DAT630
Search Engines

Search Engines, Chapters 1, 2, 5

28/09/2016

Krisztian Balog | University of Stavanger

Information Retrieval

Information Retrieval (IR)

> "Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information."
>
> (Salton, 1968)

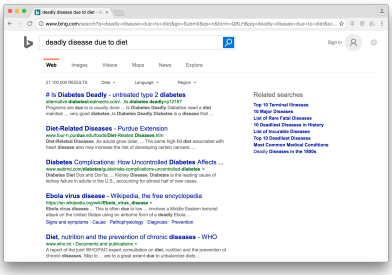
Searching in Databases

- Query: records with balance > $50,000 in branches located in Amherst, MA.

Name	Branch	Balance
Sam I. Am	Amherst, MA	$95,342.11
Patty MacPatty	Amherst, MA	$23,023.23
Bobby de West	Amherst, NY	$78,000.00
Xing O'Boston	Boston, MA	$50,000.01

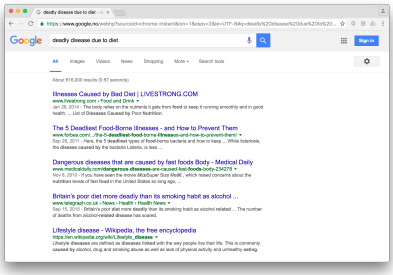
Searching in Text

- Query: *deadly disease due to diet*
- Which are relevant?



Searching in Text

- Query: *deadly disease due to diet*
- Which are relevant?



Comparing Text

- Comparing the query text to the document text and determining what is a good match is the core issue of information retrieval
- Exact matching of words is not enough
  - Many different ways to write the same thing in a "natural language" like English
  - E.g., does a news story containing the text "fatal illnesses caused by your menu" match the query?
  - Some documents will be better matches than others

Dimensions of IR

- IR is more than just text, and more than just web search
  - Although these are central
- **Content**
  - Text
  - Images
  - Video
  - Audio
  - Scanned documents

Dimensions of IR

- **Applications**
  - Web search
  - Vertical search
  - Enterprise search
  - Mobile search
  - Social search
  - Desktop search
  - Literature search
  - …

Dimensions of IR

- **Tasks**
  - Ad-hoc search
  - Filtering
  - Classification
  - Question answering

Core issues in IR

- **Relevance**
  - Simple (and simplistic) definition: A relevant document contains the information that a person was looking for when they submitted a query to the search engine
  - Many factors influence a person's decision about what is relevant: e.g., task, context, novelty
  - *Topical relevance* (same topic) vs. *user relevance* (everything else)

Core issues in IR

- **Relevance**
  - *Retrieval models* define a view of relevance
  - *Ranking algorithms* used in search engines are based on retrieval models
  - Most models based on statistical properties of text rather than linguistic
    - I.e., counting simple text features such as words instead of parsing and analyzing the sentences

Core issues in IR

- **Evaluation**
  - Experimental procedures and measures for comparing system output with user expectations
  - Typically use test collection of documents, queries, and relevance judgments
  - *Recall* and *precision* are two examples of effectiveness measures

Core issues in IR

- **Information needs**
  - Keyword queries are often poor descriptions of actual information needs
  - Interaction and context are important for understanding user intent
  - Query refinement techniques such as query expansion, query suggestion, relevance feedback improve ranking

Search Engines in Operational Environments

- Performance
  - Response time, indexing speed, etc.
- Incorporating new data
  - Coverage and freshness
- Scalability
  - Growing with data and users
- Adaptibility
  - Tuning for specific applications

Search Engine Architecture

Search Engine Architecture

- A software architecture consists of software components, the interfaces provided by those components, and the relationships between them
  - Describes a system at a particular level of abstraction
- Architecture of a search engine determined by 2 requirements
  - Effectiveness (quality of results)
  - Efficiency (response time and throughput)

Indexing Process

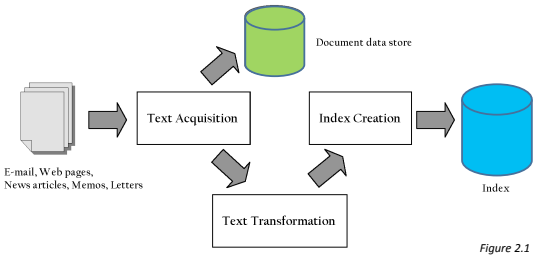


Figure 2.1

Indexing Process

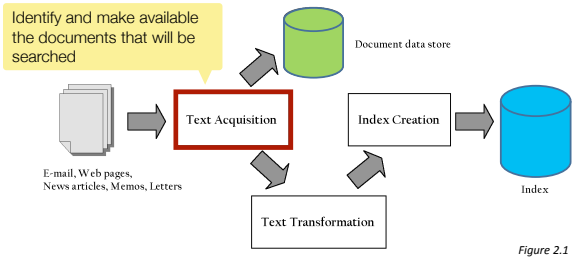


Figure 2.1

Text Acquisition

- **Crawler**
  - Identifies and acquires documents for search engine
  - Many types: web, enterprise, desktop, etc.
  - Web crawlers follow links to find documents
    - Must efficiently find huge numbers of web pages (*coverage*) and keep them up-to-date (*freshness*)
    - Single site crawlers for site search
    - *Topical* or *focused* crawlers for vertical search
  - Document crawlers for enterprise and desktop search
    - Follow links and scan directories

Text Acquisition

- **Feeds**
  - Real-time streams of documents
  - E.g., web feeds for news, blogs, video, radio, TV
  - RSS is common standard
  - RSS “reader” can provide new XML documents to search engine

Text Acquisition

- Documents need to be **converted** into a consistent text plus metadata format
  - E.g. HTML, XML, Word, PDF, etc. → XML
- Convert text encoding for different languages
  - Using a Unicode standard like UTF-8

Indexing Process

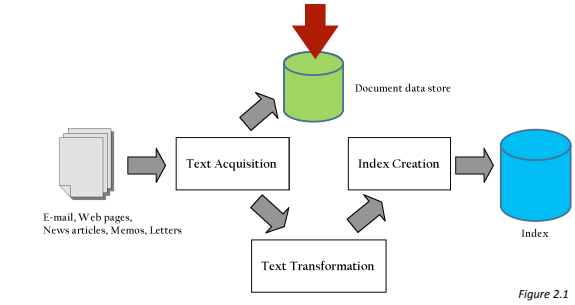


Figure 2.1

Document Data Store

- Stores text, metadata, and other related content for documents
  - Metadata is information about document such as type and creation date
  - Other content includes links, anchor text
- Provides fast access to document contents for search engine components
  - E.g. result list generation
- Could use relational database system
  - More typically, a simpler, more efficient storage system is used due to huge numbers of documents

Indexing Process

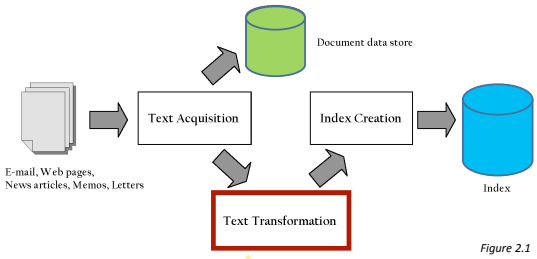


Figure 2.1

Transform documents into index terms or features

Text Transformation

- Tokenization
- Stopword removal
- Stemming
- Information extraction
  - Identify index terms that more complex than single words
    - E.g., named entity recognizers identify classes such as people, locations, companies, dates, etc
  - Important for some applications

Text Transformation

- Link Analysis
  - Makes use of links and anchor text in web pages
  - Link analysis identifies popularity and community information
    - E.g., PageRank
  - Anchor text can significantly enhance the representation of pages pointed to by links
    - Significant impact on web search
    - Less importance in other applications

Text Transformation

- Classification
  - Identifies class-related metadata for documents or part of documents
    - Topics, reading levels, sentiment, genre
    - Spam vs. non-spam
    - Non-content parts of documents, e.g., advertisements
  - Use depends on application

Indexing Process

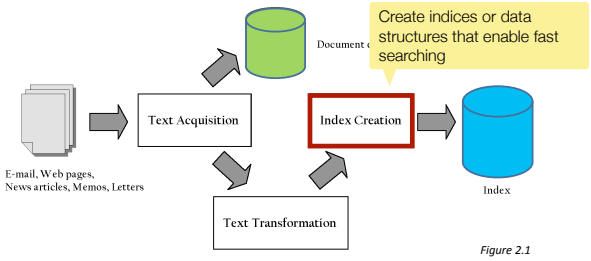


Figure 2.1

Create indices or data structures that enable fast searching

Index Creation

- Document Statistics
  - Gathers counts and positions of words and other features
  - Used in ranking algorithm
- Weighting
  - Computes weights for index terms
  - Usually reflect “importance” of term in the document
  - Used in ranking algorithm

Index Creation

- Inversion
  - Core of indexing process
  - Converts document-term information to term-document for indexing
    - Difficult for very large numbers of documents
  - Format of inverted file is designed for fast query processing
    - Must also handle updates
    - Compression used for efficiency

Index Creation

- Index Distribution
  - Distributes indexes across multiple computers and/or multiple sites
  - Essential for fast query processing with large numbers of documents
  - Many variations
    - Document distribution, term distribution, replication
  - P2P and distributed IR involve search across multiple sites

Query Process

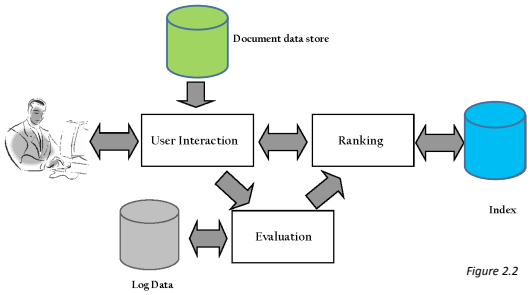


Figure 2.2

Query Process

Interface between the person doing the searching and the search engine

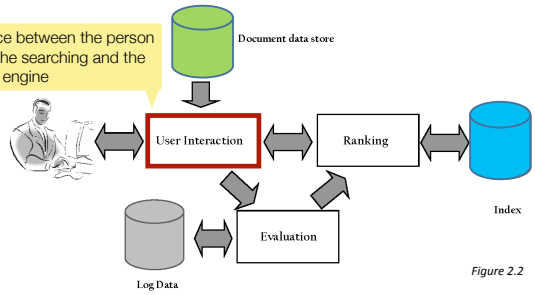


Figure 2.2

User Interaction

- Accepting the user's query and transforming it into index terms
- Taking the ranked list of documents from the search engine and organizing it into the results shown to the user
  - E.g., generating snippets to summarize documents
- Range of techniques for refining the query (so that it better represents the information need)

User Interaction

- Query input
  - Provides interface and parser for *query language*
  - Query language used to describe complex queries
    - *Operators* indicate special treatment for query text
  - Most web search query languages are very simple
    - Small number of operators
  - There are more complicated query languages
    - E.g., Boolean queries, proximity operators
    - IR query languages also allow content and structure specifications, but focus on content

User Interaction

- Query transformation
  - Improves initial query, both before and after initial search
  - Includes text transformation techniques used for documents
  - *Spell checking* and *query suggestion* provide alternatives to original query
    - Techniques often leverage query logs in web search
  - *Query expansion* and *relevance feedback* modify the original query with additional terms

User Interaction

- Results output
  - Constructs the display of ranked documents for a query
  - Generates *snippets* to show how queries match documents
  - *Highlights* important words and passages
  - Retrieves appropriate *advertising* in many applications ("related" things)
  - May provide *clustering* and other visualization tools

Query Process

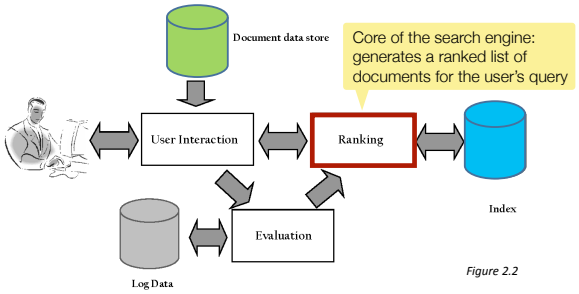


Figure 2.2

Ranking

- Scoring
  - Calculates scores for documents using a *ranking algorithm*, which is based on a *retrieval model*
  - Core component of search engine
  - Basic form of score is $\sum_i q_i d_i$
    - q_i and d_i are query and document term weights for term i
  - Many variations of ranking algorithms and retrieval models

Ranking

- Performance optimization
  - Designing ranking algorithms for efficient processing
    - *Term-at-a time* vs. *document-at-a-time* processing
    - *Safe* vs. *unsafe* optimizations
- Distribution
  - Processing queries in a distributed environment
  - *Query broker* distributes queries and assembles results
  - *Caching* is a form of distributed searching

Query Process

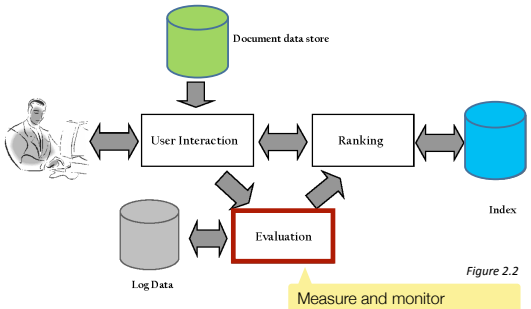


Figure 2.2

Measure and monitor effectiveness and efficiency. Record and analyze usage data

Evaluation

- Logging
  - Logging user queries and interaction is crucial for improving search effectiveness and efficiency
  - *Query logs* and *clickthrough data* used for query suggestion, spell checking, query caching, ranking, advertising search, and other components
- Ranking analysis
  - Measuring and tuning ranking effectiveness
- Performance analysis
  - Measuring and tuning system efficiency

Indexing

Indices

- Indices are data structures designed to make search faster
- Text search has unique requirements, which leads to unique data structures
- Most common data structure is the *inverted index*
  - General name for a class of structures
  - "Inverted" because documents are associated with words, rather than words with documents
    - Similar to a concordance

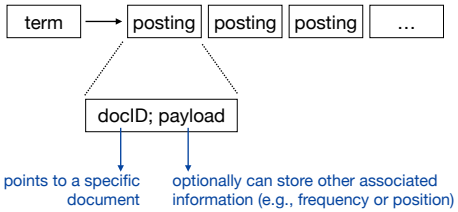
Index

Note: *italic* page numbers indicate specific methods, whilst **bold** page numbers indicate major sections on the subject.

Inverted Index

- Each index term is associated with a *postings list (*or *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 (*docID*)
  - The posting can store additional information, called the *payload*
  - Lists are usually *document-ordered* (sorted by docID)

Inverted Index



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.

Four sentences from the Wikipedia entry for *tropical fish*

Simple Inverted Index

docID

Each document that contains the term is a posting.

No additional payload.

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

Inverted Index with Counts

docID: freq

The payload is the frequency of the term in the document.

Supports better ranking algorithms.

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

Inverted Index with Positions

docID, position

There is a separate posting for each term occurrence in the document. The payload is the term position.

Supports proximity matches.
E.g., find "tropical" within 5 words of "fish"

and	1,15	marine	2,22
aquarium	3,5	often	2,2
are	3,3 4,14	only	2,10
around	1,9	pigmented	4,16
as	2,21	popular	3,4
both	1,13	refer	2,9
bright	3,11	referred	2,19
coloration	3,12 4,5	requiring	2,12
derives	4,7	salt	1,16
due	3,7	saltwater	2,16
environments	1,8	species	1,18
fish	1,2 1,4 2,7 2,18 2,23 3,2 3,6 4,3 4,13	term	2,5
		the	1,10
		their	3,9
fishkeepers	2,1	this	4,4
found	1,5	those	2,11
fresh	2,13	to	2,8
freshwater	1,14 4,2	tropical	1,1
from	4,8	typically	4,6
generally	4,15	use	2,3
in	1,6 4,1	water	1,17
include	1,3	while	4,10
including	1,12	with	2,15
iridescence	4,9	world	1,11

Issues

- Compression
  - Inverted lists are very large
  - Compression of indexes saves disk and/or memory space
- Optimization techniques to speed up search
  - Read less data from inverted lists
    - "Skipping" ahead
  - Calculate scores for fewer documents
    - Store highest-scoring documents at the beginning of each inverted list
- Distributed indexing

Exercise

- Draw the inverted index for the following document collection

Doc 1	new home sales top forecasts
Doc 2	home sales rise in july
Doc 3	increase in home sales in july
Doc 4	july new home sales rise

Solution

new	1 4
home	1 2 3 4
sales	1 2 3 4
top	1
forecasts	1
rise	2 4
in	2 3
july	2 3 4
increase	3