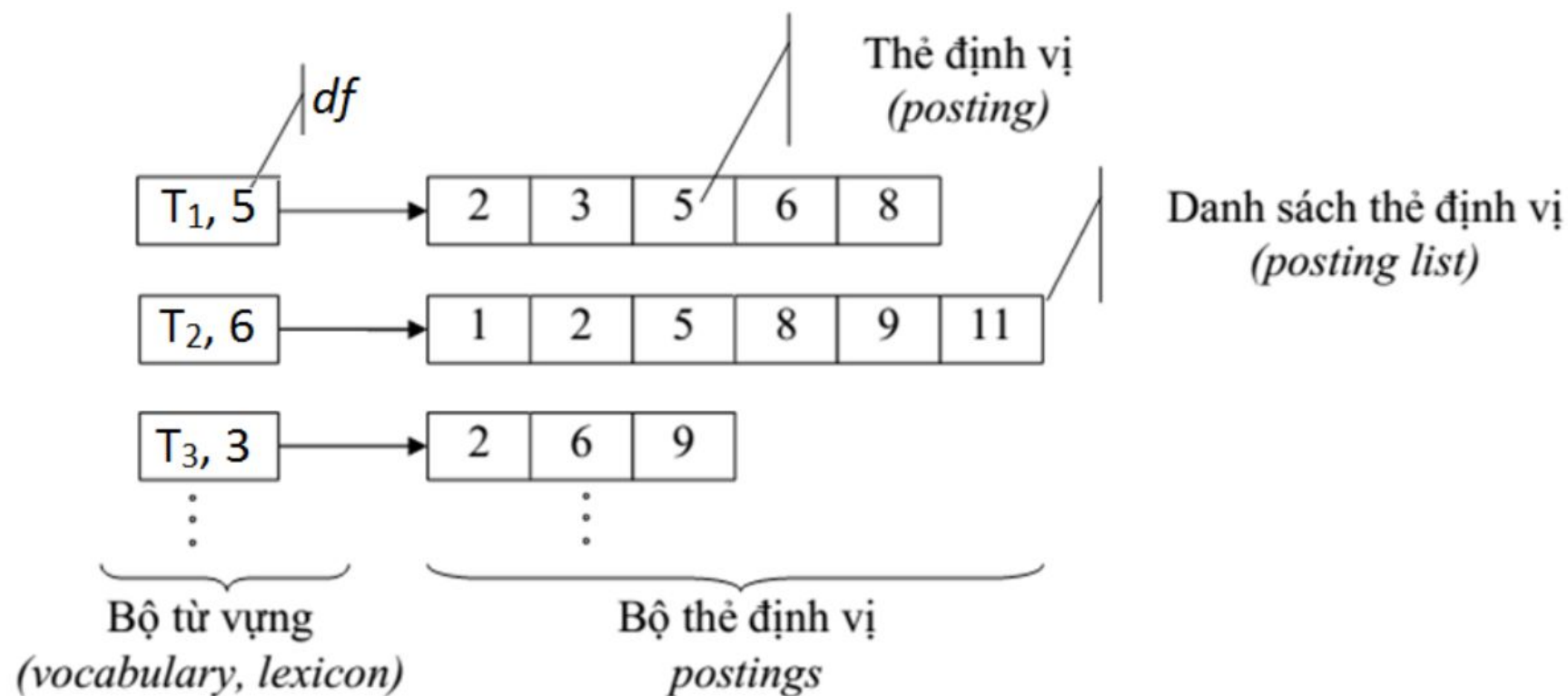
Chuỗi ký tự dài, phân đoạn và Front- coding

- Blocked storage & front coding
- Đặc điểm: Những từ đã sắp xếp thường có phần bắt đầu giống nhau
- Front-coding: Trong khối, lưu hoàn chỉnh từ đầu tiên và phần khác biệt của các từ tiếp theo

8automata8automate9automatic10automation

8automat★a1◇e2◇ic3◇ion

- Phần đầu automat
- Độ dài phần mở rộng ngoài automat.



Nén danh sách mã văn bản

Nén danh sách mã văn bản

- Xét trường hợp đơn giản nhất khi chỉ lưu mã văn bản theo trật tự tăng dần trong danh sách thẻ định vị.
 - Ví dụ, mô hình Boolean.
- Mục đích nén:
 - Giảm kích thước danh sách thẻ định vị;
 - Lưu số lượng lớn thẻ định vị trong bộ nhớ;
 - Giảm thời gian đọc từ ổ đĩa.

Danh sách khoảng cách

- Các mã văn bản trong danh sách được lưu theo thứ tự tăng dần,
 - Ví dụ: Máy tính: 33,47,154,159,202 ...
- Có thể thay bằng khoảng cách
 - Ví dụ: Máy tính: 33,14,107,5,43 ...

Mã VB

- Mã VB: Variable Bytes Code. Là phương pháp mã hóa sử dụng số byte thay đổi phù hợp với kích thước mã văn bản.
- Mã hóa:
 - Sử dụng 1 byte để lưu một nhóm,
 - Gồm nhóm 7 bits,
 - **Đặt bit cao nhất (bit c) của byte phải nhất bằng 1, với các bytes còn lại đặt $c = 0$,**
 - Dãy byte thu được là mã VB của khoảng cách G.

Mã VB

VBENCODENUMBER(n)

```
1  bytes  $\leftarrow \langle \rangle$ 
2  while true
3  do PREPEND(bytes,  $n \bmod 128$ )
4      if  $n < 128$ 
5          then BREAK
6       $n \leftarrow n \div 128$ 
7  bytes[LENGTH(bytes)]  $\mathrel{+}= 128$ 
8  return bytes
```

VBENCODE(*numbers*)

```
1  bytestream  $\leftarrow \langle \rangle$ 
2  for each  $n \in \textit{numbers}$ 
3  do bytes  $\leftarrow$  VBENCODENUMBER( $n$ )
4      bytestream  $\leftarrow$  EXTEND(bytestream, bytes)
5  return bytestream
```

VBDECODE(*bytestream*)

```
1  numbers  $\leftarrow \langle \rangle$ 
2   $n \leftarrow 0$ 
3  for  $i \leftarrow 1$  to LENGTH(bytestream)
4  do if bytestream[ $i$ ]  $< 128$ 
5      then  $n \leftarrow 128 \times n + \textit{bytestream}[i]$ 
6      else  $n \leftarrow 128 \times n + (\textit{bytestream}[i] - 128)$ 
7          APPEND(numbers,  $n$ )
8           $n \leftarrow 0$ 
9  return numbers
```

Mã VB

docIDs	824	829	215406
gaps		5	214577
VB code	00000110 10111000	10000101	00001101 00001100 10110001

214577

00001101 00001100 10110001

824

00000011 00111000

00000110 10111000 00001101 00001100 10110001

Đơn vị mã hóa tối ưu cho mã VB

- Nếu sử dụng đơn vị mã hóa lớn sẽ lãng phí bộ nhớ đối với các khoảng cách nhỏ, ngược lại nếu sử dụng đơn vị nhỏ sẽ lãng phí bộ nhớ đối với giá trị lớn.
- Có thể sử dụng các đơn vị mã hóa khác: 32 bits, 16 bits, 4 bits tùy theo đặc điểm phân bố giá trị số;
- Hoặc gom các giá trị nhỏ thành giá trị lớn hơn, v.v.

Mã Unary code

Biểu diễn số n như chuỗi n số 1 thêm số 0 ở cuối.

Unary code của 3 là 1110.

Unary code của 40 là 111111111111111111111111111111111111110 .

Mã Gamma

- Biểu diễn một khoảng cách G bằng *offset* và *length*
- *offset* là mã nhị phân của G loại bỏ bit đứng đầu
 - Ví dụ $13 = 1101$; $\text{offset}(13) = 101$
- *length* là Unary Code của độ dài của *offset*
 - Với 13 : $\text{offset} = 101$, $\text{length} = 1110$.
- Mã Gamma = $\text{length} + \text{offset}$
 - Mã Gamma của 13 là 1110101

Mã Gamma

number	unary code	length	offset	γ code
0	0			
1	10	0		0
2	110	10	0	10,0
3	1110	10	1	10,1
4	11110	110	00	110,00
9	1111111110	1110	001	1110,001
13		1110	101	1110,101
24		11110	1000	11110,1000
511		1111111110	11111111	111111110,11111111
1025		111111111110	0000000001	111111111110,0000000001

Hãy biểu diễn $n=30$ dùng mã VB và mã gamma

Mã Gamma vs. mã VB

Mã Gamma vs. mã VB

- Luôn có thể giải mã cùng tiến trình đọc dữ liệu.
- Mã Gamma có tỉ lệ nén ổn định cho mọi giá trị mã văn bản và nén tốt hơn mã VB;
- Mã Gamma sử dụng các thao tác trên bits nên chậm hơn mã VB.

Bài tập

Cho danh sách mã văn bản sau:

1, 3, 10, 120, 121

a) Hãy xác định danh sách khoảng cách;

b) Hãy mã hóa kết quả mục a bằng mã VB, đơn vị mã hóa là 8 bits;

c) Hãy mã hóa kết quả mục a bằng mã Gamma.

Bài tập

a) Dãy bits sau là mã VB của danh sách khoảng cách, đơn vị mã hóa là 4 bits (nibbles):

1010 0001 1011 0101 1000

Hãy xác định danh sách mã văn bản ban đầu.

b) Dãy bits sau là mã Gamma của danh sách khoảng cách:

110011110000101

Hãy xác định danh sách mã văn bản ban đầu.