Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Natural Language Processing

Information Retrieval

Potsdam, 14 June 2012

Saeedeh Momtazi
Information Systems Group

based on the slides of the course book

Outline

Outline

Information Retrieval

- The most popular usages of computer (and Internet)
  - Search
  - Communication
- Information Retrieval (IR)
  - The field of computer science that is mostly involved with R&D for search
- Primary focus of IR is on text and documents
  - Mostly textual content
  - A bit structured data
    - Papers: title, author, date, publisher
    - Email: subject, sender, receiver, date

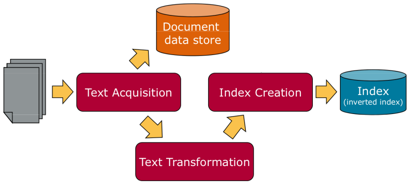
Fully structured data is normally covered by data mining

IR Dimensions

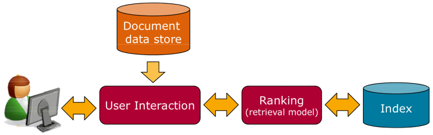
- Web search
  - Most common
    - Search engines
- Vertical search
  - Restricted domain/topic
    - Books, movies, suppliers
- Enterprise search
  - Corporate intranet
    - Emails, web pages, documentations, codes, wikis, tags, directories, presentations, spreadsheets
- Desktop search
  - Personal enterprise search
- P2P search
  - No centralized control
    - File sharing, shared locality
- Forum search
- ...

IR Architecture

- Indexing
  - Text Acquisition
  - Text Transformation
  - Index Creation

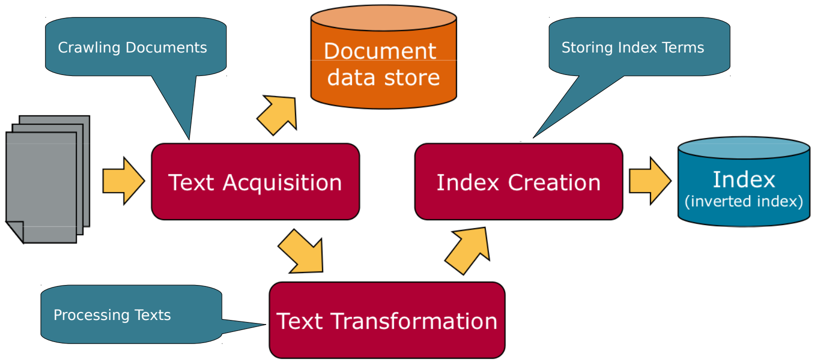


- Querying
  - User Interaction
  - Ranking



Outline

Indexing Architecture



Outline

Web Crawler

- Identifying and acquiring documents for search engine

- Following links to find documents
  - Finding huge numbers of web pages efficiently (coverage)
  - Keeping the crawled data up-to-date (freshness)

Outline

Text Processing

- Parsing
- Tokenizing
- Stopping
- Stemming

Parsing

- Recognizing structural elements
  - Titles
  - Links
  - Headings
  - ...

- Using the syntax of markup languages to identify structures

Tokenizing

- Basic model
  - Considering white-space as delimiter

- Main issues that should be considered and normalized
  - Capitalization
    - apple vs. Apple
  - Apostrophes
    - O'Conner vs. owner's
  - Hyphens
  - Non-alpha characters
  - Word segmentation in Chinese

Stopping

- Removing the stop words from documents

- Stop words: the most common words in a language
  - *and, or, the, in*
  - Around 400 stop words for English

- Advantages
  - Effective
    - Do not consider as important query terms to be searched
  - Efficient
    - Reduce storage size and time complexity

- Disadvantages
  - Problem with queries with higher impact of stop words
    - *To be or not to be*

Stemming

- Grouping words derived from a common stem
  - *Computer, computers, computing, compute*
  - *Fish, fishing, fisherman*

Outline

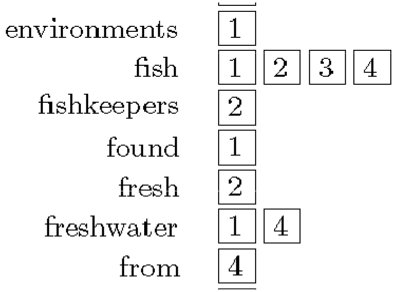
Index Storing

- Storing document-words information in an inverse format
  - Converting document-term statistics to term-document for indexing

- Increasing the efficiency of the retrieval engine

Index Storing

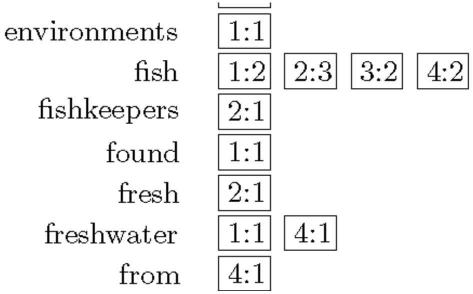
- Index-term with document ID



environments	1			
fish	1	2	3	4
fishkeepers	2			
found	1			
fresh	2			
freshwater	1	4		
from	4			

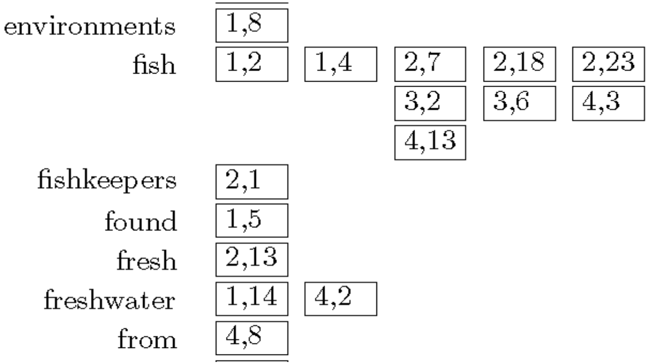
Index Storing

- Index-term with document ID and frequency



Index Storing

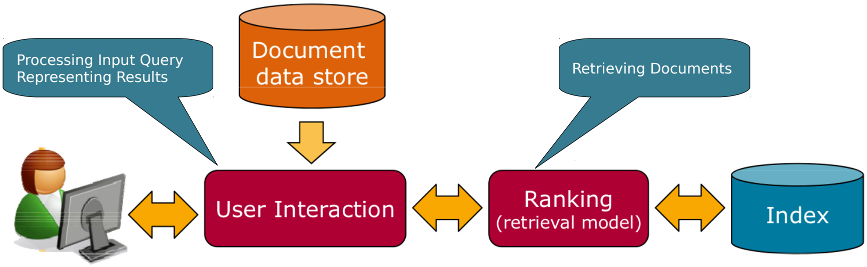
- Index-term with document ID and position



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				

Outline

Querying Architecture



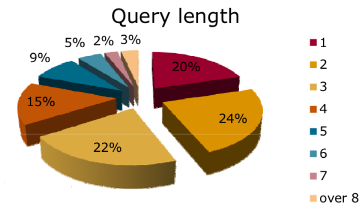
Outline

Query Processing

- Applying similar techniques used in text processing for documents
  - Tokenization, stopping, stemming
- Spell checking
  - Correcting the query if there exists any spelling error in it
- Query suggestion
  - Providing alternatives words to the original query (based on query logs)
- Query expansion and relevance feedback
  - Modifying the original query with additional terms

Query Expansion

- Short search queries are under-specified



- Keyword queries are often poor descriptions of actual information need

car inventor

Nicolas Joseph Cugnot built the first automobile

Outline

Retrieval Models

- Calculating the scores for documents using a ranking algorithm
  - Boolean model
  - Vector space model
  - Probabilistic model
  - Language model

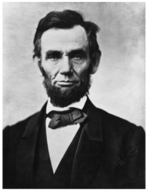
Boolean Model

- Two possible outcomes for query processing
  - TRUE or FALSE
  - All matching documents are considered equally relevant

- Query usually specified using Boolean operators
  - AND, OR, NOT

Boolean Model

- Search for news articles about President Lincoln

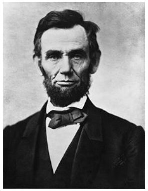


lincoln

Result:
cars
places
people

Boolean Model

- Search for news articles about President Lincoln

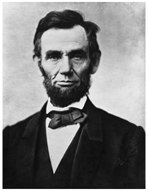


president AND lincoln

Result:
“Ford Motor Company today announced that Darryl Hazel will succeed Brian Kelley as president of Lincoln Mercury ”

Boolean Model

- Search for news articles about President Lincoln

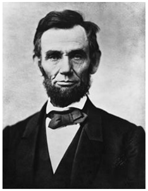


president AND lincoln AND NOT (automobile OR car)

Not in result:
"President Lincoln's body departs Washington in a nine-car funeral train."

Boolean Model

- Search for news articles about President Lincoln



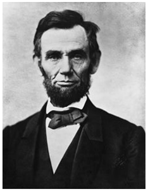
president AND lincoln AND biography AND life AND birthplace AND gettysburg AND NOT (automobile OR car)

Result:

$\varnothing$

Boolean Model

- Search for news articles about President Lincoln



president AND lincoln AND (biography OR life OR birthplace OR gettysburg) AND NOT (automobile OR car)

Top result might be:
“President’s Day - Holiday activities - crafts, mazes, mazes word searches, ... ‘The Life of Washington’ Read the entire searches The Washington book online! Abraham Lincoln Research Site ...”

Boolean Model

- Advantages
  - Results are predictable and relatively easy to explain
- Disadvantages
  - Relevant documents have no order
  - Complex queries are difficult to write

Retrieval Models

- Calculating the scores for documents using a ranking algorithm
  - Boolean model
  - Vector space model
  - Probabilistic model
  - Language model

Vector Space Model

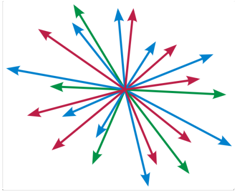
- Very popular model, even today
- Documents and query represented by a vector of term weights
  - t is number of index terms (i.e., very large)

$$D_i = (d_{i1}, d_{i2}, ..., d_{it})$$

$$Q = (q_1, q_2, ..., q_t)$$

- Collection represented by a matrix of term weights

$$\begin{array}{lllll} & Term_1 & Term_2 & \dots & Term_t \\ Doc_1 & d_{11} & d_{12} & \dots & d_{1t} \\ Doc_2 & d_{21} & d_{22} & \dots & d_{2t} \\ \vdots & \vdots & & & \\ Doc_n & d_{n1} & d_{n2} & \dots & d_{nt} \end{array}$$



Vector Space Model

- Ranking each document by distance between points representing query and document
  - Using cosine similarity
    - Cosine of angle between document and query vectors
    - Normalized dot-product

$$Cosine(D_i, Q) = \frac{\sum_{j=1}^{t} d_{ij} \times q_j}{\sqrt{\sum_j d_{ij}^2}\sqrt{\sum_j q_j^2}}$$

- Having no explicit definition of relevance as a retrieval model
  - Implicit: Closer documents are more relevant.

Vector Space Model

- Term frequency weight (*tf*)
  - Measuring the importance of term *t* in document *i*

$$tf_{ik} = \frac{f_{ik}}{\sum_{j=1}^{t} f_{ij}}$$

- Inverse document frequency (*idf*)
  - Measuring importance of the term *t* in collection

$$idf_k = \log \frac{N}{n_k}$$

Final weighting: multiplying *tf* and *idf*, called *tf.idf*

Vector Space Model

- Advantages
  - Simple computational framework for ranking
  - Any similarity measure or term weighting scheme can be used
- Disadvantages
  - Assumption of term independence

Retrieval Models

- Calculating the scores for documents using a ranking algorithm
  - Boolean model
  - Vector space model
  - Probabilistic model
  - Language model

Language Model

- Representing each document as a language model
  - i.e., as a distribution over words

- Possibilities of using language models for retrieval
  - Query likelihood model
    - Probability of generating the query text from a document language model
  - Document likelihood model
    - Probability of generating the document text from a query language model
  - Kullback-Leibler divergence
    - Comparing the language models representing the query and the document context

Query Likelihood Model

- Ranking documents based on the probability that the query could be generated by the document model

- Unigram model as the simplest model

$$P(Q|D) = \prod_{i=1}^{n} P(q_i|D)$$

$$P(q_i|D) = \frac{\#(q_i, D)}{|D|}$$

Query Likelihood Model

$$P(Q|D) = \prod_{i=1}^{n} P(q_i|D)$$

- Zero probability if query word does not appear in the document
  $\Rightarrow$ Smoothing
- Assumption of term independence
  $\Rightarrow$ Higher order *n*-grams: bigram, trigram, ...

Outline

Results Output

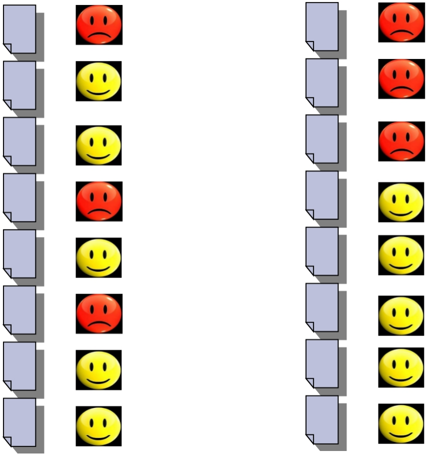
- Constructing the display of ranked documents for a query
- Generating snippets to show how queries match documents
- Highlighting important words and passages
- Providing clustering (if required)

Outline

Evaluation Metrics

- Evaluation of unranked sets
  - Precision
  - Recall
  - F-measure
  - Accuracy

Unranked vs Ranked Sets



Evaluation Metrics

- Evaluation of ranked sets
  - Precision-recall curve
  - Mean average precision
  - Precision at *n*
  - Mean reciprocal rank

Average Precision

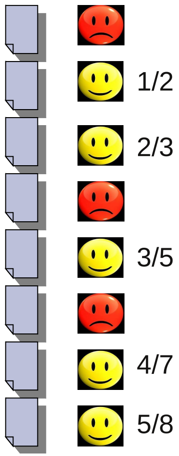
- Average precision
  - Calculating precision of the system after retrieving each relevant document
  - Averaging these precision values

- Mean average precision
  - Reporting the mean of the average precisions over all queries in the query set

$$AveragePrecision = \frac{1}{K}\sum_{k=1}^{K} P@k$$

k is the rank of relevant documents in the retrieved list

Average Precision



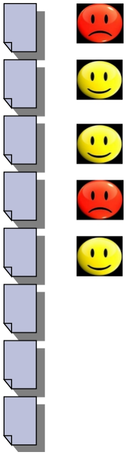
$$AP = \frac{(\frac{1}{2}+\frac{2}{3}+\frac{3}{5}+\frac{4}{7}+\frac{5}{8})}{5} = 0.5926$$

Precision at *n*

- Being used by people who want to receive good results at the first *n* items of the retrieved documents

- Calculating precision at *n*
  - Considering the top *n* documents only
  - Ignoring the rest documents

Precision at 5



$$P@5 = \frac{3}{5} = 0.60$$

Reciprocal Rank

- Being used by people who need only one correct answer
  (no matter how many relevant items are available in the retrieved list, as soon as the first correct item appears, the user is satisfied with the results.)

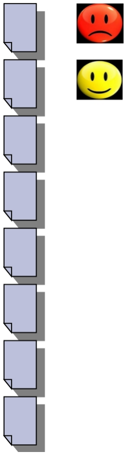
- Reciprocal Rank
  - Inversing the score of the rank at which the first correct answer is returned

- Mean Reciprocal Rank
  - Reporting the mean of the reciprocal rank over all queries in the query set

$$ReciprocalRank = \frac{1}{R}$$

R is the position of the first correct item in the ranked list

Reciprocal Rank



$$ReciprocalRank = \frac{1}{2} = 0.50$$

Further Reading

- Search Engines
  Information Retrieval in Practice
  Croft, Metzler, Strohman

- Introduction to Information Retrieval
  Manning, Raghavan, Schütze