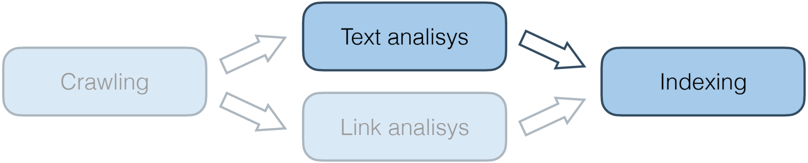
Information Retrieval 1

Indexing

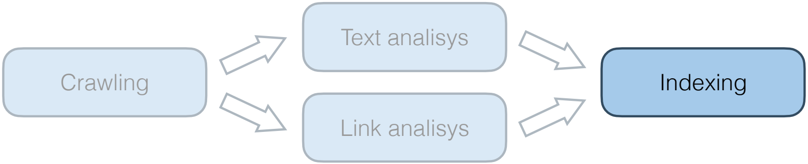
Ilya Markov
i.markov@uva.nl

University of Amsterdam

Recap IR0



Indexing



Outline

1. Data structures
2. Inverted index
3. Constructing an index
4. Updating an index

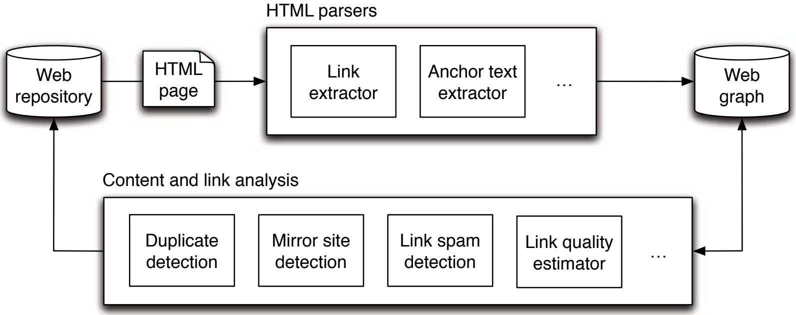
Outline

1. Data structures
2. Inverted index
3. Constructing an index
4. Updating an index

Full indexing architecture

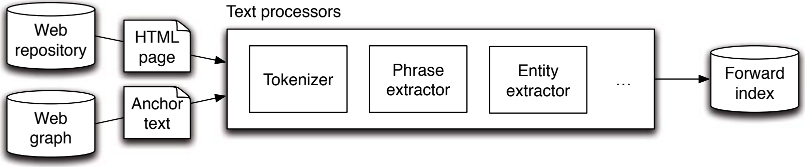
- Inverted index
- Web graph
- Forward index
- Page attribute file

Web graph



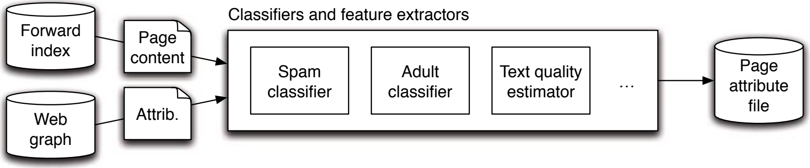
B. Cambazoglu and R. Baeza-Yates, "Scalability Challenges in Web Search Engines"

Forward index



B. Cambazoglu and R. Baeza-Yates, "Scalability Challenges in Web Search Engines"

Page attribute file



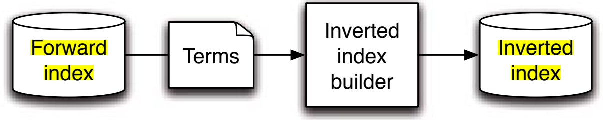
B. Cambazoglu and R. Baeza-Yates, "Scalability Challenges in Web Search Engines"

Page attribute file

Feature	Source	Description
Language	Page content	Language of the page
Length	Page content	Number of words or characters in the page
Content spam	Page content	Score indicating the likelihood that the page content is spam
Text quality	Page content	Score combining various text quality features (e.g., readability)
Link quality	Web graph	Page importance estimated based on page's link structure
CTR	Query logs	Click-through rate of the page in search results (if available)
Dwell time	Query logs	Average time spent by the users on the page
Page load time	Web server	Average time it takes to receive the page from the server
URL depth	URL	Number of slashes in the absolute path of the URL

B. Cambazoglu and R. Baeza-Yates, "Scalability Challenges in Web Search Engines"

Inverted index



B. Cambazoglu and R. Baeza-Yates, “Scalability Challenges in Web Search Engines”

Outline

1. Data structures
2. Inverted index
3. Constructing an index
4. Updating an index

Inverted index

1. Dictionary
   - Each entry contains
     - Number of pages containing the term
     - Pointer to the start of the inverted list
     - Other meta-data about the term
   - B+ tree, hash table
2. Inverted lists

Example

S_1 Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

S_2 Fishkeepers often use the term tropical fish to refer only those requiring fresh water, with saltwater tropical fish referred to as marine fish.

S_3 Tropical fish are popular aquarium fish, due to their often bright coloration.

S_4 In freshwater fish, this coloration typically derives from iridescence, while salt water fish are generally pigmented.

Croft et al., "Search Engines, Information Retrieval in Practice"

Document identifiers

and	1
aquarium	3
are	3 4
around	1
as	2
both	1
bright	3
coloration	3 4
derives	4
due	3
environments	1
fish	1 2 3 4
fishkeepers	2
found	1
fresh	2
freshwater	1 4
from	4
generally	4
in	1 4
include	1
including	1
iridescence	4
marine	2
often	2 3
only	2
pigmented	4
popular	3
refer	2
referred	2
requiring	2
salt	1 4
saltwater	2
species	1
term	2
the	1 2
their	3
this	4
those	2
to	2 3
tropical	1 2 3
typically	4
use	2
water	1 2 4
while	4
with	2
world	1

Croft et al., "Search Engines, Information Retrieval in Practice"

Frequencies

and	1:1			
aquarium	3:1			
are	3:1	4:1		
around	1:1			
as	2:1			
both	1:1			
bright	3:1			
coloration	3:1	4:1		
derives	4:1			
due	3:1			
environments	1:1			
fish	1:2	2:3	3:2	4:2
fishkeepers	2:1			
found	1:1			
fresh	2:1			
freshwater	1:1	4:1		
from	4:1			
generally	4:1			
in	1:1	4:1		
include	1:1			
including	1:1			
iridescence	4:1			
marine	2:1			
often	2:1	3:1		
only	2:1			
pigmented	4:1			
popular	3:1			
refer	2:1			
referred	2:1			
requiring	2:1			
salt	1:1	4:1		
saltwater	2:1			
species	1:1			
term	2:1			
the	1:1	2:1		
their	3:1			
this	4:1			
those	2:1			
to	2:2	3:1		
tropical	1:2	2:2	3:1	
typically	4:1			
use	2:1			
water	1:1	2:1	4:1	
while	4:1			
with	2:1			
world	1:1			

Croft et al., “Search Engines, Information Retrieval in Practice”

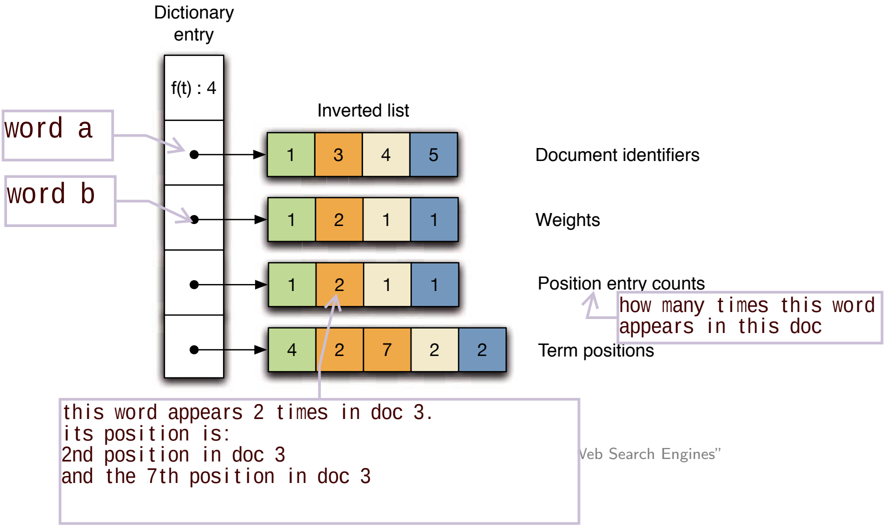
Positions

and	1,15				
aquarium	3,5				
are	3,3	4,14			
around	1,9				
as	2,21				
both	1,13				
bright	3,11				
coloration	3,12	4,5			
derives	4,7				
due	3,7				
environments	1,8				
fish	1,2	1,4	2,7	2,18	2,23
			3,2	3,6	4,3
			4,13		
fishkeepers	2,1				
found	1,5				
fresh	2,13				
freshwater	1,14	4,2			
from	4,8				
generally	4,15				
in	1,6	4,1			
include	1,3				
including	1,12				
iridescence	4,9				

marine	2,22				
often	2,2	3,10			
only	2,10				
pigmented	4,16				
popular	3,4				
refer	2,9				
referred	2,19				
requiring	2,12				
salt	1,16	4,11			
saltwater	2,16				
species	1,18				
term	2,5				
the	1,10	2,4			
their	3,9				
this	4,4				
those	2,11				
to	2,8	2,20	3,8		
tropical	1,1	1,7	2,6	2,17	3,1
typically	4,6				
use	2,3				
water	1,17	2,14	4,12		
while	4,10				
with	2,15				
world	1,11				

Croft et al., "Search Engines, Information Retrieval in Practice"

Full inverted index



word a

word b

how many times this word appears in this doc

this word appears 2 times in doc 3.
its position is:
2nd position in doc 3
and the 7th position in doc 3

Veb Search Engines”

Summary

- Inverted lists
    - Document identifiers
    - Frequencies
    - Positions
    - Weights

is a table which contains the following info

Outline

1 Data structures

2 Inverted index

3 Constructing an index

4 Updating an index

Simple indexer

```
procedure BuildIndex($D$)
    $I \leftarrow$ HashTable()
    $n \leftarrow 0$
    for all documents $d \in D$ do
        $n \leftarrow n + 1$
        $T \leftarrow$ Parse($d$)
        Remove duplicates from $T$
        for all tokens $t \in T$ do
            if $I_t \notin I$ then
                $I_t \leftarrow$ Array()
            end if
            $I_t$.append($n$)
        end for
    end for
    return $I$
end procedure
```

append doc to the inverted list

Croft et al., "Search Engines, Information Retrieval in Practice"

What are the problems with this simple indexer?

previous slide

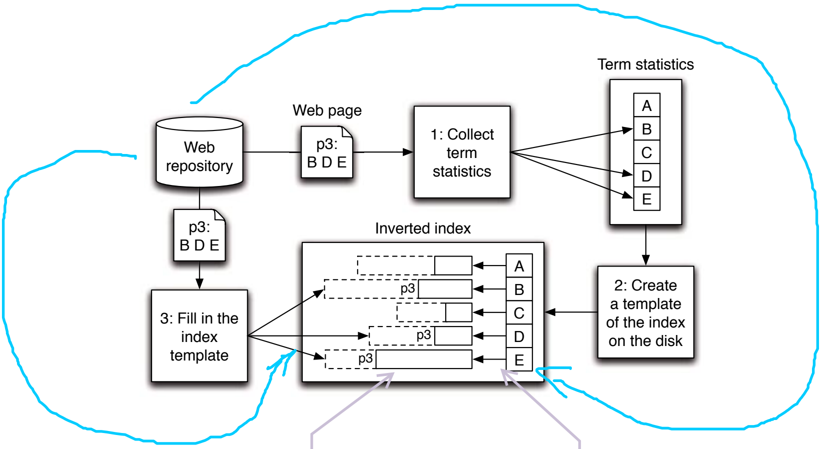
drawback1: not feasible for large collection which does not fit into memory
solution: there are two solutions. they both constructs inverted list and both not put inverted list into memory

1. In-memory
   - Two-pass index
   - One-pass index with merging
2. Single-threaded
   - Distributed indexing

drawback2: running slow for large collection

solution to drawback 2: change single threaded to distributed

Two-pass index


Web repository
Web page
p3:
B D E
1: Collect term statistics
Term statistics
A
B
C
D
E
p3:
B D E
Inverted index
A
B
C
D
E
p3
p3
p3
2: Create a template of the index on the disk
3: Fill in the index template

B. Camb

a list of which doc does term E occurs

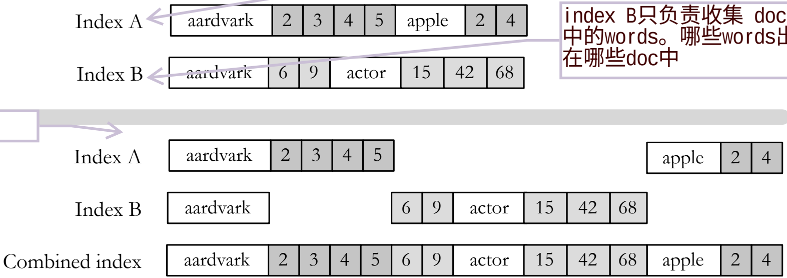
A,B,C are terms or words

One-pass index with merging

step 1

index A只负责收集 doc1-5中的words。哪些words出现在哪些doc中

index B只负责收集 doc5-70中的words。哪些words出现在哪些doc中



Index A: aardvark 2 3 4 5 apple 2 4

Index B: aardvark 6 9 actor 15 42 68

step 2

Index A: aardvark 2 3 4 5 … apple 2 4

Index B: aardvark … 6 9 actor 15 42 68

Combined index: aardvark 2 3 4 5 6 9 actor 15 42 68 apple 2 4

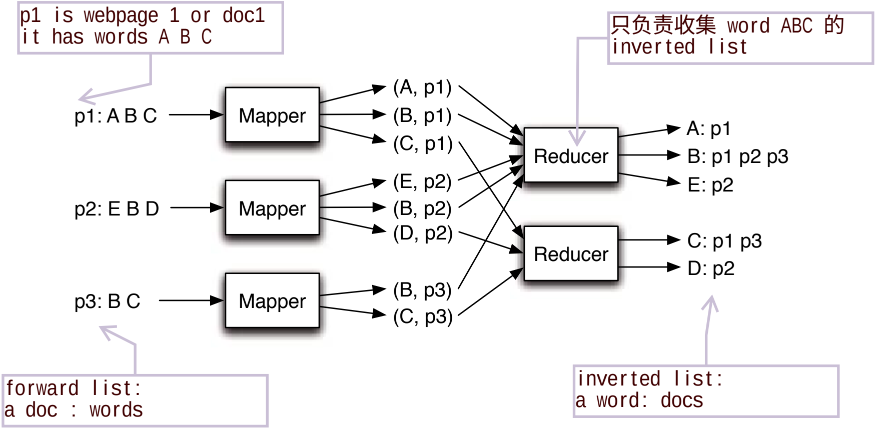
merge index A and B

Croft et al., “Search Engines, Information Retrieval in Practice”

Aardvark

Picture taken from https://en.wikipedia.org/wiki/Aardvark

Distributed indexing (MapReduce)



B. Cambazoglu and R. Baeza-Yates, “Scalability Challenges in Web Search Engines”

Summary

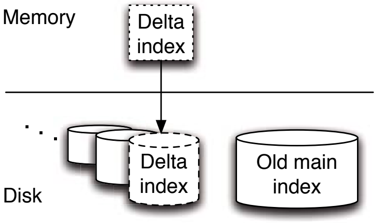
1. In-memory problem
    - Two-pass index
    - One-pass index with merging
2. Single-threaded problem
    - Distributed indexing

Outline

1. Data structures
2. Inverted index
3. Constructing an index
4. Updating an index

how to update an inverted index when we have a new webpage added, new doc added, or doc deleted.

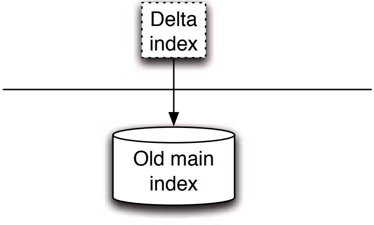
No merge



- Low index maintenance cost
- High query processing cost

B. Cambazoglu and R. Baeza-Yates, "Scalability Challenges in Web Search Engines"

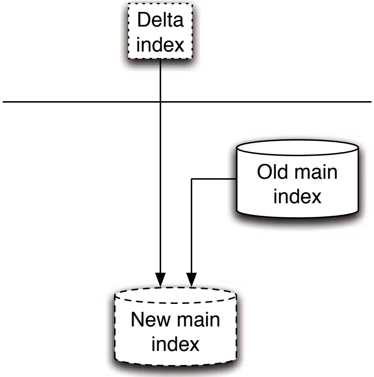
Incremental update



- Keeps free buffer space
- No read/write of entire index when updating
- Inverted lists are accessed concurrently
- Run out of free buffer space

B. Cambazoglu and R. Baeza-Yates, "Scalability Challenges in Web Search Engines"

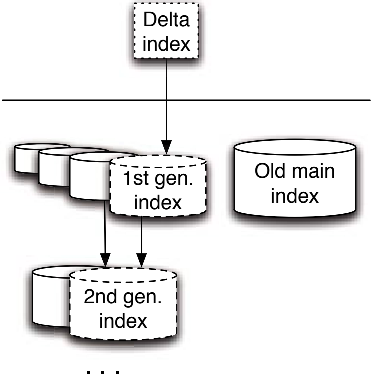
Immediate merge (in-memory)



- Always a single index
- Read/write of entire index when updating

B. Cambazoglu and R. Baeza-Yates, "Scalability Challenges in Web Search Engines"

Lazy merge



- Trade-off between index maintenance cost and query processing cost

B. Cambazoglu and R. Baeza-Yates, "Scalability Challenges in Web Search Engines"

Page deletions

- Maintain identifiers of deleted documents in memory, access during query processing
- Garbage collection (e.g., during index merging)

Summary

- Updating strategies
  - No merge
  - Incremental update
  - Immediate merge
  - Lazy merge
- Page deletions

Summary

1. Data structures
2. Inverted index
3. Constructing an index
4. Updating an index

Materials

- Croft et al., Chapter 5
- Manning et al., Chapters 1.2, 2.3–2.4, 3.1–3.2, 4, 5
- B. Barla Cambazoglu and Ricardo Baeza-Yates
  Scalability Challenges in Web Search Engines
  Morgan & Claypool Publishers, 2017