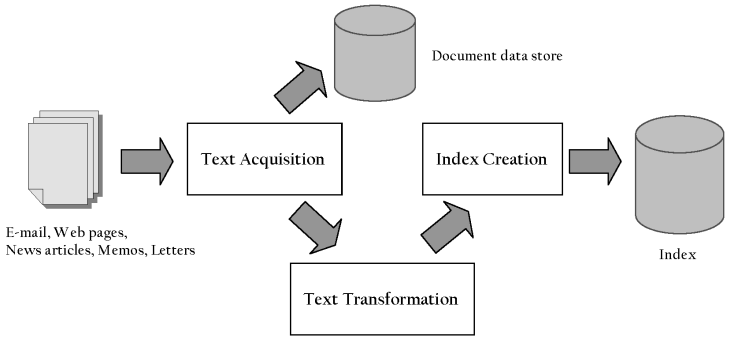
# Information Retrieval

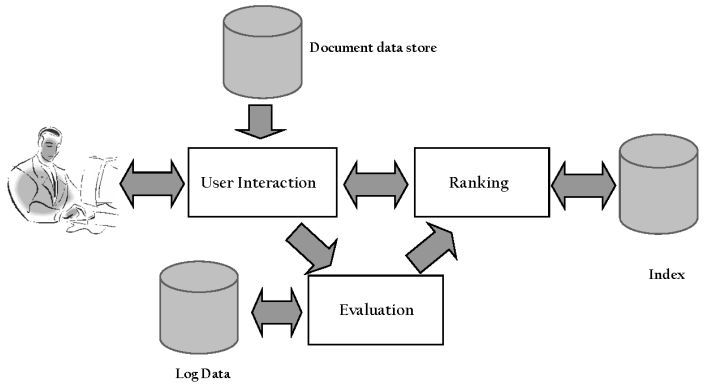
INFO 4300 / CS 4300

- Last class
  - Precision/recall exercise
  - Search engine architecture
    - » The indexing process
    - » The querying process
- Today
  - Web crawlers
    - » Retrieving web pages
    - » Crawling the web

## Indexing Process




## Query Process




## Index Creation

- Document Statistics
  - Gathers counts and positions of words and other features
  - Ranking algorithm uses to compute doc scores
- Weighting
  - Computes weights for index terms
  - Used in ranking algorithm
  - 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
  - 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 on a network
  - 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

## Information Retrieval

INFO 4300 / CS 4300

- Last class
  - Precision/recall exercise
  - Search engine architecture
    - » The indexing process
- Today
  - » The querying process
  - Web crawlers
    - » Retrieving web pages
    - » Crawling the web

## User Interaction

- Query input
  - Provides interface and parser for ***query language***
  - Most web queries are very simple (few **operators**), other applications may use forms
  - Query language used to describe more complex queries and results of query transformation
    - » e.g., Boolean queries, Indri and Galago query languages
    - » similar to SQL language used in database applications
    - » 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 (e.g. tokenization, stopping)
  - ***Spell checking*** and ***query suggestion*** provide alternatives to original query
  - ***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
  - May provide ***clustering*** and other visualization tools

## Ranking

- Scoring
  - Calculates scores for documents using a ranking algorithm
  - 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

## Evaluation

- Logging
  - Logging user queries and interaction is crucial for improving search effectiveness and efficiency
  - *Query logs* and *clickthrough data* or *dwell time* 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

## How Does It *Really* Work?

- This course explains these 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
    - » i.e., explain a small number of approaches in detail rather than many approaches
  - "Importance" based on research results and use in actual search engines

## Information Retrieval

INFO 4300 / CS 4300

- Last class
  - Precision/recall exercise
  - Search engine architecture
    - » The indexing process
- Today
    - » The querying process
  - Web crawlers
    - » Retrieving web pages
    - » Crawling the web

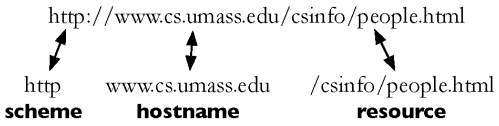
## Web Crawler

If the right documents are not stored in the search engine, no search technique will find relevant information!!

- **Crawlers:** find and download web pages automatically
  - Provide the collection for searching
  - What pages should we search?

## Retrieving Web Pages

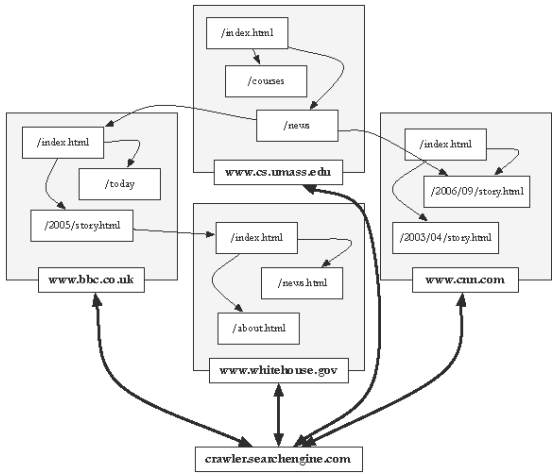
- Every page has a unique ***uniform resource locator*** (URL)
- Web pages are stored on ***web servers*** that use **HTTP** to exchange information with client software
- e.g.,




## Retrieving Web Pages

- Web crawler client program connects to a ***domain name system*** (DNS) server
- DNS server translates the hostname into an ***internet protocol*** (IP) address and tries to connect to a server with that address
- Crawler then attempts to connect to server host using specific ***port***
- After connection, crawler sends an HTTP request to the web server to request a page
  - usually a GET request

## Crawling the Web




## Web Crawler

- Starts with a set of ***seeds***, which are a set of URLs given to it as parameters
- Seeds are added to a URL ***request queue***
- Crawler starts fetching pages from the request queue
- Downloaded pages are parsed to find link tags that might contain other useful URLs to fetch
- New URLs added to the crawler's request queue, or ***frontier***
- Continue until no more new URLs or disk full