Natural Language Processing
SoSe 2014

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Information Retrieval

Dr. Mariana Neves

June 18th, 2014

(based on the slides of Dr. Saeedeh Momtazi)

Outline

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

Outline

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

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

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

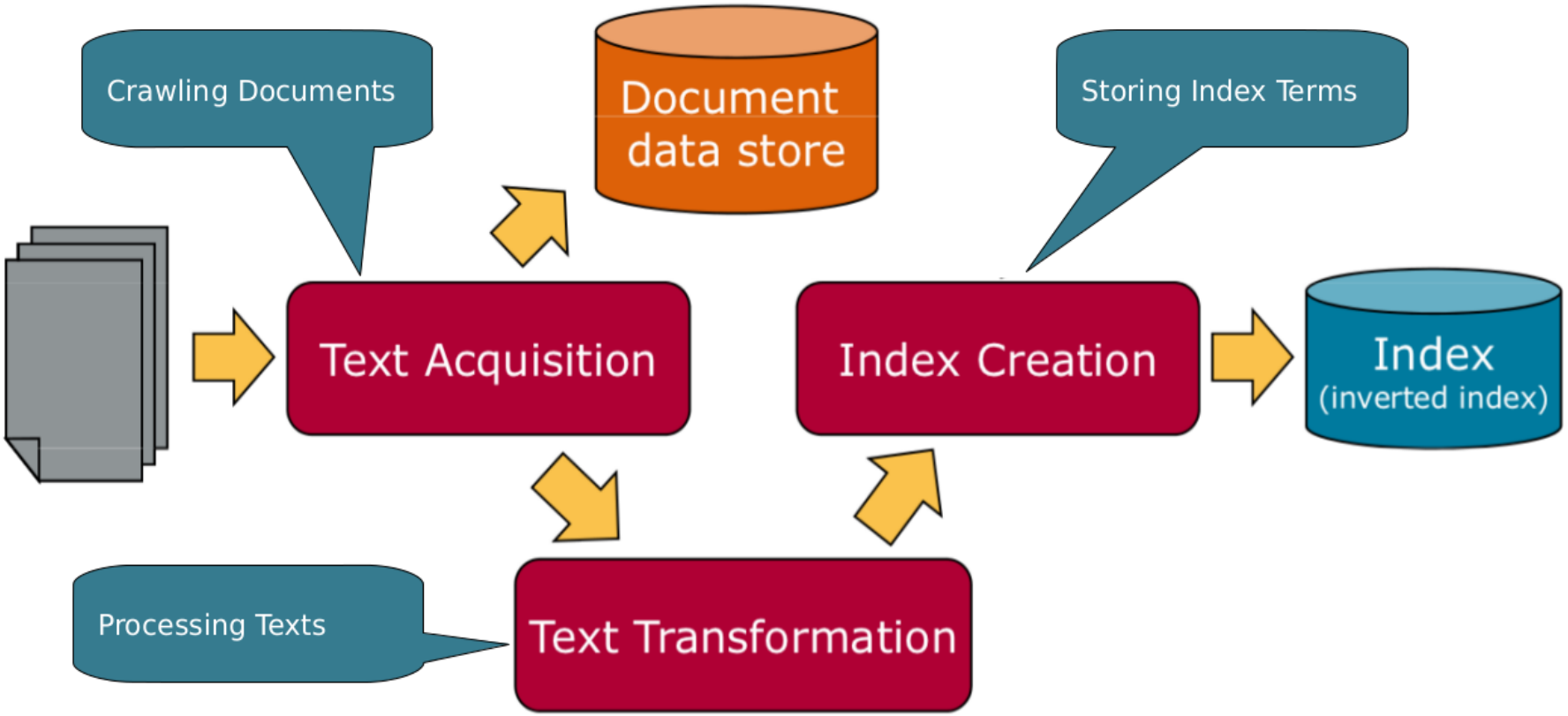
- Indexing
  - Text Acquisition
  - Text Transformation
  - Index Creation

- Querying
  - User Interaction
  - Ranking

Outline

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

Indexing Architecture



(figure taken from the slides of Dr. Saeedeh Momtazi)

Outline

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

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

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

Text Processing

- Parsing
- Tokenizing
- Stopword filtering
- Stemming

Parsing

- Recognizing structural elements
  - Titles
  - Links
  - Headings
  - ...
- Using the syntax of markup languages to identify structures

Tokenization

- 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 (e.g., in Chinese or German)

Stopwords filtering

- 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

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

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

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			

Index Storing

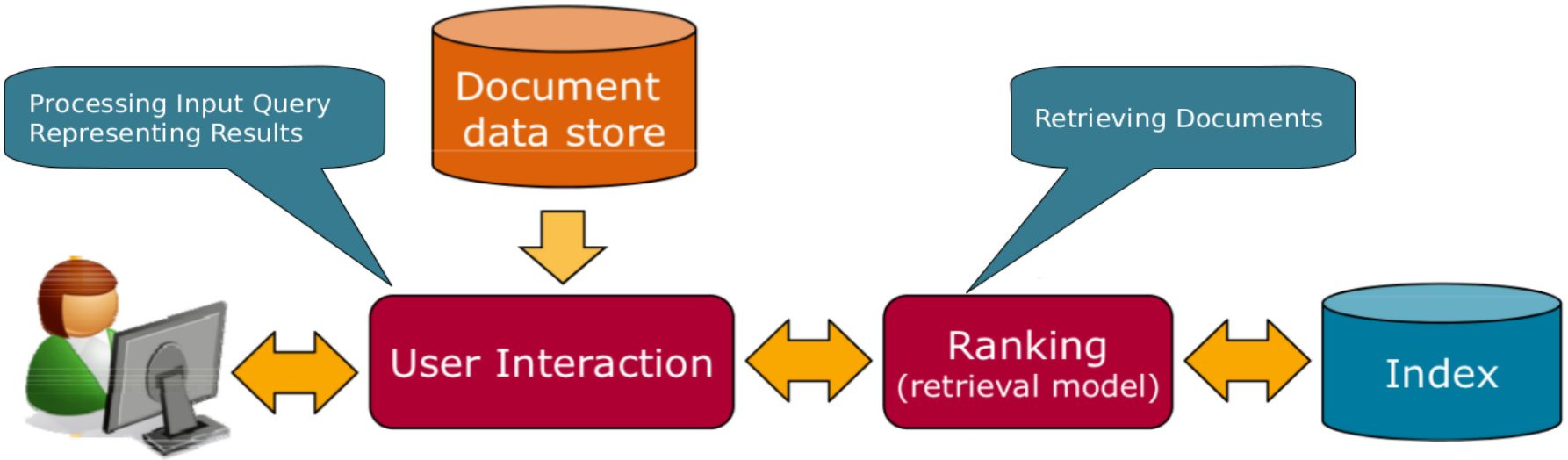
- Index-term with document ID and position

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

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

Querying Architecture



(figure taken from the slides of Dr. Saeedeh Momtazi)

Outline

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

Query Processing

- Applying similar techniques used in text processing for documents
  - Tokenization, stopwords filtering, 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
  - http://www.tamingthebeast.net/blog/web-marketing/search-query-lengths.h
    - 26.45% (1 word), 23.66% (2 words), 19.34 (3 words), 13.17% (4 words), 7.69% (5 words), 4.12% (6 words), 2.26% (7 words)
  - Google:
    - „54.5% of user queries are greater than 3 words“
- Keyword queries are often poor descriptions of actual information need
  - car inventor
  - Nicolas Joseph Cugnot built the first automobile

Outline

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

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
    - cars
    - places
    - people

Boolean Model

- Search for news articles about President Lincoln
  - president AND lincoln
    - “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)
    - "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 OR life OR birthplace OR gettysburg) AND NOT (automobile OR car)
    - "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

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

	$Term_1$	$Term_2$	...	$Term_t$
Doc_1	d_{11}	d_{12}	...	d_{1t}
Doc_2	d_{21}	d_{22}	...	d_{2t}
...	...	...		...
Doc_n	d_{n1}	d_{n2}	...	d_{nt}

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

- 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

Outline

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

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

- Introduction
- Indexing Block
  - Document Crawling
  - Text Processing
  - Index Storing
- Querying Block
  - Query Processing
  - Retrieval Models
  - Result Representation
- Evaluation

Evaluation Metrics

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

Evaluation Metrics

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

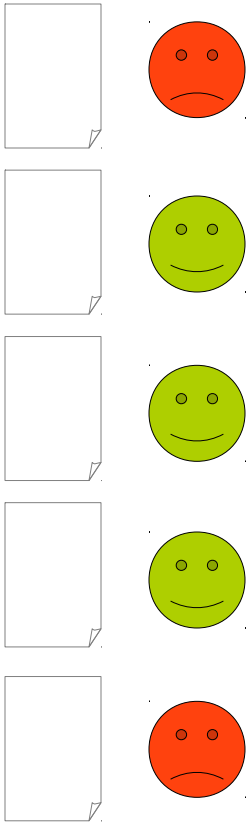
Average Precision

- 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

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

(k is the rank of relevant documents in the retrieved list)

Average Precision



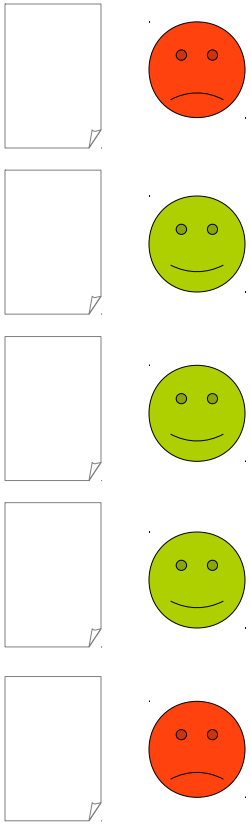
$$AP = \frac{\frac{1}{2} + \frac{2}{3} + \frac{3}{4}}{3} = 0.64$$

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 of documents

Precision at 5



$$P@5 = \frac{3}{5} = 0.6$$

Reciprocal Rank

- Being used by people who need only one correct answer

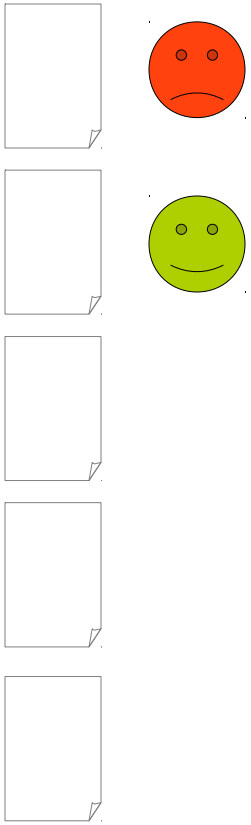
  (no matter how many relevant items are available in the retrieved list, as soon as the first correct item appears, the user is satisfied with the results.)

- 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



$$ReciprocalRank = \frac{1}{2} = 0.5$$

Further Reading