Informatics 225
Computer Science 221

Information Retrieval

Lecture 3

These course materials borrow, with permission, from those of Prof. Cristina Videira Lopes, Prof. Alberto Krone-Martins, Addison Wesley 2008, Chris Manning, Pandu Nayak, Hinrich Schütze, Heike Adel, Sascha Rothe, Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie. Powerpoint theme by Prof. André van der Hoek.

Announcements

- Remember to complete the Course Policies ‘quiz’
  - 1 bonus point for doing this in due time!

Information Retrieval and Web Search

Introduction to Information Retrieval

Classic IR Assumptions

- **Corpus**: fixed document collection
- Goal: retrieve information content relevant to the information need

Classic IR Goal

- "**Relevance**"
  - For each **query Q**, and stored **document D**, **there exists** a relevance **score R(Q, D)**
  - **Maximize R(Q, D)**
    - Context is ignored
    - User is ignored
    - Corpus is static

Web IR

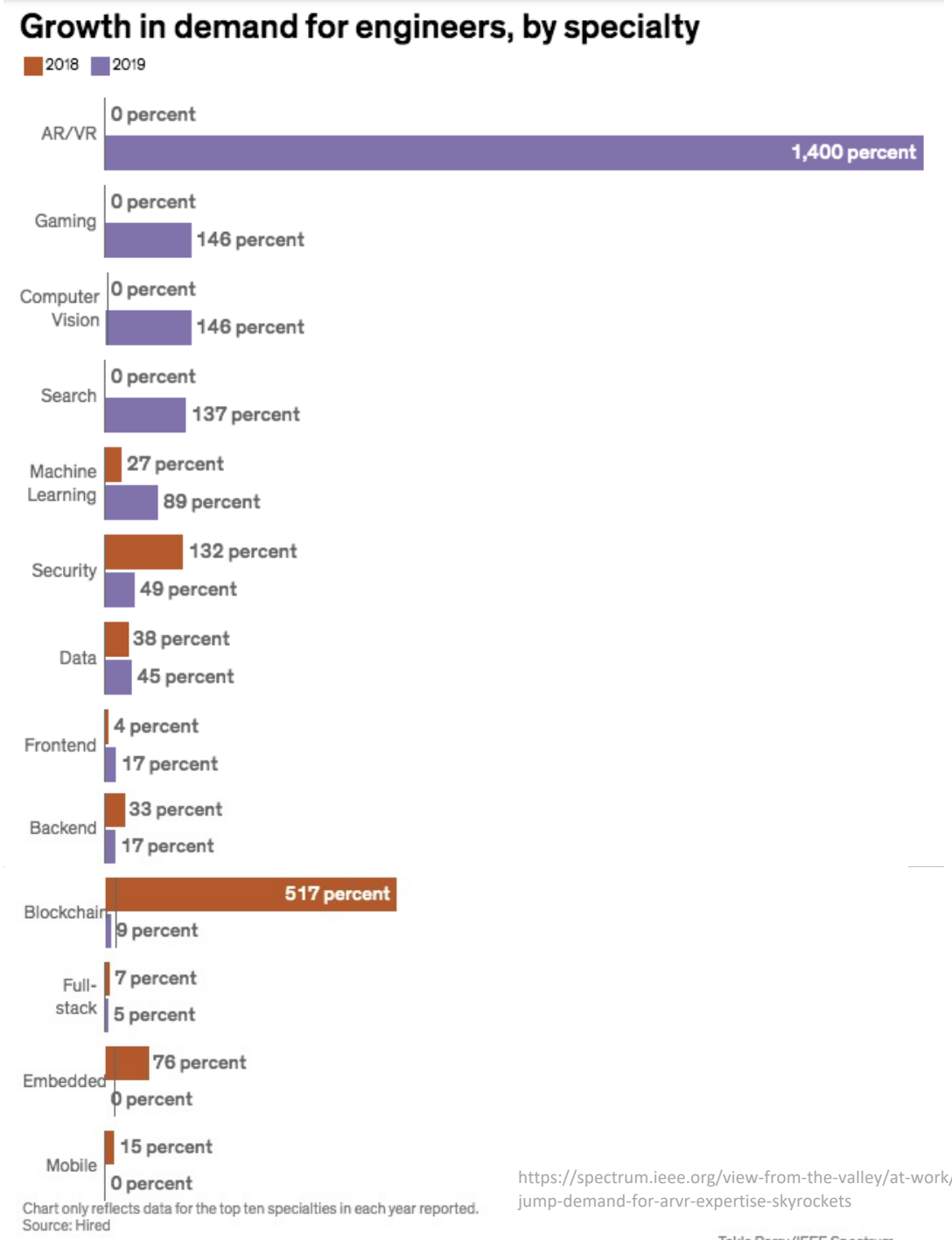
- The Web is huge (cannot store in any centralized memory!)
  - Corpus is not centralized!
- The Web changes all the time (needs to update constantly!)
  - Corpus is not static!
- There is information to avoid (adversarial IR!)
  - Context cannot be ignored!
- One interface for hugely divergent needs :
  - User cannot be ignored!

Web Search Engines

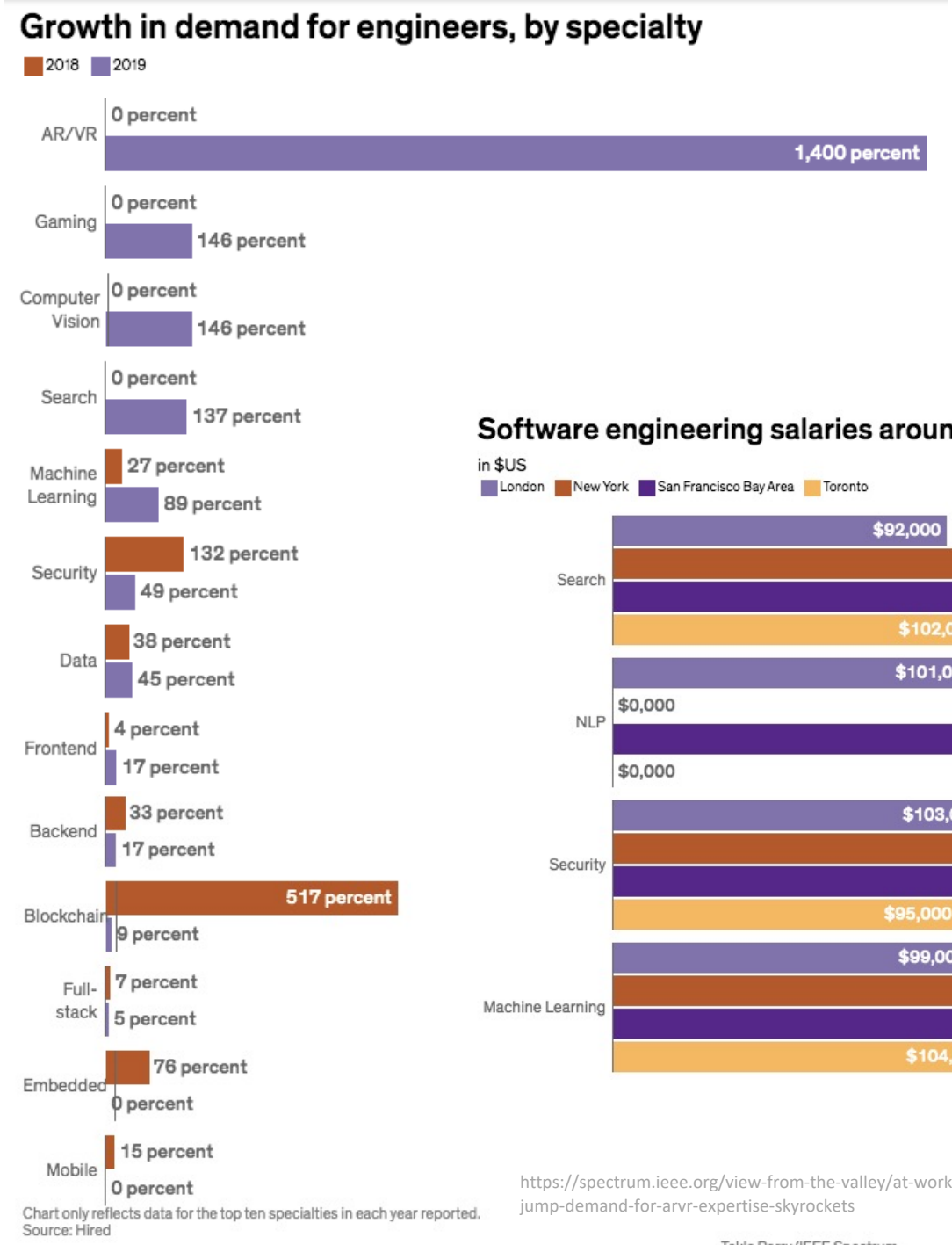
- Practical and useful applications of IR
- Must crawl tera to peta bytes of web pages and provide sub-second response to queries *(but indexing can take "a bit" longer!)*
- Big Issues in the design, additionally to IR issues:
  - Performance
    - Response time
    - Query throughput
    - Indexing speed
  - Speed in discovery and integration of new documents
  - Coverage
  - Freshness
  - Spam

Search Engineers

- Develop and maintain search engines
- Design or optimize content for search engines



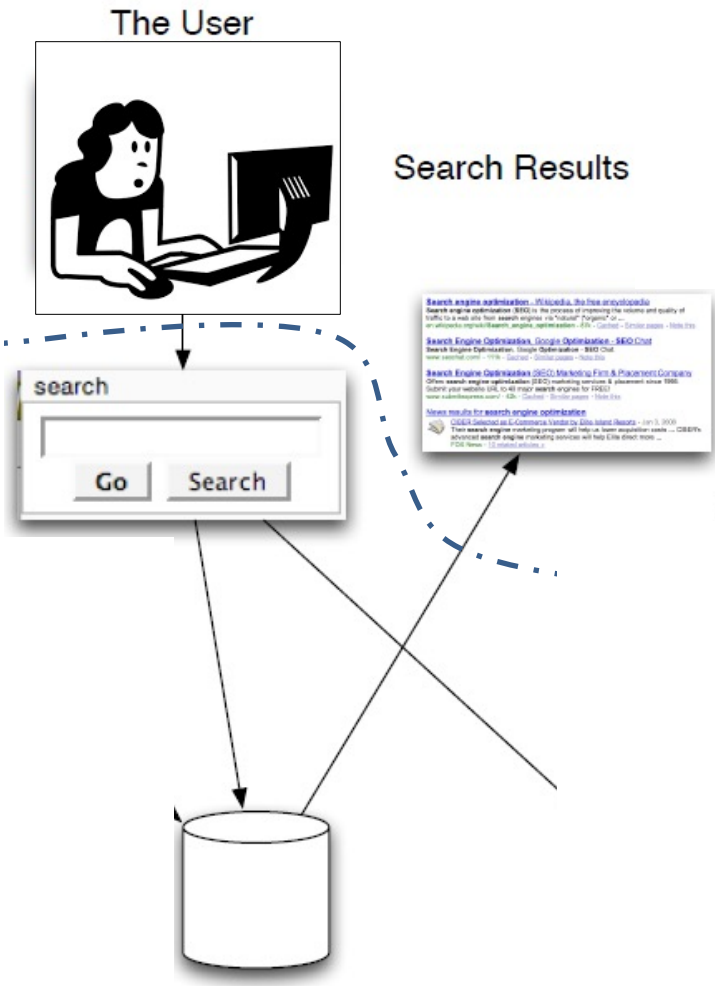
https://spectrum.ieee.org/view-from-the-valley/at-work/tech-careers/software-engineering-salaries-jump-demand-for-arvr-expertise-skyrockets



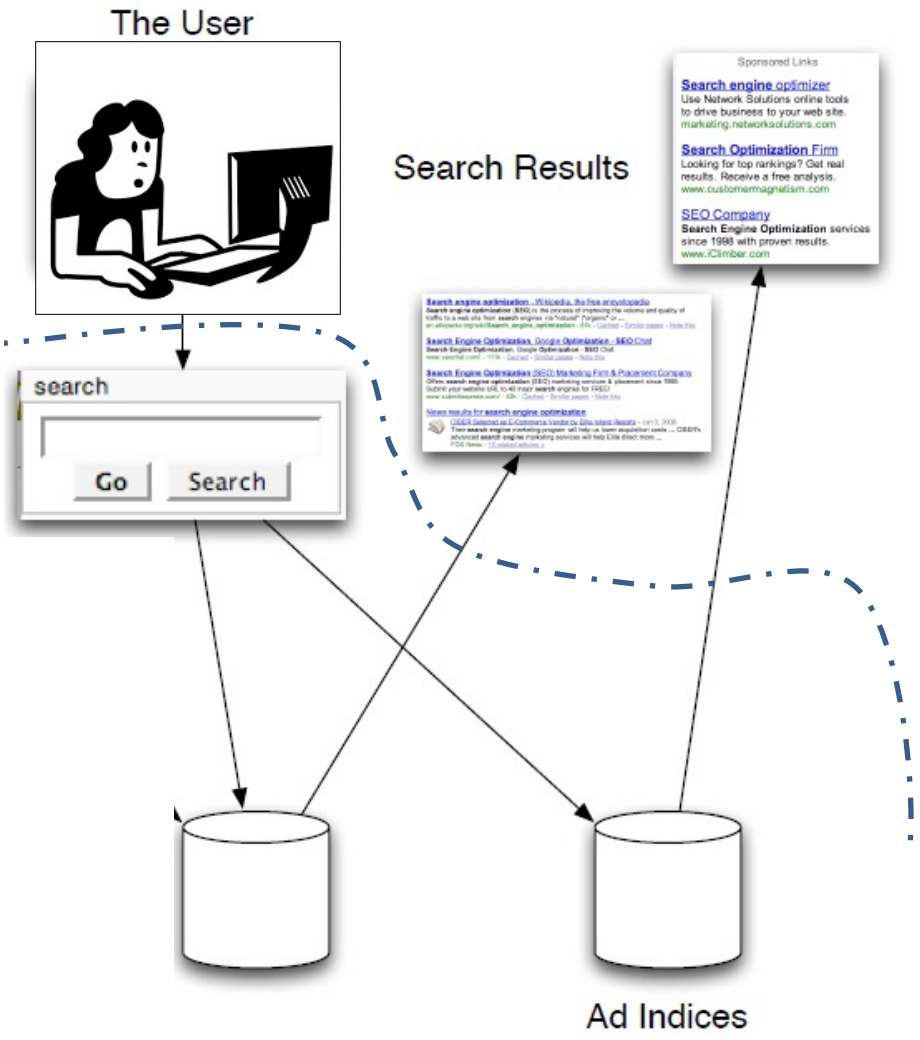
https://spectrum.ieee.org/view-from-the-valley/at-work/tech-careers/software-engineering-salaries-jump-demand-for-arvr-expertise-skyrockets

Workflow

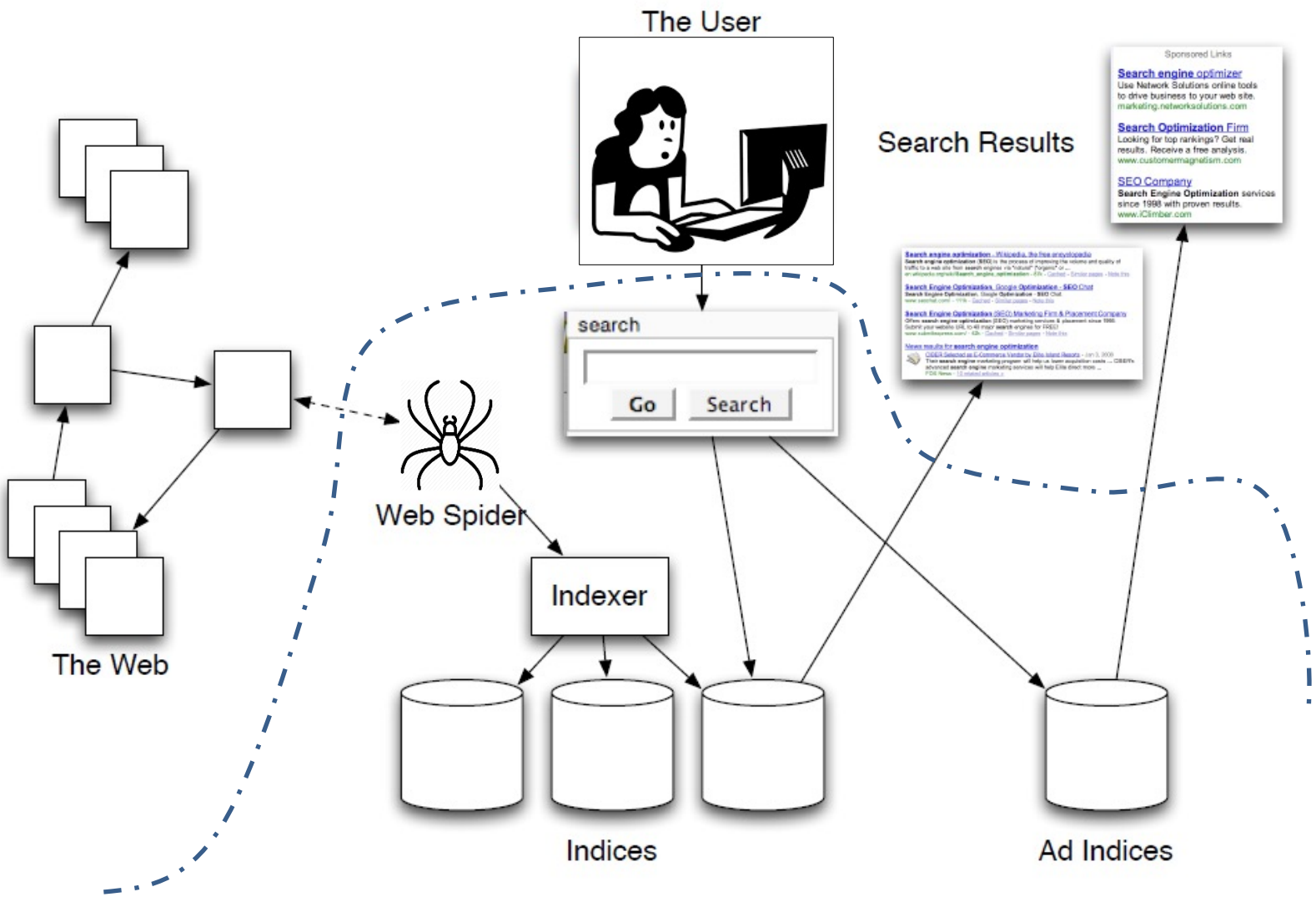
Workflow



Workflow


The User
search
Go
Search
Search Results
Sponsored Links
Search engine optimizer
Use Network Solutions online tools to drive business to your web site.
marketing.networksolutions.com
Search Optimization Firm
Looking for top rankings? Get real results. Receive a free analysis.
www.customermagnetism.com
SEO Company
Search Engine Optimization services since 1998 with proven results.
www.iClimber.com
Ad Indices

Workflow

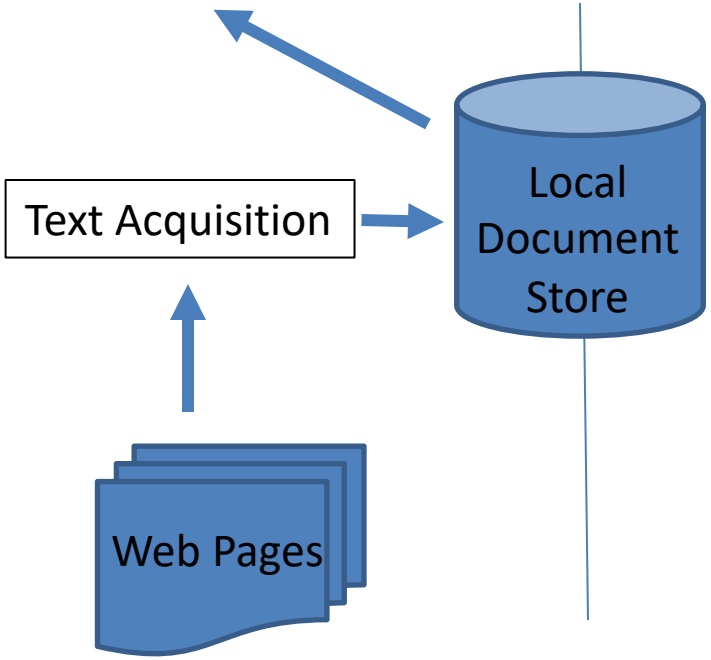

The User
Search Results
Sponsored Links
Search engine optimizer
Use Network Solutions online tools to drive business to your web site.
marketing.networksolutions.com
Search Optimization Firm
Looking for top rankings? Get real results. Receive a free analysis.
www.customermagnetism.com
SEO Company
Search Engine Optimization services since 1998 with proven results.
www.iClimber.com
search
Go
Search
Web Spider
Indexer
The Web
Indices
Ad Indices

Architecture

Preprocessing Steps

Architecture

Preprocessing Steps

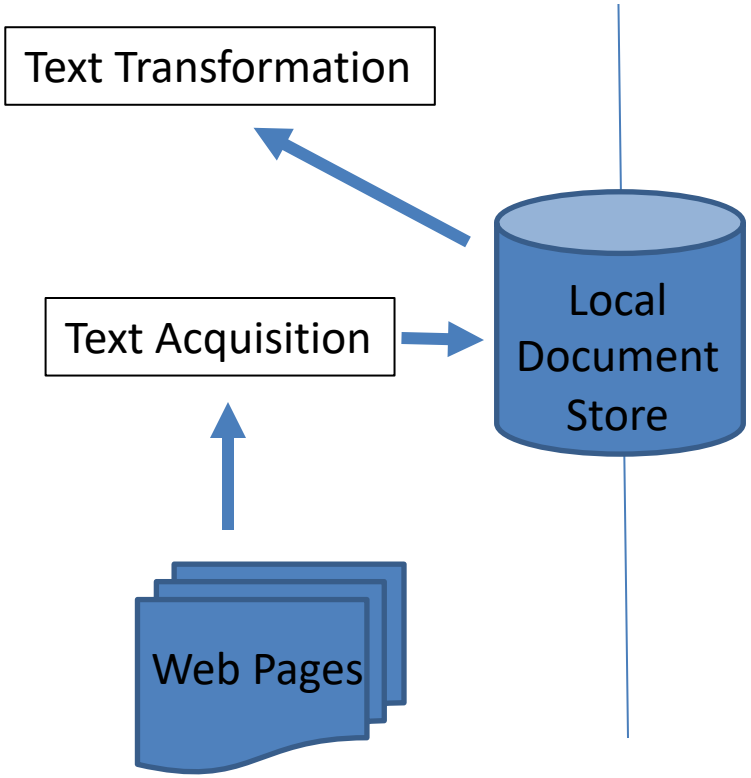


Text Acquisition

- Crawler
- Feeds
- Data dumps
- Text conversion (e.g. PDF -> text)
- Document Store

Architecture

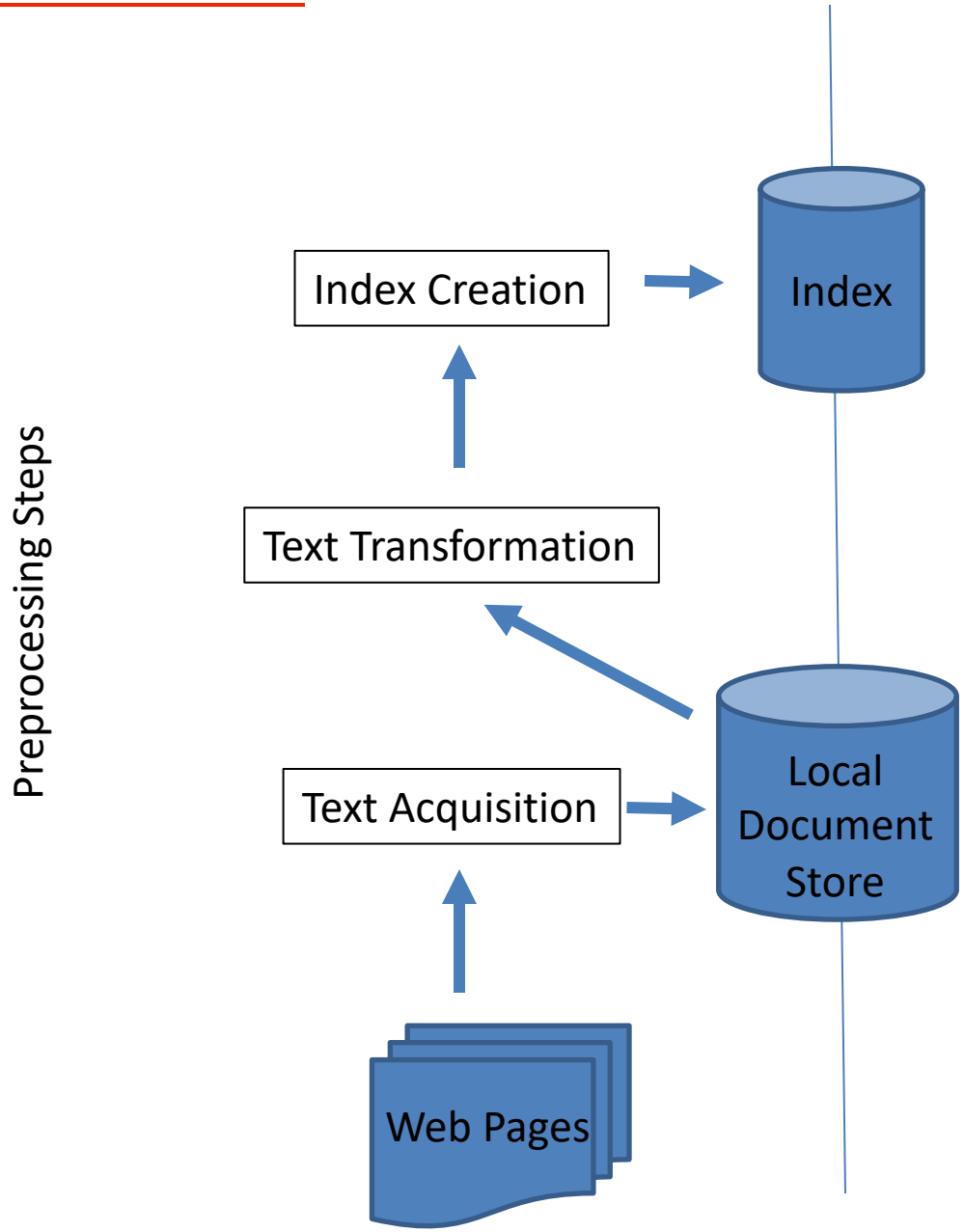
Preprocessing Steps



Text Transformation

- Parser = Tokenizer + Structure
- Stopping : removing words “the”, “of”, etc.
- Stemming : grouping similar words together (e.g. fishes -> fish)
- Link extraction and analysis: how popular is a certain link
- Information Extraction (text structure)
- Classifier (topic, non-content, spam, etc.)

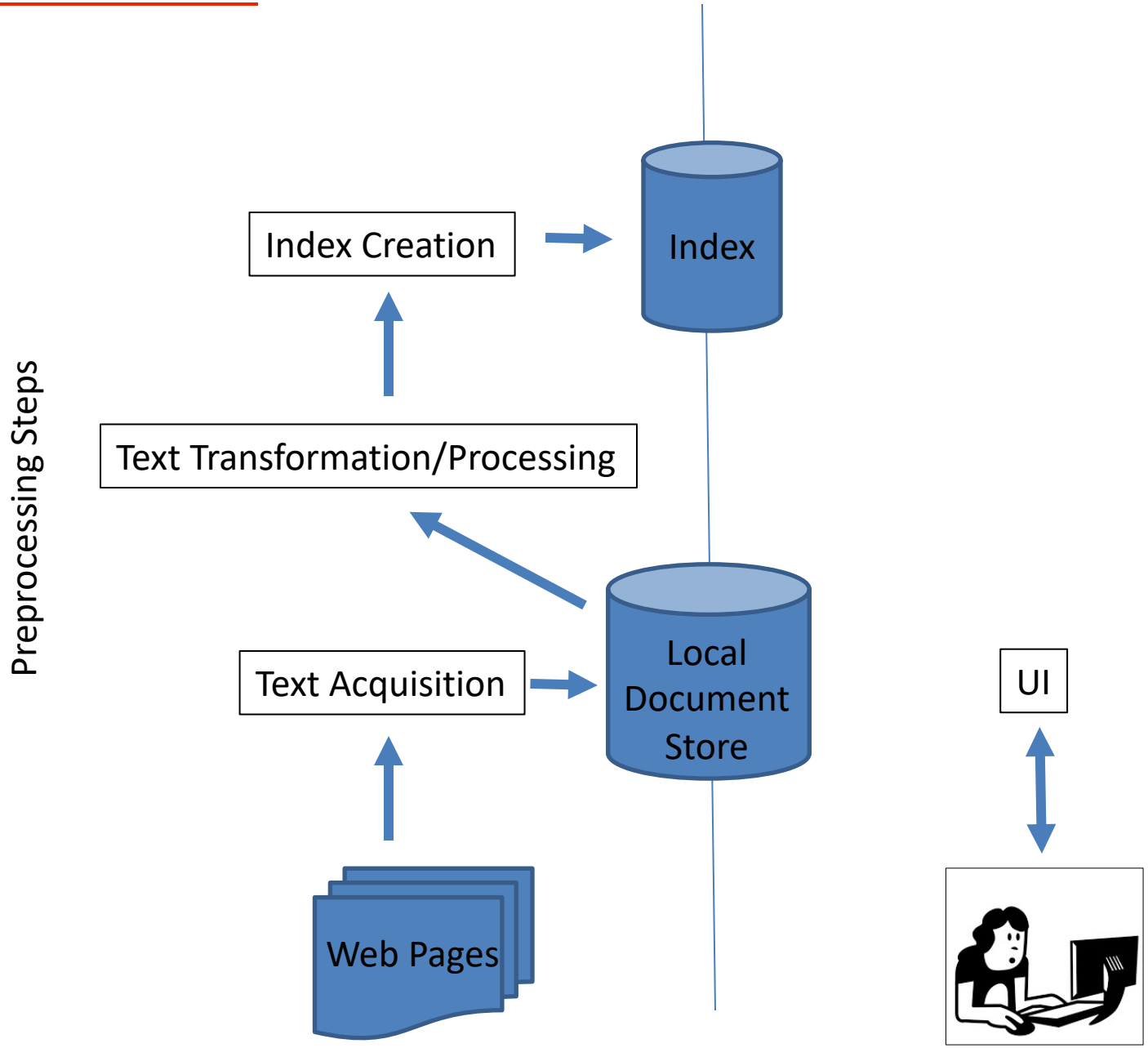
Architecture



Index Creation

- Corpus statistics
- Term weighting : how important is a word in documents
- Index inversion: doc → term ➔ term → doc
- Index distribution: essential for large indexes

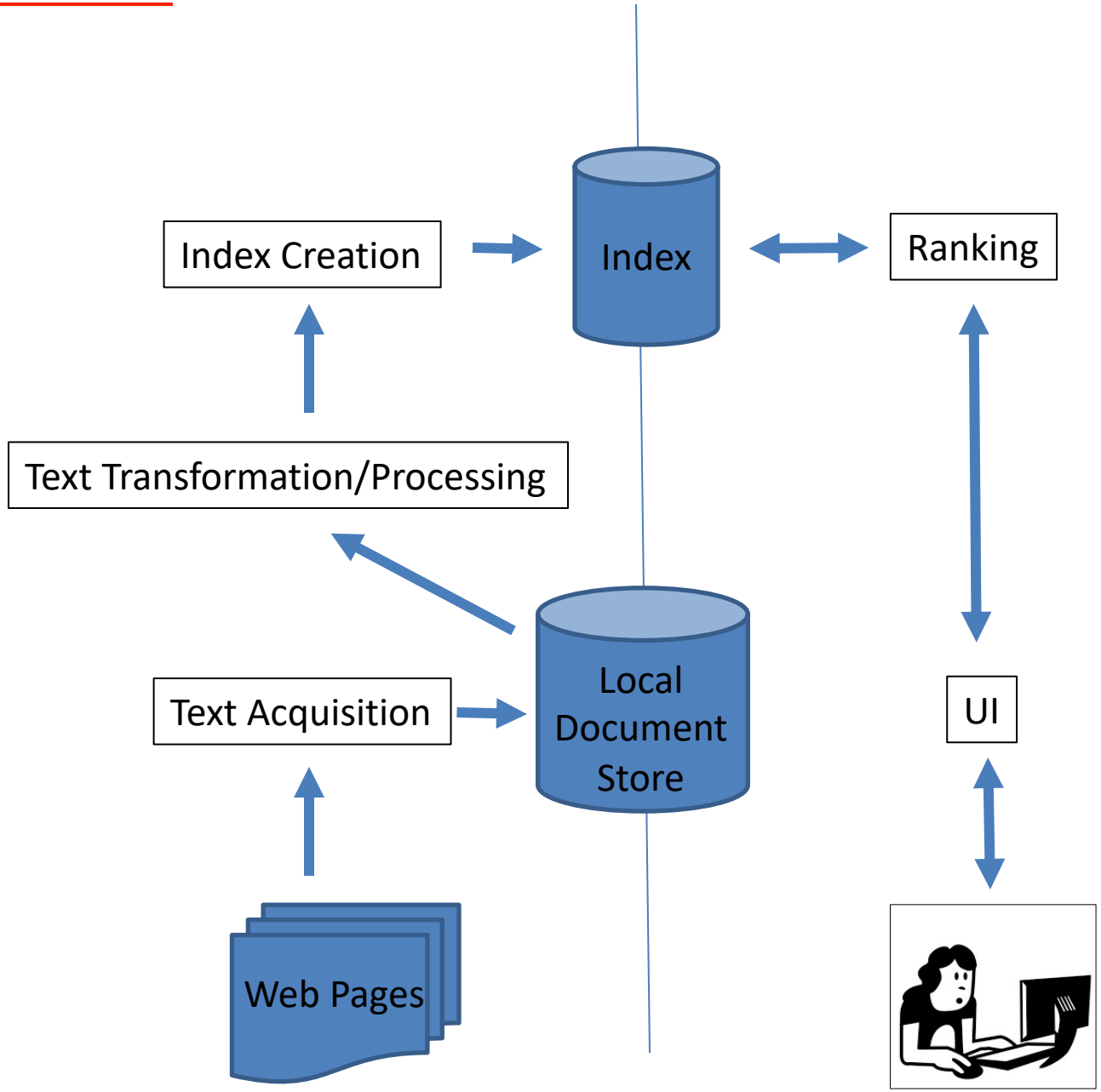
Architecture



User Interaction

- Query input : keywords, operators (e.g. AND), etc.
- Query transformation
  - Spell checking
  - Query suggestion and expansion
- Results output
  - Summaries of the pages
  - Highlighting important words

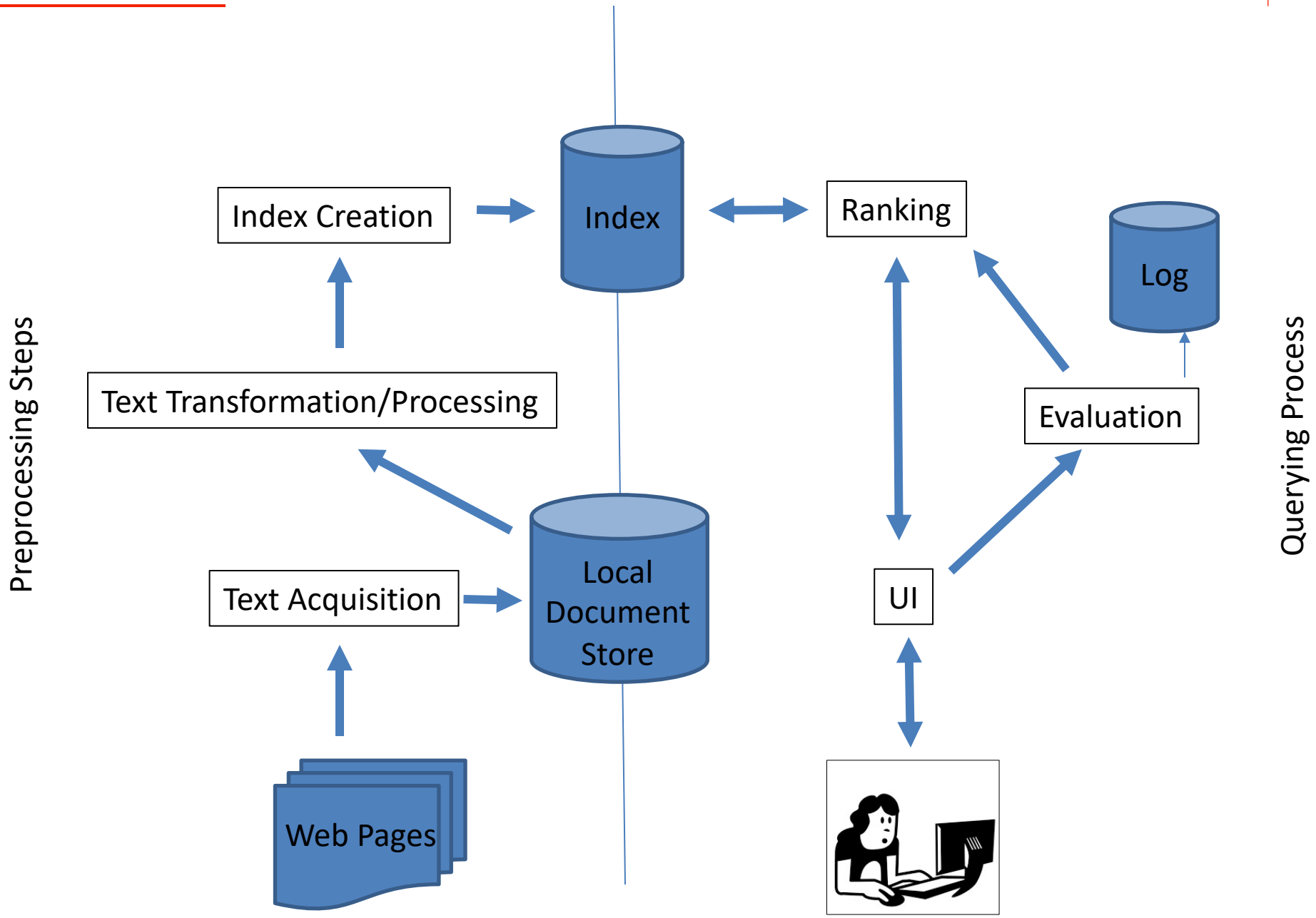
Architecture



Ranking

- Relevance Scoring : how well each doc matches the query
- Performance optimization : decrease response time
- Distribution
  - Query broker : allocate queries to different processors

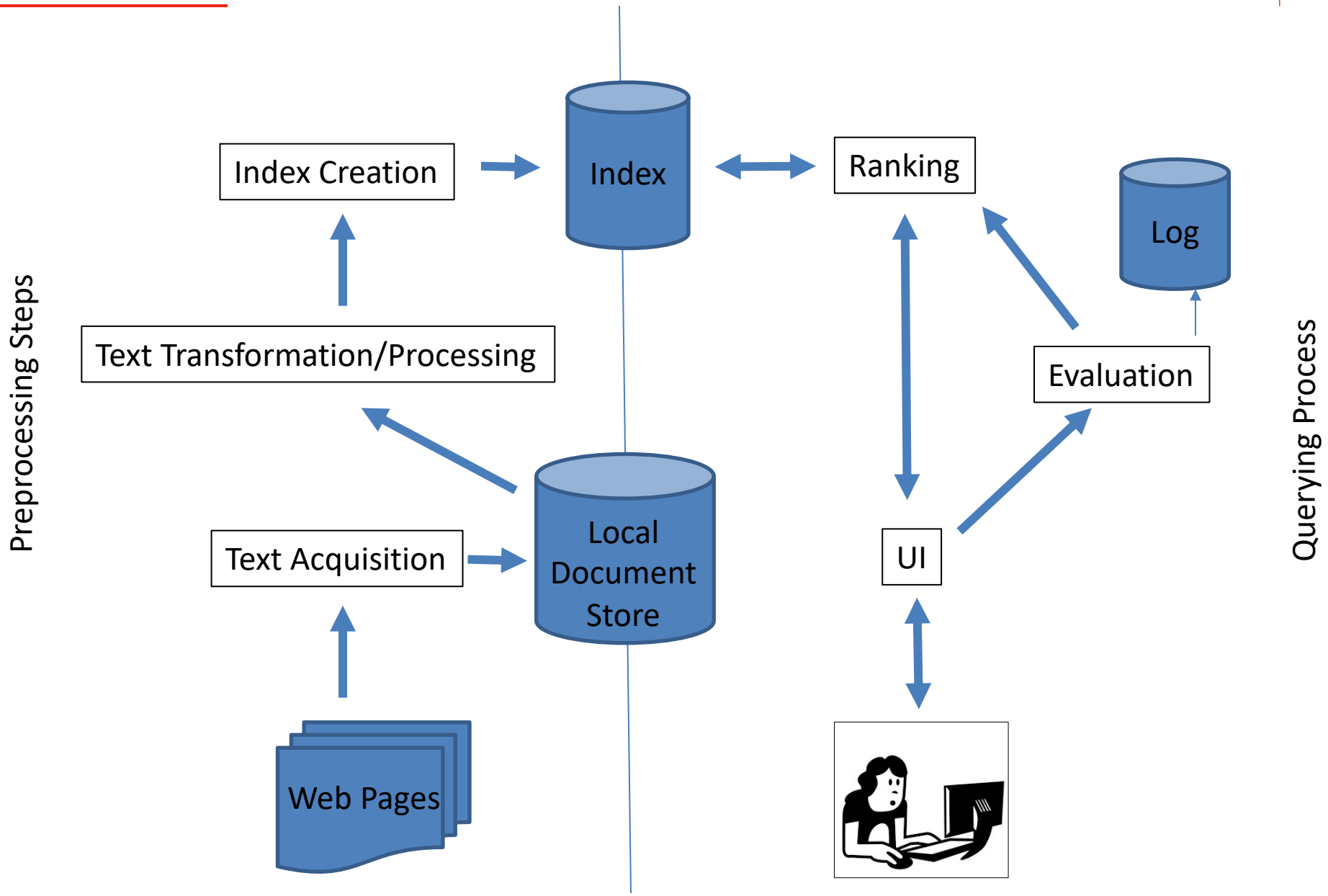
Architecture



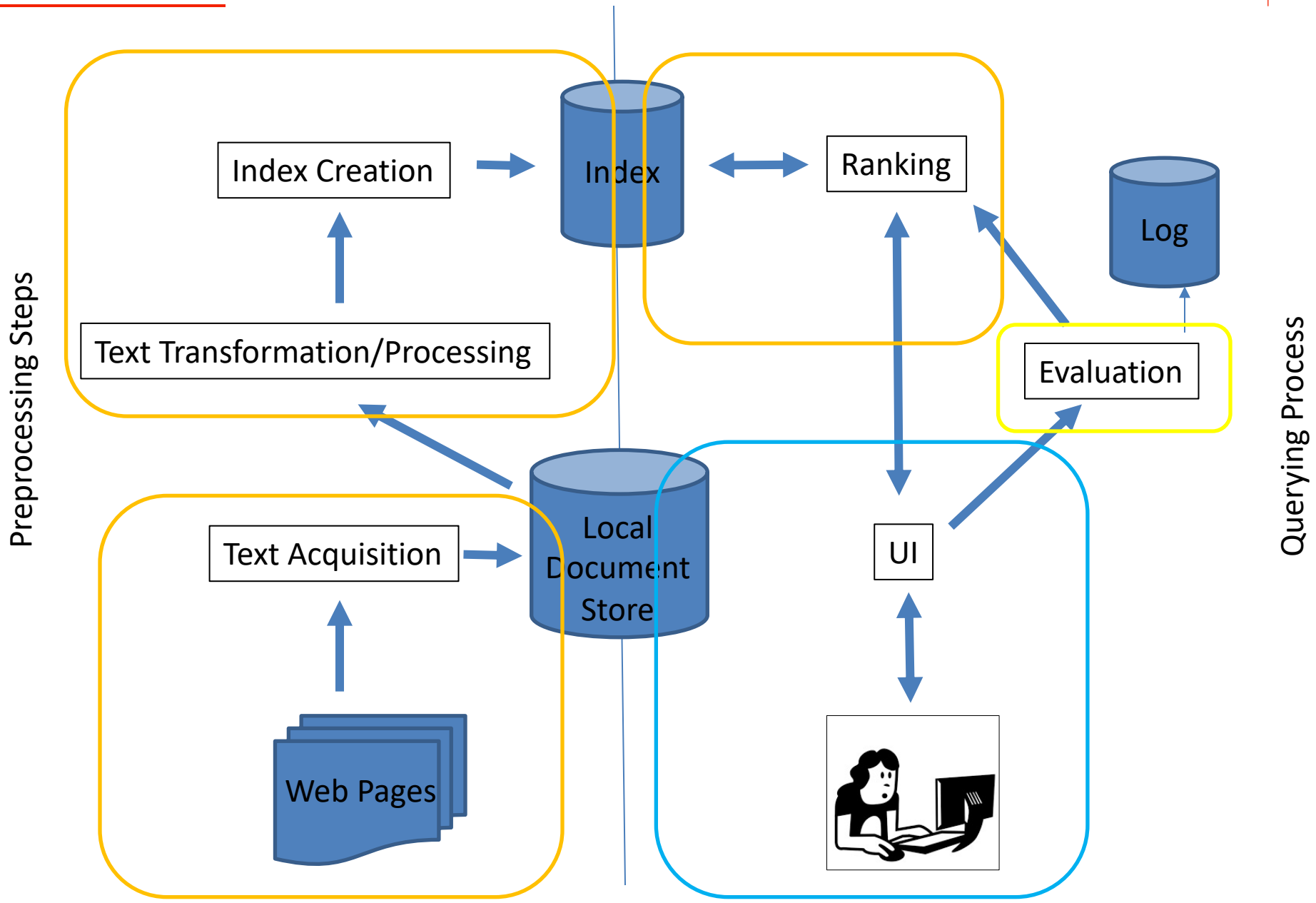
Evaluation

- Logging
  - Improves the search engine in the long run
- Ranking analysis
  - Considering the user behavior, the search engine analyzes if the ranking heuristics work well for its public
- Performance analysis
  - Humans expect fast answers, so the search engine is monitored to improve the performance in the long run

Architecture



Architecture



Assignment 1 : Tokenizer from scratch

You should write the tokenizer in Python

(3.6+; but preferably 3.6 because this is what you will have in the openlab.ics.uci.edu machines).

This will help you with your next two assignments!

Assignment 1 : Tokenizer from scratch

Very important: **At certain points, the assignment may seem underspecified – this is by design.**

In those cases, make your own choices and assumptions and be prepared to defend them.

Assignment 1 : Tokenizer from scratch

Your program must run!

It must be executable from the command line.

You should get the file names from command line arguments.

Reminder!

- Remember to sign the Course Policies ‘quiz’
  - 1 bonus point for doing this in due time!