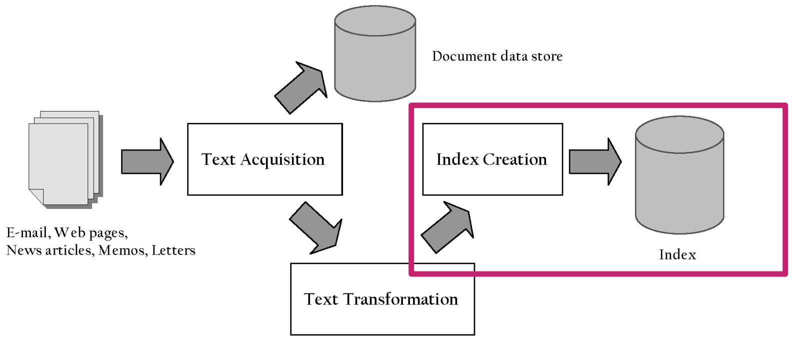
Information Retrieval

INFO 4300 / CS 4300

- Last classes
  - Text transformation
- Next topics
  - Indexing
    - Index construction
    - Compression
    - Ranking model

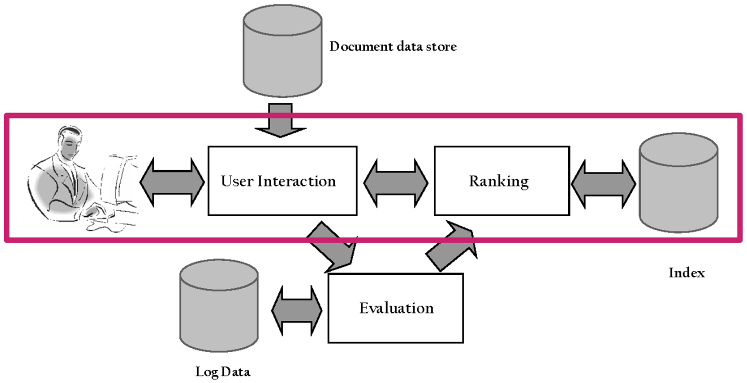
Indexing Process



Indexes

- ***Indexes*** are a specialized data structure designed to make search faster
- Most common: ***inverted index***
  - general name for a class of structures
  - "inverted" because documents are associated with words, rather than words with documents
  - at the core of all modern web search engines
  - support well over 500,000,000 queries/day

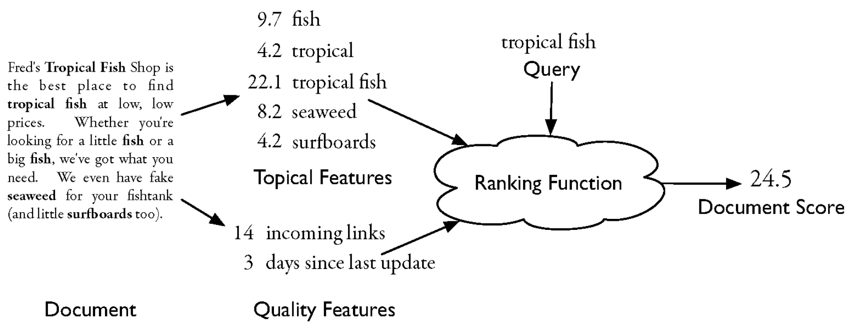
Query Process



Indexes and Ranking

- Indexes are designed to support ***search***
  - faster response time, supports updates
- Text search engines use a particular form of search: ***ranking***
  - documents are retrieved in sorted order according to a score computed using the document representation, the query, and a ***ranking algorithm***
- What is a reasonable abstract model for ranking?
  - lets us discuss indexes without details of the retrieval model

Abstract Model of Ranking

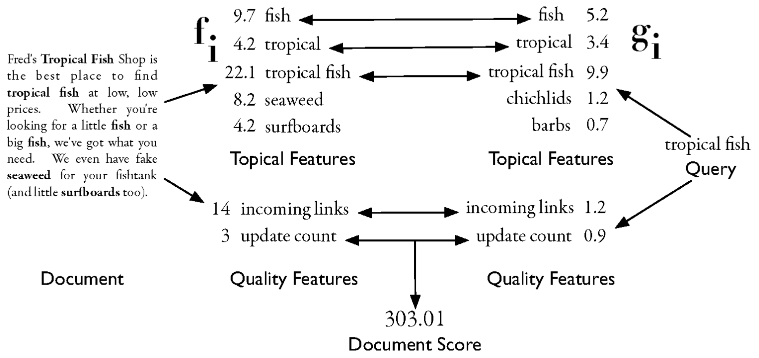


Fred's **Tropical Fish** Shop is the best place to find **tropical fish** at low, low prices. Whether you're looking for a little **fish** or a big **fish**, we've got what you need. We even have fake **seaweed** for your fishtank (and little **surfboards** too).

More Concrete Model

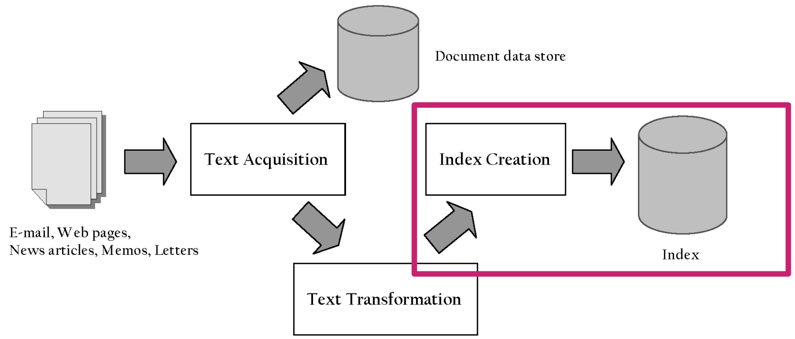
$$R(Q,D) = \sum_i g_i(Q) f_i(D)$$

f_i is a document feature function
g_i is a query feature function



Fred's **Tropical Fish** Shop is the best place to find **tropical fish** at low, low prices. Whether you're looking for a little **fish** or a big **fish**, we've got what you need. We even have fake **seaweed** for your fishtank (and little **surfboards** too).

Back to index construction...



Inverted Index

- Each index term is associated with an ***inverted list***
  - Contains lists of documents, or lists of word occurrences in documents, and other information
  - Each entry is called a ***posting***
  - The part of the posting that refers to a specific document or location is called a ***pointer***
  - Each document in the collection is given a unique number
  - Lists are usually ***document-ordered*** (sorted by document number)

Example "Collection"

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.

Four sentences from the Wikipedia entry for *tropical fish*

Simple Inverted Index

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			

Inverted Index with counts

- supports better ranking algorithms

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			

Inverted Index with positions

- supports proximity matches

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				

Proximity Matches

- Matching phrases or words within a window
  - e.g., "`tropical fish`", or “find tropical within 5 words of fish”
- Word positions in inverted lists make these types of query features efficient
  - e.g.,

tropical	1,1		1,7	2,6	2,17		3,1			
fish	1,2	1,4		2,7	2,18	2,23	3,2	3,6	4,3	4,13

Fields and Extents

- Document structure is useful in search
  - ***field*** restrictions
    - » e.g., date, from:, etc.
  - some fields more important
    - » e.g., title, headings
- Options:
  - separate index (set of inverted lists) for each field type
  - add information about fields to postings
  - use ***extent lists***

Extent Lists

- An ***extent*** is a contiguous region of a document
  - represent extents using word positions
  - inverted list records all extents for a given field type
  - e.g.,

fish	1,2	1,4		2,7	2,18	2,23	3,2	3,6	4,3	4,13
title	1:(1,3)		2:(1,5)							4:(9,15)

extent list

Other Issues

- Precomputed scores in inverted list
  - e.g., list for “fish” [(1:3.6), (3:2.2)], where 3.6 is total feature value for document 1
  - improves speed but reduces flexibility
- Score-ordered lists
  - query processing engine can focus only on the top part of each inverted list, where the highest-scoring documents are recorded
  - very efficient for single-word queries

Auxiliary Structures

- Inverted lists usually stored together in a single file for efficiency
  - *Inverted file*
- *Vocabulary* or *lexicon*
  - Contains a lookup table from index terms to the byte offset of the inverted list in the inverted file
  - Either hash table in memory or B-tree for larger vocabularies
- Term statistics stored at start of inverted lists
- Collection statistics stored in separate file