Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Search Engines
Chapter 2 – Architecture

14.4.2011

Felix Naumann

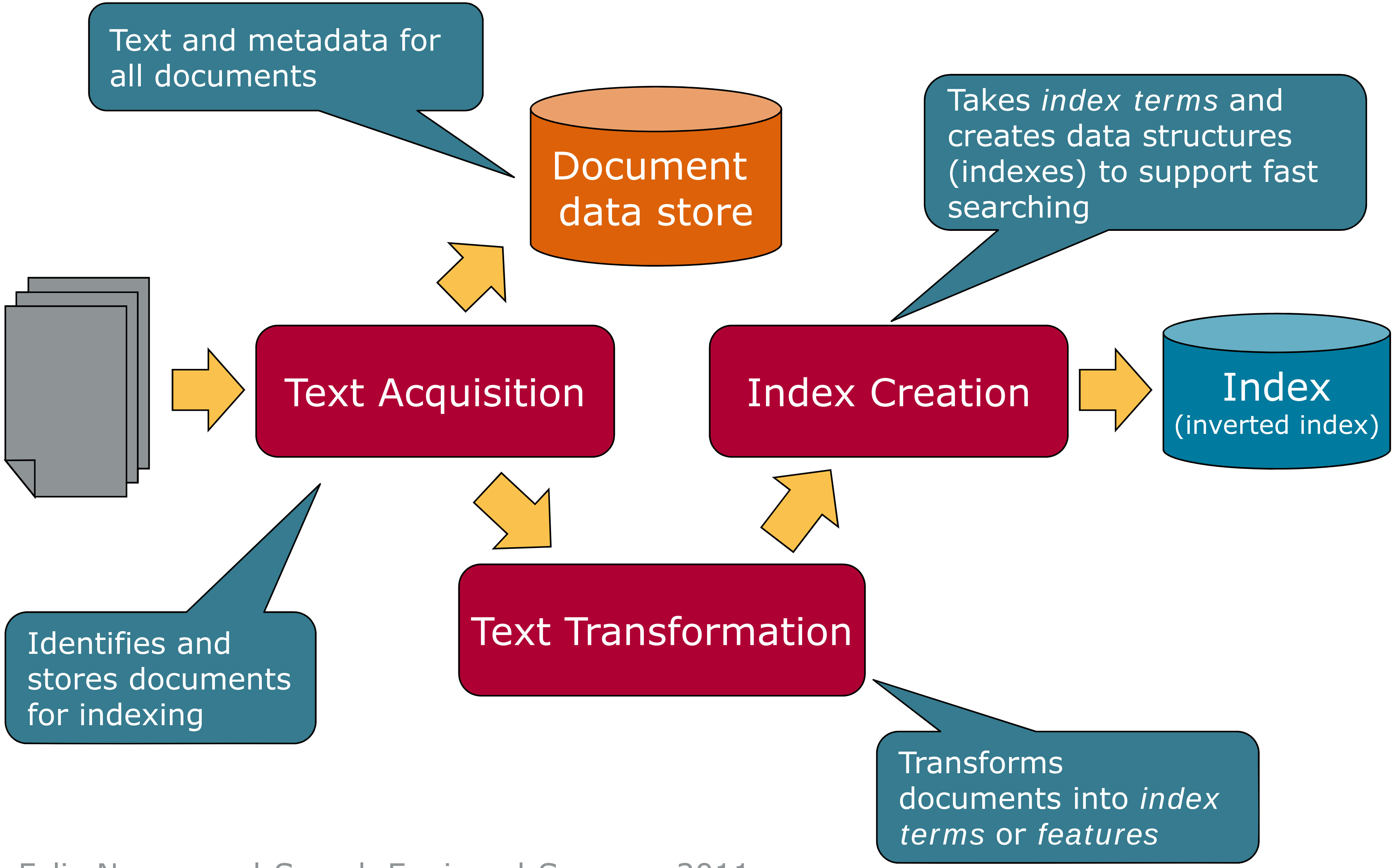
Overview

- Basic Building Blocks
- Indexing
  - Text Acquisition
  - Text Transformation
  - Index Creation
- Querying
  - User Interaction
  - Ranking
  - Evaluation

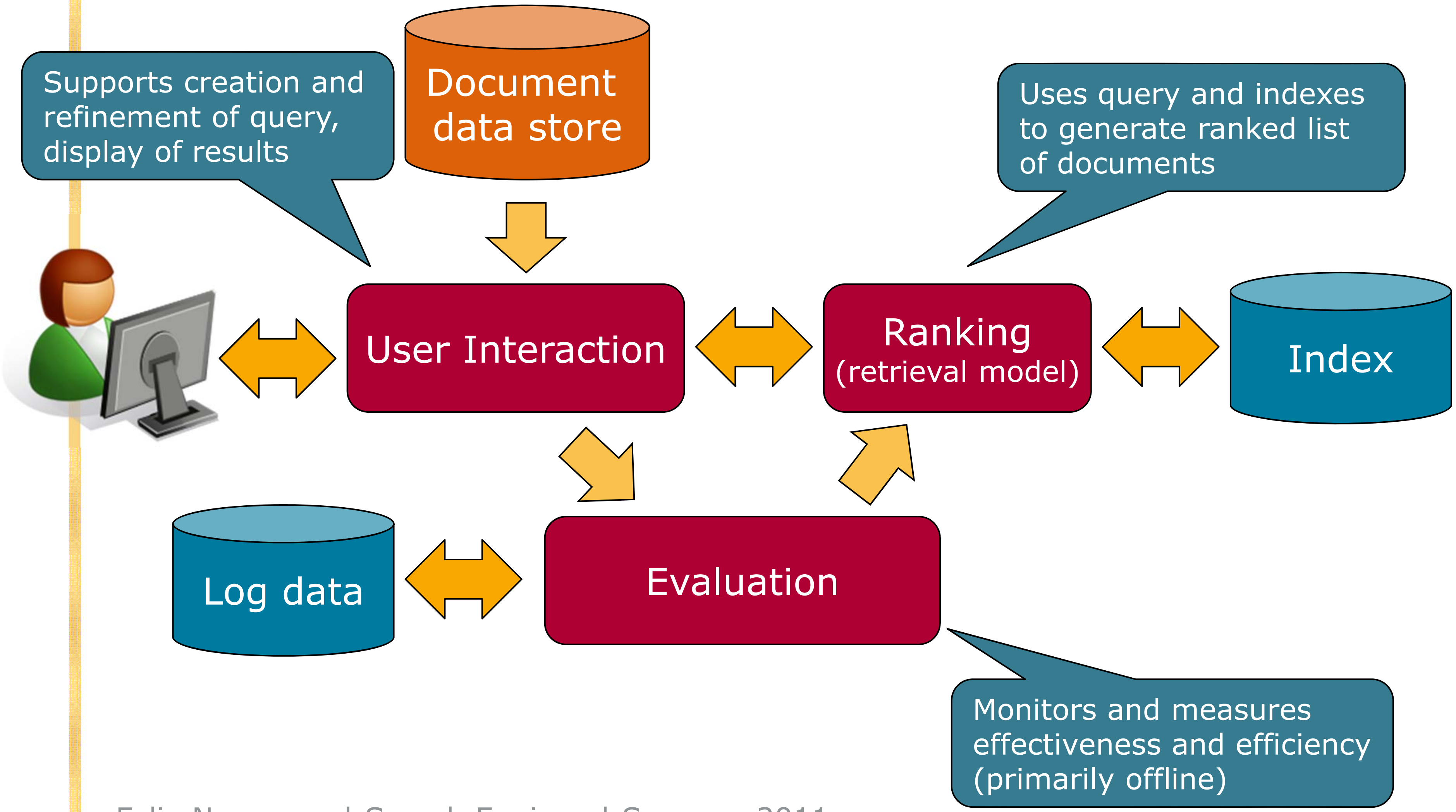
Search Engine Architecture

- Determined by two main requirements
  - **Effectiveness** (quality of results)
    - As good as possible
  - **Efficiency** (response time and throughput)
    - As quickly as possible
- Other requirements fall into these categories
  - Changing documents -> Effectiveness and efficiency
  - Personalization: Effectiveness
  - Spam: Effectiveness and efficiency
  - …

The Indexing Process


Text and metadata for all documents
Document data store
Takes *index terms* and creates data structures (indexes) to support fast searching
Text Acquisition
Index Creation
Index
(inverted index)
Text Transformation
Identifies and stores documents for indexing
Transforms documents into *index terms* or *features*

The Query Process



Supports creation and refinement of query, display of results

Document data store

Uses query and indexes to generate ranked list of documents

User Interaction

Ranking (retrieval model)

Index

Log data

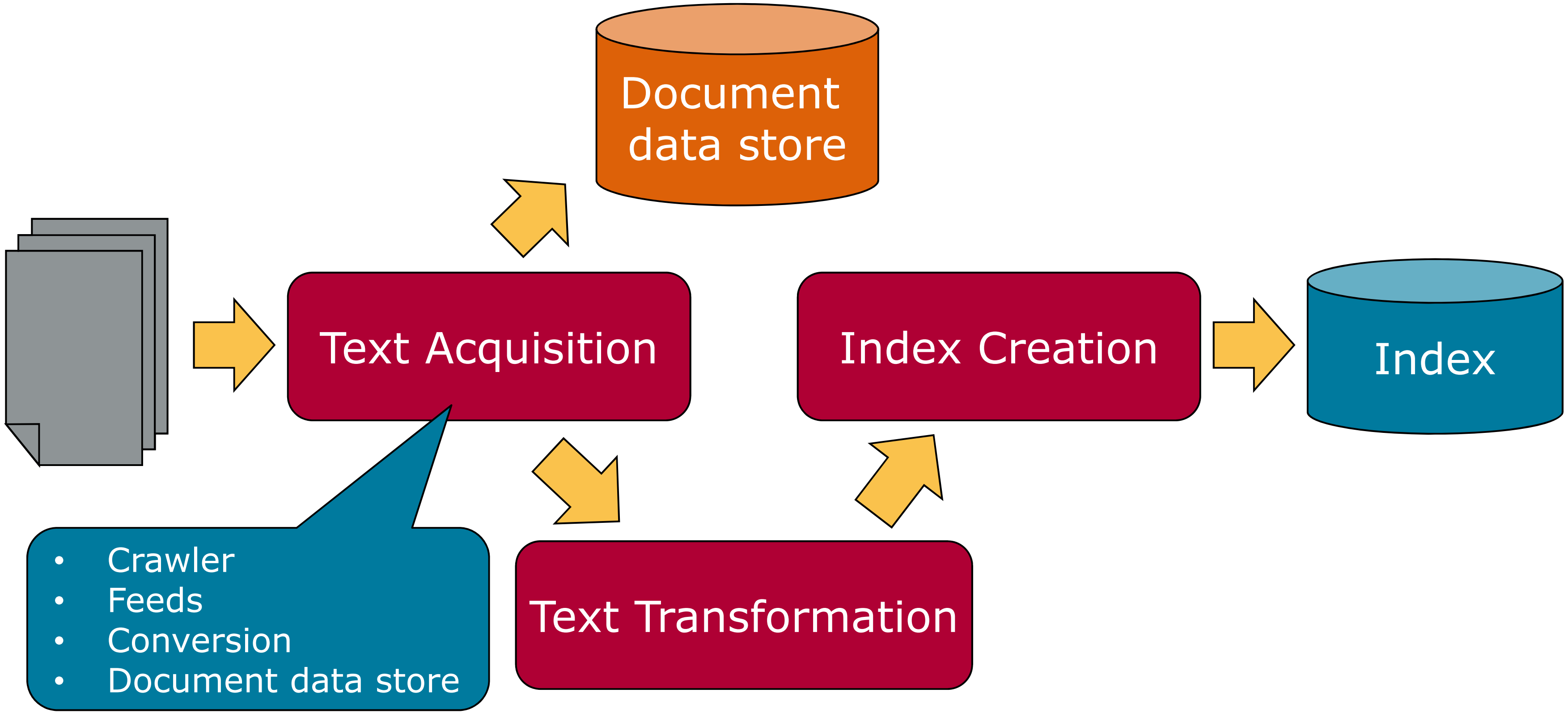
Evaluation

Monitors and measures effectiveness and efficiency (primarily offline)

Overview

- Basic Building Blocks
- Indexing
  - Text Acquisition
  - Text Transformation
  - Index Creation
- Querying
  - User Interaction
  - Ranking
  - Evaluation

The Indexing Process



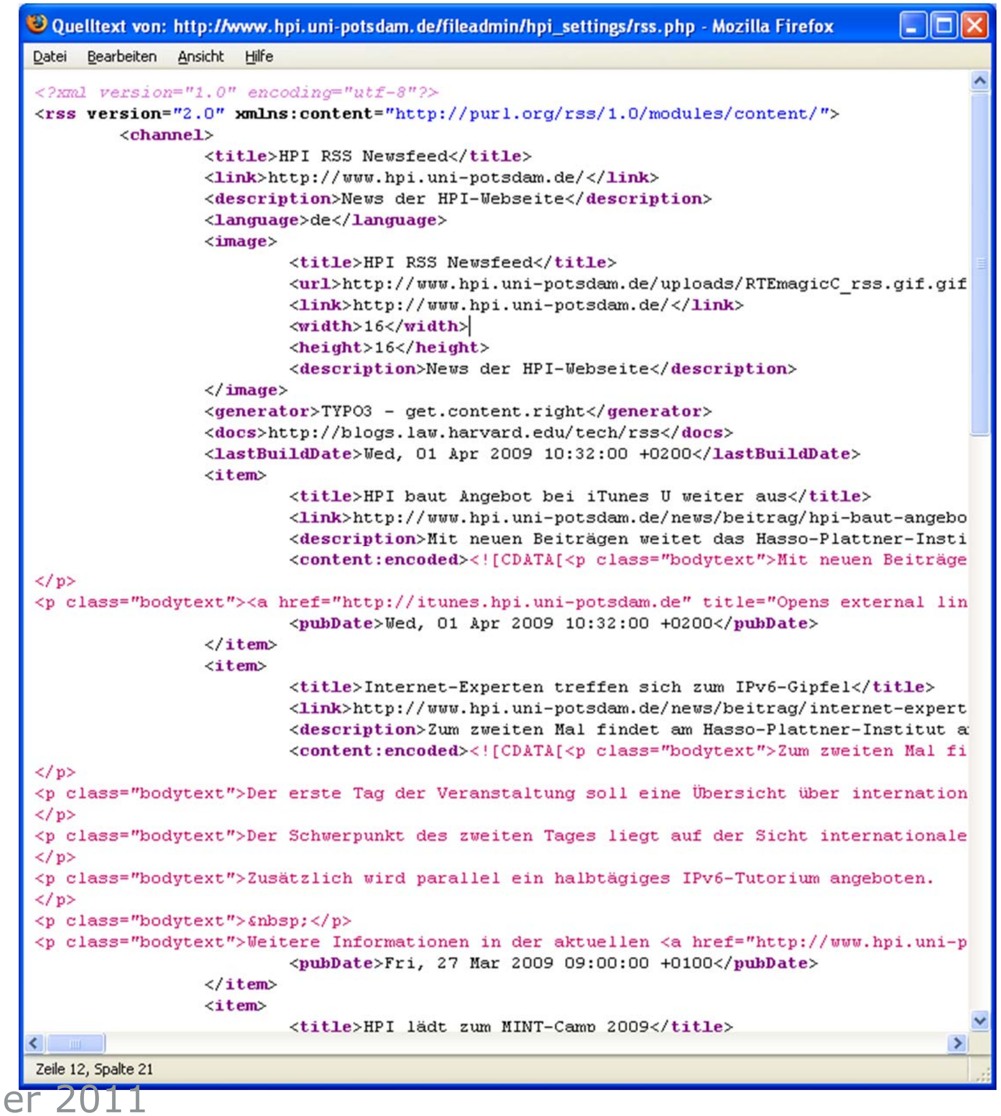
- Crawler
- Feeds
- Conversion
- Document data store

Text Acquisition – Crawler

- Identifies and acquires documents for search engine
- Many types
  - Web, enterprise, desktop
- 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

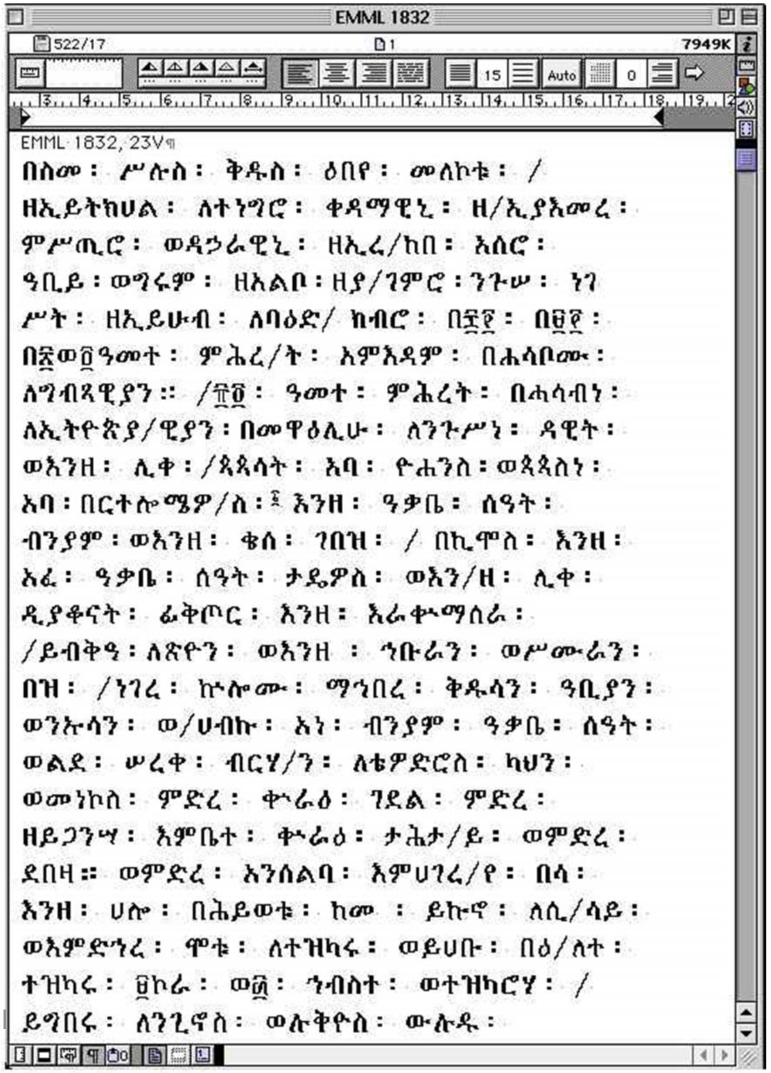
- Real-time streams of documents
  - □ Web feeds for news, blogs, video, radio, TV
- RSS is common standard
  - □ Rich Site Summary (RSS-Versions 0.9x)
  - □ RDF Site Summary (RSS-Versions 0.9 und 1.0)
  - □ Really Simple Syndication (RSS 2.0)
  - □ RSS “reader” can provide new XML documents to search engine


Quelltext von: http://www.hpi.uni-potsdam.de/fileadmin/hpi_settings/rss.php - Mozilla Firefox
Datei
Bearbeiten
Ansicht
Hilfe
Zeile 12, Spalte 21

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0" xmlns:content="http://purl.org/rss/1.0/modules/content/">
        <channel>
                <title>HPI RSS Newsfeed</title>
                <link>http://www.hpi.uni-potsdam.de/</link>
                <description>News der HPI-Webseite</description>
                <language>de</language>
                <image>
                        <title>HPI RSS Newsfeed</title>
                        <url>http://www.hpi.uni-potsdam.de/uploads/RTEmagicC_rss.gif.gif
                        <link>http://www.hpi.uni-potsdam.de/</link>
                        <width>16</width>
                        <height>16</height>
                        <description>News der HPI-Webseite</description>
                </image>
                <generator>TYPO3 - get.content.right</generator>
                <docs>http://blogs.law.harvard.edu/tech/rss</docs>
                <lastBuildDate>Wed, 01 Apr 2009 10:32:00 +0200</lastBuildDate>
                <item>
                        <title>HPI baut Angebot bei iTunes U weiter aus</title>
                        <link>http://www.hpi.uni-potsdam.de/news/beitrag/hpi-baut-angebo
                        <description>Mit neuen Beiträgen weitet das Hasso-Plattner-Insti
                        <content:encoded><![CDATA[<p class="bodytext">Mit neuen Beiträge
</p>
<p class="bodytext"><a href="http://itunes.hpi.uni-potsdam.de" title="Opens external lin
                        <pubDate>Wed, 01 Apr 2009 10:32:00 +0200</pubDate>
                </item>
                <item>
                        <title>Internet-Experten treffen sich zum IPv6-Gipfel</title>
                        <link>http://www.hpi.uni-potsdam.de/news/beitrag/internet-expert
                        <description>Zum zweiten Mal findet am Hasso-Plattner-Institut a
                        <content:encoded><![CDATA[<p class="bodytext">Zum zweiten Mal fi
</p>
<p class="bodytext">Der erste Tag der Veranstaltung soll eine Übersicht über internation
</p>
<p class="bodytext">Der Schwerpunkt des zweiten Tages liegt auf der Sicht internationale
</p>
<p class="bodytext">Zusätzlich wird parallel ein halbtägiges IPv6-Tutorium angeboten.
</p>
<p class="bodytext">&nbsp;</p>
<p class="bodytext">Weitere Informationen in der aktuellen <a href="http://www.hpi.uni-p
                        <pubDate>Fri, 27 Mar 2009 09:00:00 +0100</pubDate>
                </item>
                <item>
                        <title>HPI lädt zum MINT-Camp 2009</title>
```

Text Acquisition – Conversion

- Convert variety of document formats into a consistent text-plus-metadata format
    - e.g., HTML, XML, Word, PDF, etc. → XML
- Convert text encoding for different languages
    - Using a standard like UTF-8
    - Be consistent throughout application
- Non-content data (tags, metadata) is either removed or stored as metadata.
- First step towards text transformation



EMML 1832, 23V¶
በስመ : ሥሉስ : ቅዱስ : ዕበየ : መለኮቱ : /
ዘኢይትክሀል : ለተነግሮ : ቀዳማዊኒ : ዘ/ኢያእመረ :
ምሥጢሮ : ወዳኃራዊኒ : ዘኢረ/ከበ : አሰሮ :
ዓቢይ : ወግሩም : ዘአልቦ : ዘያ/ገምሮ : ንጉሠ : ነገ
ሥት : ዘኢይሁብ : ለባዕድ/ ክብሮ : በ፳፪ : በ፱፪ :
በ፳ወ፬ዓመተ : ምሕረ/ት : አምእዳም : በሐሳቦሙ :
ለግብጻዊያን :: /፹፬ : ዓመተ : ምሕረት : በሐሳብነ :
ለኢትዮጵያ/ዊያን : በመዋዕሊሁ : ለንጉሥነ : ዳዊት :
ወእንዘ : ሊቀ : /ጳጳሳት : አባ : ዮሐንስ : ወጳጳስነ :
አባ : በርተሎሜዎ/ስ :፮ እንዘ : ዓቃቤ : ሰዓት :
ብንያም : ወእንዘ : ቄሰ : ገበዝ : / በኪሞስ : እንዘ :
አፈ : ዓቃቤ : ሰዓት : ታዴዎስ : ወእን/ዘ : ሊቀ :
ዲያቆናት : ፊቅጦር : እንዘ : እራቀ ማሰራ :
/ይብቅዓ : ለጽዮን : ወእንዘ : ኅቡራን : ወሥሙራን :
በዝ : /ነገረ : ኵሎሙ : ማኅበረ : ቅዱሳን : ዓቢያን :
ወንኡሳን : ወ/ሀብኩ : አነ : ብንያም : ዓቃቤ : ሰዓት :
ወልደ : ሡረቀ : ብርሃ/ን : ለቴዎድሮስ : ካህን :
ወመነኮስ : ምድረ : ቍራዕ : ገደል : ምድረ :
ዘይጋንሣ : እምቤተ : ቍራዕ : ታሕታ/ይ : ወምድረ :
ደበዛ :: ወምድረ : አንሰልባ : እምሀገሩ/የ : በሳ :
እንዘ : ሀሎ : በሕይወቱ : ከመ : ይኩኖ : ለሲ/ሳይ :
ወእምድኅረ : ሞቱ : ለተዝካሩ : ወይሀቡ : በዕ/ለተ :
ተዝካሩ : ፱ኮራ : ወ፴ : ኅብስተ : ወተዝካሮሃ : /
ይግበሩ : ለንጊኖስ : ወሉቅዮስ : ውሉዱ :

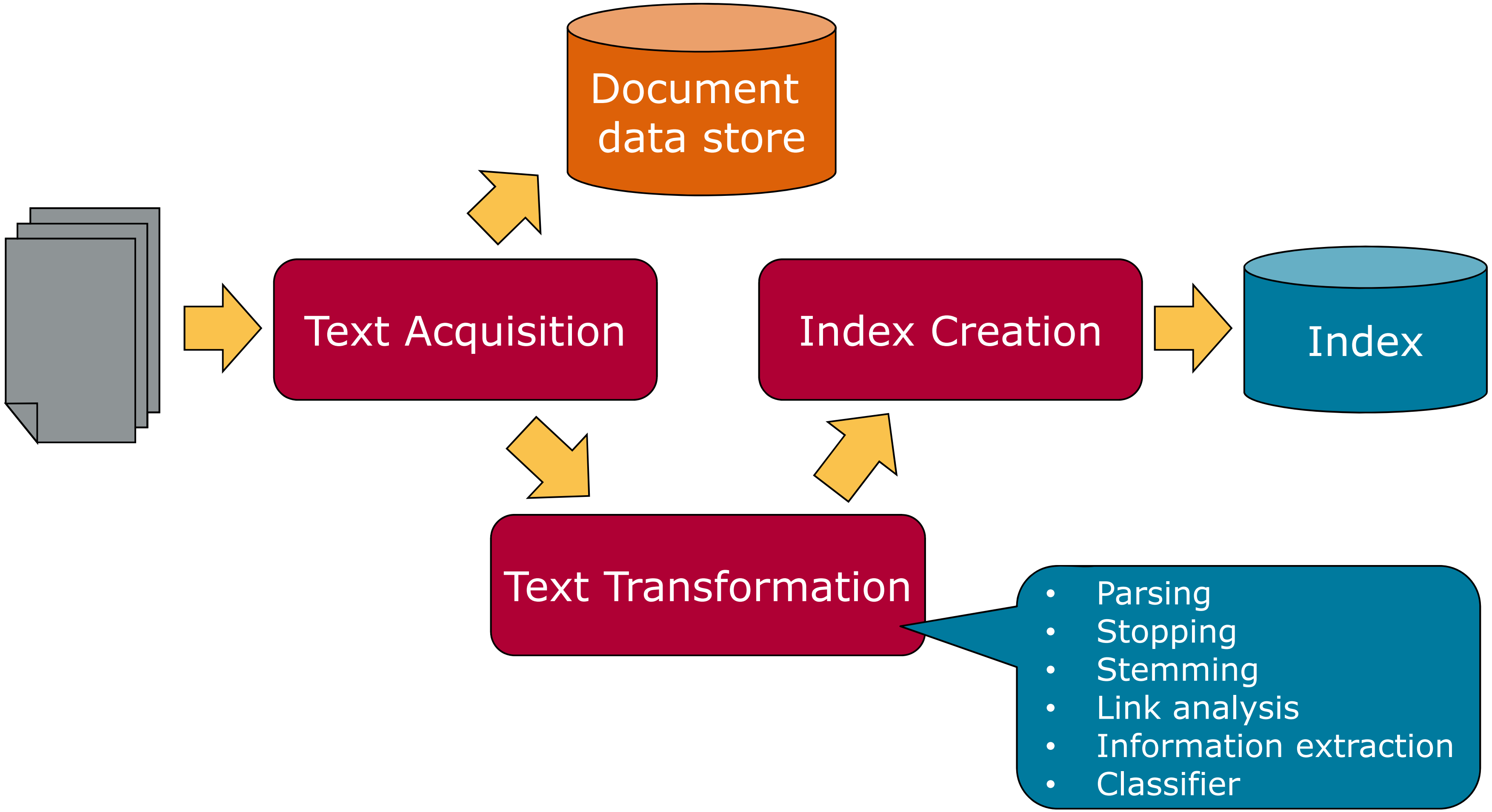
http://www.uni-mainz.de/Organisationen/TLA/dokumentation/sgml_eng.html

Text Acquisition – Document data store

- Two parts
  - Unstructured text (compressed)
  - Structured metadata
- Stores text, metadata, and other related content for documents
  - Metadata is information about document
    - Type, creation date, …
  - Other content includes links, anchor text
- Why store documents? They are available on the Web anyway…
  - Provide fast access to document contents for search engine components
    - Result list generation, document summary, snippets
- Could use relational database system
  - More typically, a simpler, more efficient storage system is used due to huge numbers of documents
    - Key-value-stores

More in Chapter 3

The Indexing Process

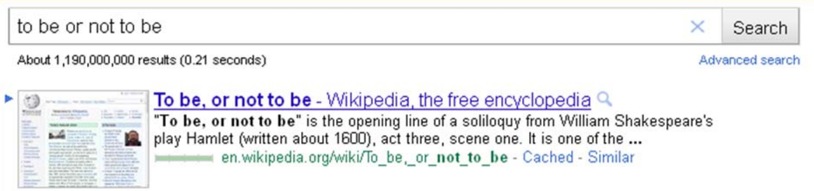


- Parsing
- Stopping
- Stemming
- Link analysis
- Information extraction
- Classifier

Text Transformation – Parsing

- Processing the sequence of text *tokens* in the document to recognize structural elements
  - Titles, links, headings, etc.
- *Tokenizer* recognizes “words” in the text.
  - Must consider issues like capitalization, hyphens, apostrophes, non-alpha characters, separators
  - Many decisions up front:
    - apple vs. Apple
    - O’Conner vs. owner’s
    - Word separation in Chinese
- *Markup languages* such as HTML, XML often used to specify structure
  - *Tags* used to specify document *elements*
    - E.g., <h2> Overview </h2>
  - Document parser uses *syntax* of markup language (or other formatting) to identify structure
    - E.g. email format, MS Word metadata etc.

Text Transformation – Stopping



About 1,190,000,000 results (0.21 seconds)

To be, or not to be - Wikipedia, the free encyclopedia
"**To be, or not to be**" is the opening line of a soliloquy from William Shakespeare's play Hamlet (written about 1600), act three, scene one. It is one of the ...
en.wikipedia.org/wiki/To_be,_or_**not_to_be** - Cached - Similar

- Remove common words
  - □ “and”, “or”, “the”, “in”, …
- Some impact on efficiency and effectiveness
- Can be a problem for some queries
  - □ *To be or not to be*

I a about an are as at be by com de en for from how in is it la of on or that the this to was what when where who will with und the www

See also: http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words

aber als am an auch auf aus bei bin bis bist da dadurch daher darum das daß dass dein deine dem den der des dessen deshalb die dies dieser dieses doch dort du durch ein eine einem einen einer eines er es euer eure für hatte hatten hattest hattet hier hinter ich ihr ihre im in ist ja jede jedem jeden jeder jedes jener jenes jetzt kann kannst können könnt machen mein meine mit muß mußt musst müssen müßt nach nachdem nein nicht nun oder seid sein seine sich sie sind soll sollen sollst sollt sonst soweit sowie und unser unsere unter vom von vor wann warum was weiter weitere wenn wer werde werden werdet weshalb wie wieder wieso wir wird wirst wo woher wohin zu zum zur über

http://www.ranks.nl/stopwords/german.html

Text Transformation – Stemming

- Group words derived from a common *stem*
  - “computer”, “computers”, “computing”, “compute”
  - Fish, fishing, fisherman
- Usually effective, but not for all queries
  - Aggressive vs. conservative vs. not at all
- Benefits vary for different languages
  - Arabic: Very complicated morphology
  - Chinese: Few word variations anyway

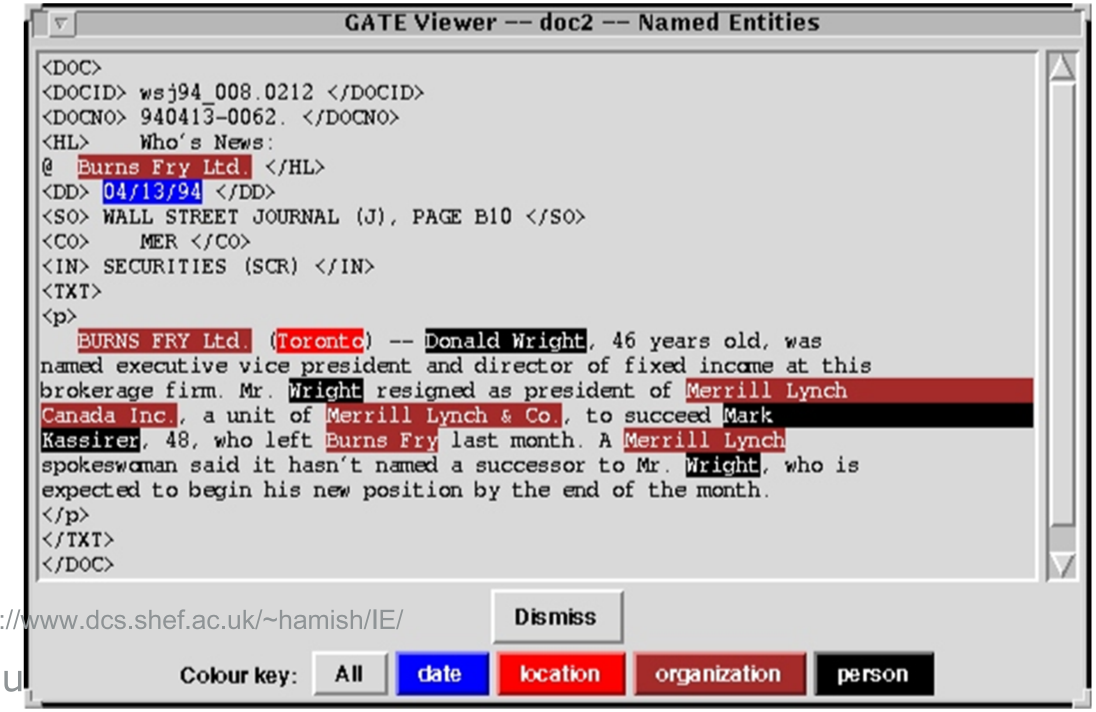
Text Transformation – Link Analysis

- Makes use of *links* and *anchor text* in web pages.
  - □ Stored and indexed separately
  - □
```
<a href = http://www.hpi.uni-potsdam.de/naumann/home.html>
  Information Systems Group
</a>
```
- 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

http://www.guardian.co.uk/media/2008/jul/14/mediatop100200896

Text Transformation – Information Extraction

- Identify classes of index terms that are important for some applications
- Simple: Bold-face, heading, title
- Part of speech tagging
- *Named entity recognizers (NER)* identify classes such as
  - *People*
  - *Locations*
  - *Companies*
  - *Dates*
  - etc.



```
<DOC>
<DOCID> wsj94_008.0212 </DOCID>
<DOCNO> 940413-0062. </DOCNO>
<HL>    Who's News:
@  Burns Fry Ltd. </HL>
<DD> 04/13/94 </DD>
<SO> WALL STREET JOURNAL (J), PAGE B10 </SO>
<CO>    MER </CO>
<IN> SECURITIES (SCR) </IN>
<TXT>
<p>
   BURNS FRY Ltd. (Toronto) -- Donald Wright, 46 years old, was
named executive vice president and director of fixed income at this
brokerage firm. Mr. Wright resigned as president of Merrill Lynch
Canada Inc., a unit of Merrill Lynch & Co., to succeed Mark
Kassirer, 48, who left Burns Fry last month. A Merrill Lynch
spokeswoman said it hasn't named a successor to Mr. Wright, who is
expected to begin his new position by the end of the month.
</p>
</TXT>
</DOC>
```

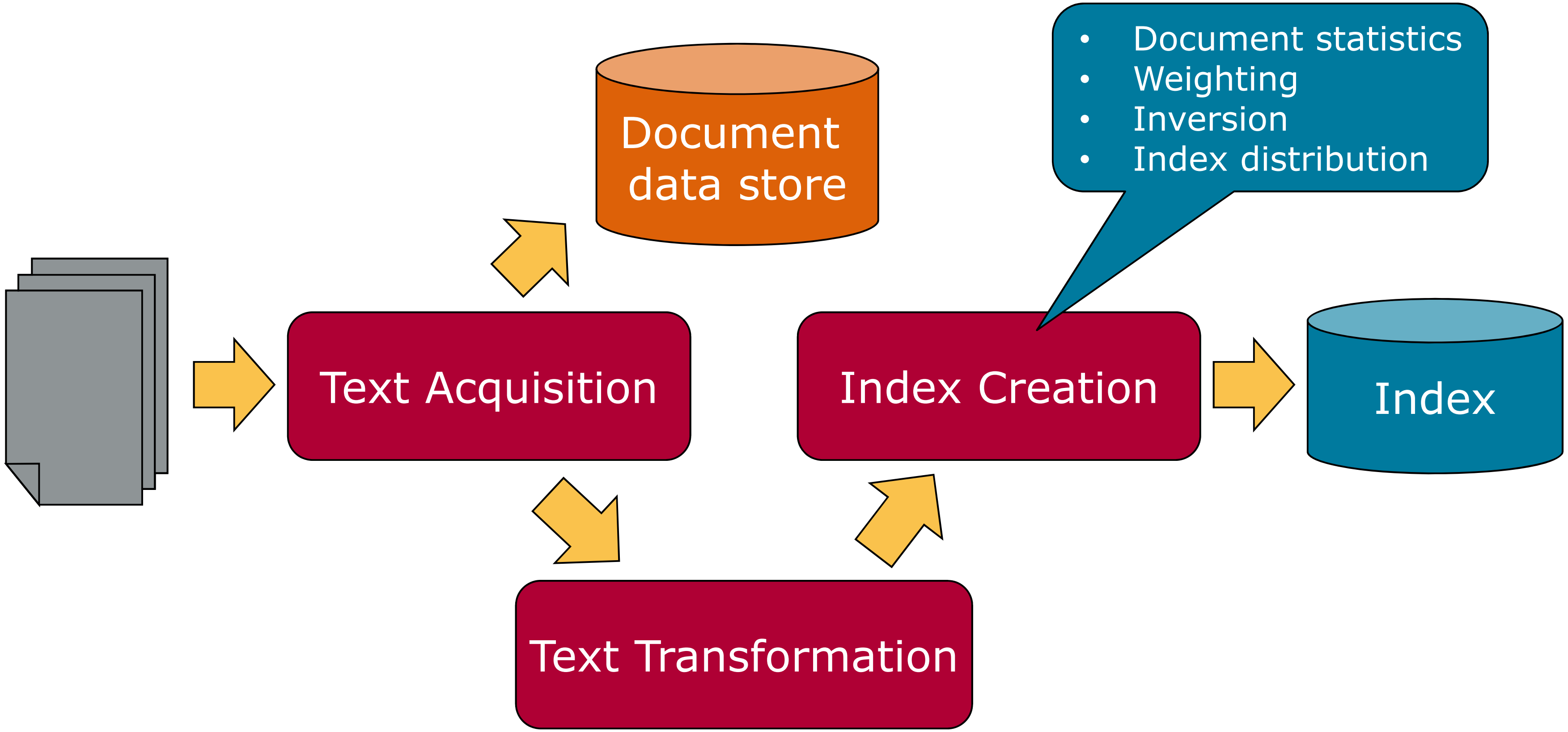
http://www.dcs.shef.ac.uk/~hamish/IE/

Text Transformation – Classifier

- Identifies class-related metadata for documents
  - i.e., assigns labels to documents
  - e.g., topics, reading levels, sentiment, genre
  - Spam!
  - Advertisements in documents
- Use depends on application

More in Chapter 4

The Indexing Process



Index Creation - Document Statistics

- Statistical information about words, features and documents
- Gathers counts and positions of words and other features
  - Within a document
  - Across groups of documents
  - Across all documents
- Used in ranking algorithm

Index Creation – Weighting

- Computes weights for index terms
  - Relative importance of words in documents
- Used in ranking algorithm
  - Global weight
  - Query-dependent weight
- e.g., *tf.idf* weight
  - Combination of *term frequency* in document
  - and *inverse document frequency* in the collection

Index Creation – Inversion

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

Index Creation – Index Distribution

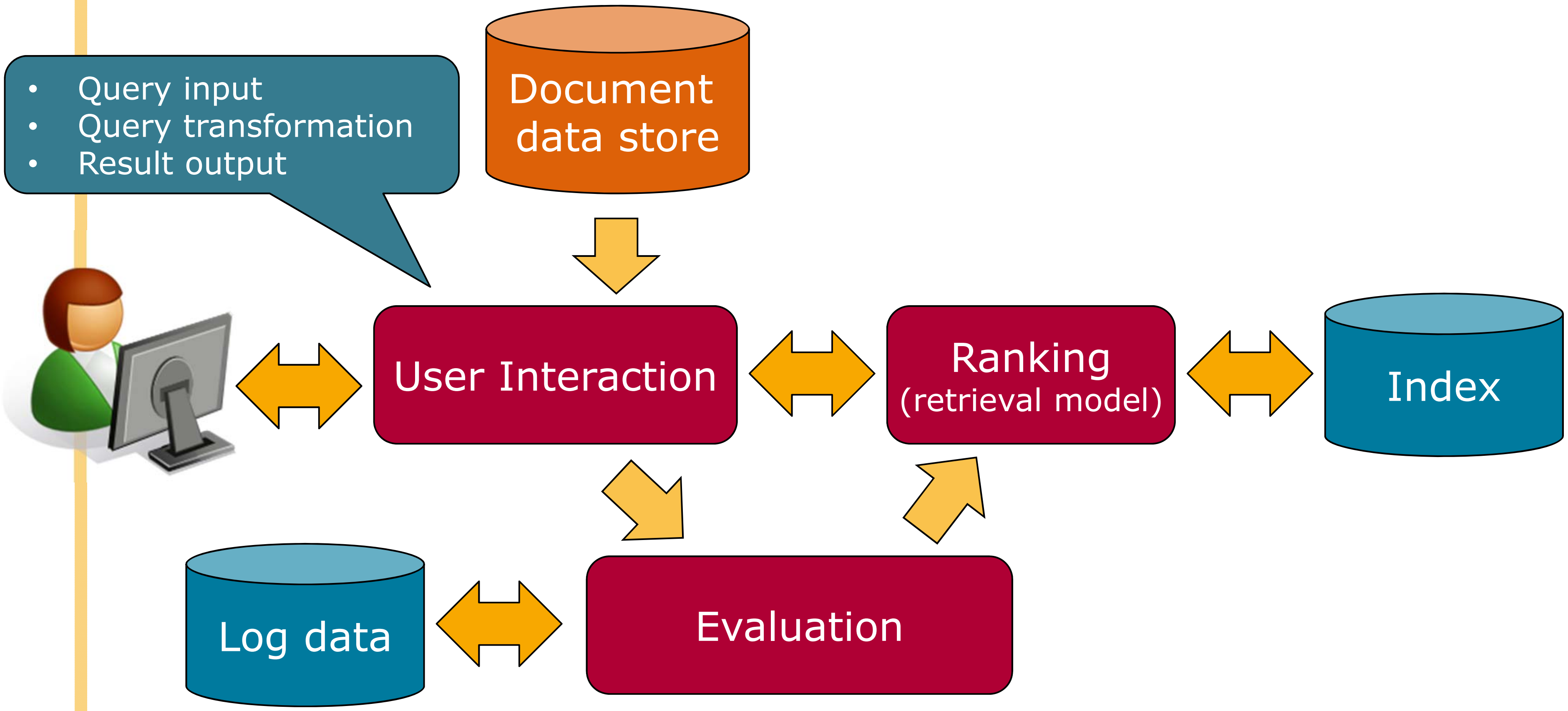
- Distribute indexes
  - across multiple computers
  - and/or multiple sites
- Essential for fast query processing with large numbers of documents
- Many variations
  - Document distribution: Distribute index for subsets of documents
  - Term distribution: Distribute index for subset of terms
  - Replication
- *P2P* and *Distributed IR* involve search across multiple sites

More in Chapter 5

Overview

- Basic Building Blocks
- Indexing
  - Text Acquisition
  - Text Transformation
  - Index Creation
- Querying
  - User Interaction
  - Ranking
  - Evaluation

The Query Process

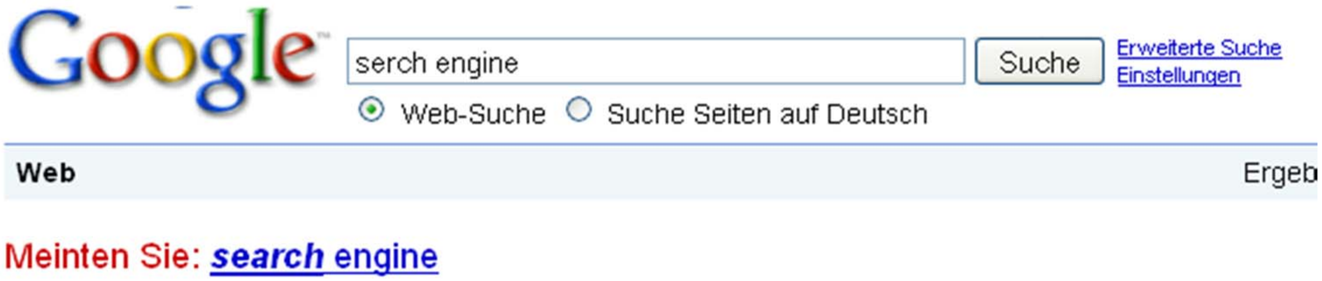


User Interaction - Query input

- Provides interface and parser for internal *query language*
- Most web queries are very simple
  - Other applications may use forms
- Query language used to describe more complex queries and results of query transformation
  - +, -, " ", ~, site:, AND, OR, …
  - Similar to SQL language used in database applications
    - Not for "end users"
  - IR query languages also allow structure specifications, but focus on content

User Interaction – Query transformation

- Improves initial query
  - both before and after initial search
- Includes same text transformation techniques used for documents
  - Tokenization, stemming, stopping
- *Spell checking* and *query suggestion* provide alternatives to original query
  - Based on query logs
- *Query expansion* and *relevance feedback* modify the original query with additional terms

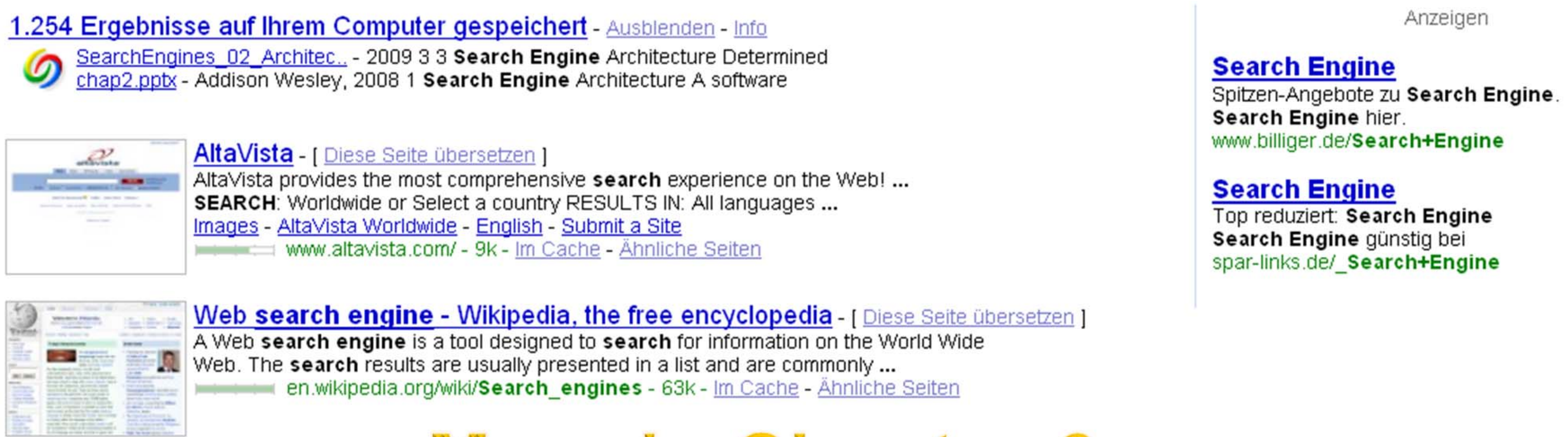


serch engine

Meinten Sie: ***search*** engine

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
- May provide *clustering* and other visualization tools
- May translate results from foreign languages



1.254 Ergebnisse auf Ihrem Computer gespeichert - Ausblenden - Info

SearchEngines_02_Architec.. - 2009 3 3 **Search Engine** Architecture Determined
chap2.pptx - Addison Wesley, 2008 1 **Search Engine** Architecture A software

AltaVista - [Diese Seite übersetzen]
AltaVista provides the most comprehensive **search** experience on the Web! ...
SEARCH: Worldwide or Select a country RESULTS IN: All languages ...
Images - AltaVista Worldwide - English - Submit a Site
www.altavista.com/ - 9k - Im Cache - Ähnliche Seiten

Web **search engine** - Wikipedia, the free encyclopedia - [Diese Seite übersetzen]
A Web **search engine** is a tool designed to **search** for information on the World Wide Web. The **search** results are usually presented in a list and are commonly ...
en.wikipedia.org/wiki/**Search_engines** - 63k - Im Cache - Ähnliche Seiten

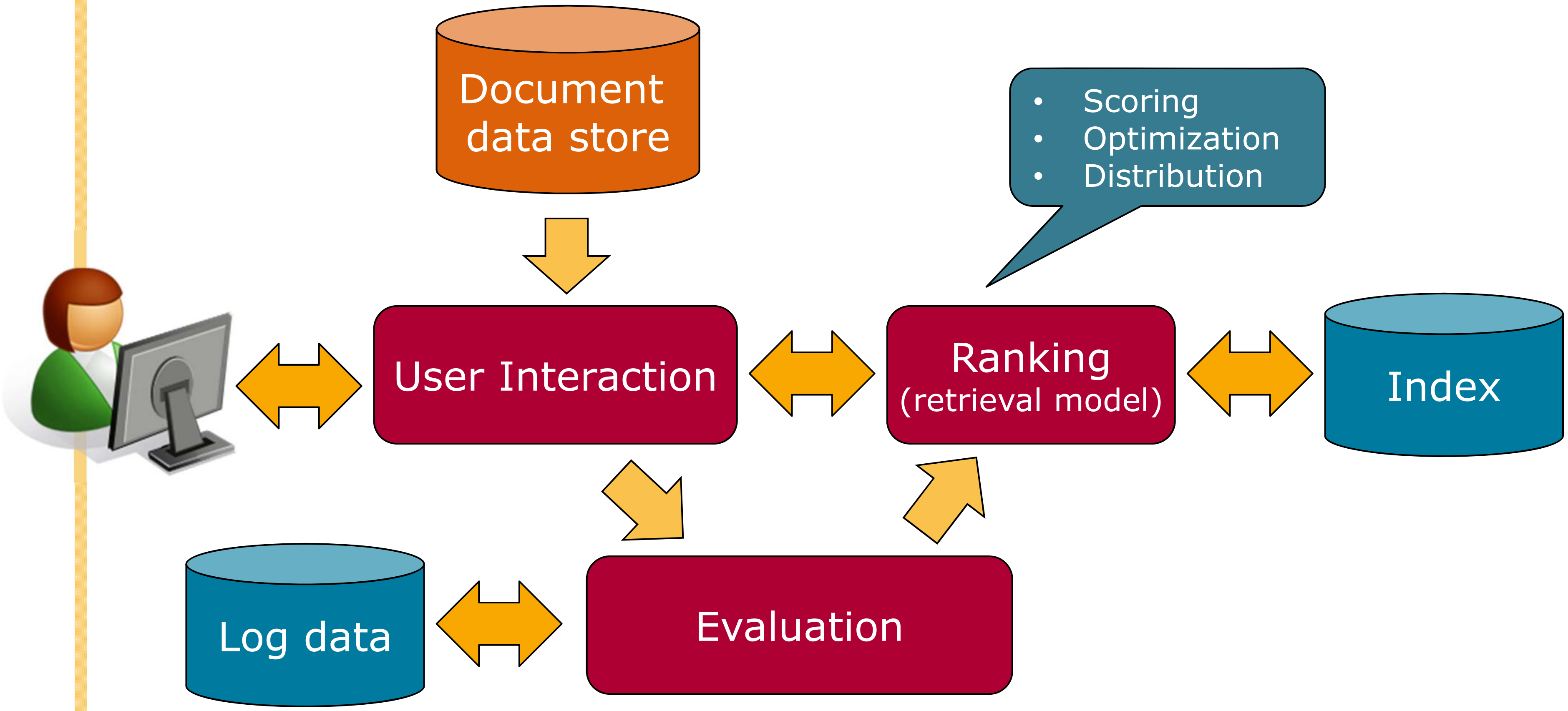
Anzeigen

Search Engine
Spitzen-Angebote zu **Search Engine**.
Search Engine hier.
www.billiger.de/**Search+Engine**

Search Engine
Top reduziert: **Search Engine**
Search Engine günstig bei
spar-links.de/_**Search+Engine**

More in Chapter 6

The Query Process



Ranking – Scoring

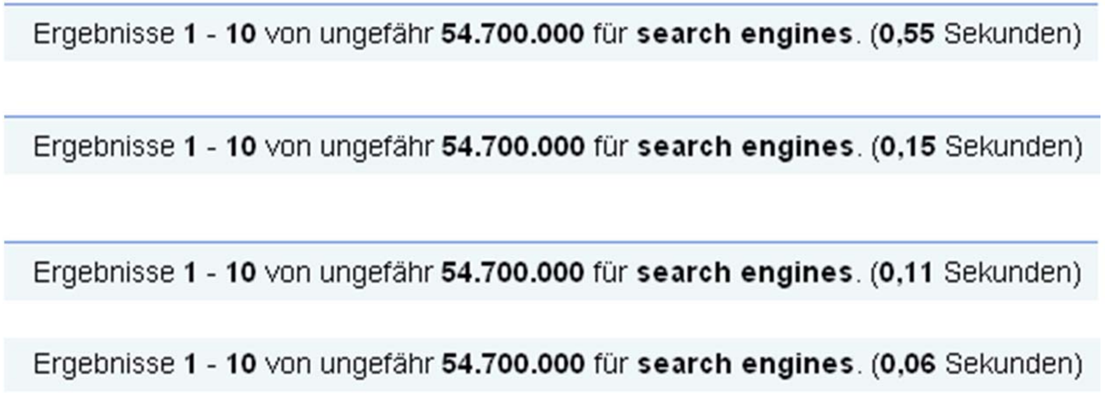
- ≈ database query processing
- Calculates scores for documents using a ranking algorithm
  - □ Based on retrieval model
- Core component of search engine
- Basic form of score is $\sum_i q_i \cdot d_i$
  - □ Summation over vocabulary of collection
  - □ q_i and d_i are query and document term weights for term i
- Many variations of ranking algorithms and retrieval models
- Key requirement: Fast execution!

Ranking – Performance optimization

- Designing ranking algorithms for efficient processing
  - *Term-at-a time* vs. *document-at-a-time* processing
  - *Safe* vs. *unsafe* optimizations
    - Trade-off between speed and quality

Ranking – Distribution

- Processing queries in a distributed environment
- *Query broker* distributes queries and assembles results
- *Caching* is a form of distributed searching



Ergebnisse **1** - **10** von ungefähr **54.700.000** für **search engines**. (**0,55** Sekunden)

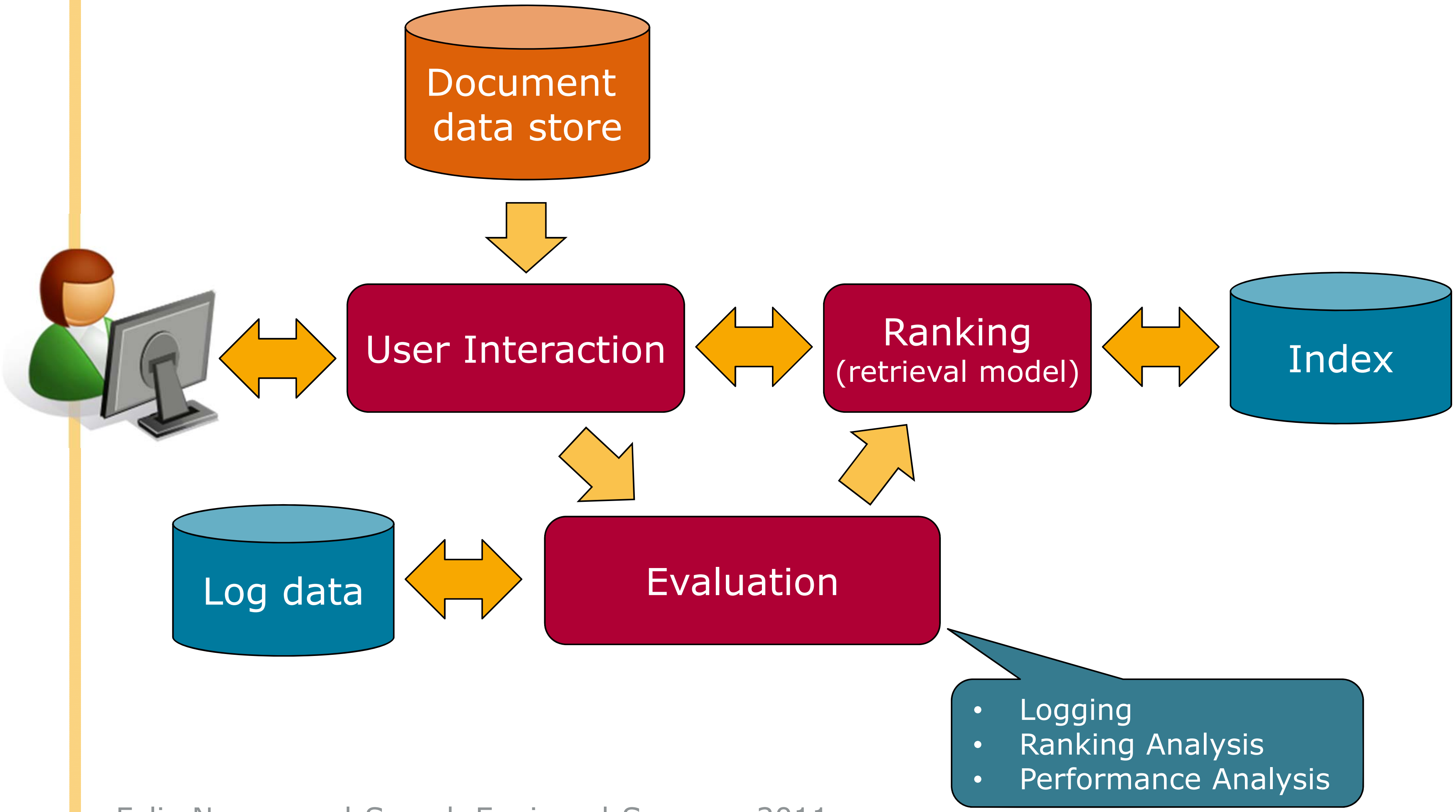
Ergebnisse **1** - **10** von ungefähr **54.700.000** für **search engines**. (**0,15** Sekunden)

Ergebnisse **1** - **10** von ungefähr **54.700.000** für **search engines**. (**0,11** Sekunden)

Ergebnisse **1** - **10** von ungefähr **54.700.000** für **search engines**. (**0,06** Sekunden)

More in Chapter 7

The Query Process



- Logging
- Ranking Analysis
- Performance Analysis

Evaluation – Logging

- Logging user queries and interaction is crucial for improving search effectiveness and efficiency.
- *Query logs* and *clickthrough data (& dwell time)* used for
  - □ Query suggestion
  - □ Spell checking
  - □ Query caching
  - □ Ranking
  - □ Advertising search
  - □ …
- Assumption: Pages clicked on are relevant to query.

Evaluation – Ranking and Performance Analysis

- Ranking analysis
  - Measuring and tuning ranking effectiveness
  - Variety of measures
- Performance analysis
  - Measuring and tuning system efficiency
  - Response time, throughput
  - Simulation

More in Chapter 8

How Does It *Really* Work?

- This course explains the components of a search engine in more detail.
- Often many possible approaches and techniques for a given component
  - Focus is on the most important alternatives
    - Explain a small number of approaches in detail rather than many approaches
  - "Importance" based on research results and use in actual search engines
  - Alternatives described in references (see book)

Summary

- Indexing
  - □ Text Acquisition
  - □ Text Transformation
  - □ Index Creation
- Querying
  - □ User Interaction
  - □ Ranking
  - □ Evaluation