Information Retrieval

• “Information retrieval is a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information.” (Salton, 1968)

• General definition that can be applied to many types of information and search applications

• Primary focus of IR since the 50s has been on text and documents
What is a Document?

• Examples:
  – web pages, email, books, news stories, scholarly papers, text messages, Word™, Powerpoint™, PDF, forum postings, patents, IM sessions, etc.

• Common properties
  – Significant text content
  – Some structure (e.g., title, author, date for papers; subject, sender, destination for email)

Documents vs. Database Records

• Database records (or tuples in relational databases) are typically made up of well-defined fields (or attributes)
  – e.g., bank records with account numbers, balances, names, addresses, social security numbers, dates of birth, etc.

• Easy to compare fields with well-defined semantics to queries in order to find matches

• Text is more difficult
Documents vs. Records

• Example bank database query
  – *Find records with balance > $50,000 in branches located in Amherst, MA.*
  – Matches easily found by comparison with field values of records

• Example search engine query
  – *bank scandals in western mass*
  – This text must be compared to the text of entire news stories

Comparing Text

• Comparing the query text to the document text and determining what is a good match is the core issue of information retrieval

• Exact matching of words is not enough
  – Many different ways to write the same thing in a “natural language” like English
  – e.g., does a news story containing the text “bank director in Amherst steals funds” match the query?
  – Some stories will be better matches than others
Dimensions of IR

• IR is more than just text, and more than just web search
  — although these are central
• People doing IR work with different media, different types of search applications, and different tasks

Other Media

• New applications increasingly involve new media
  — e.g., video, photos, music, speech
• Like text, content is difficult to describe and compare
  — text may be used to represent them (e.g. tags)
• IR approaches to search and evaluation are appropriate
Dimensions of IR

<table>
<thead>
<tr>
<th>Content</th>
<th>Applications</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Web search</td>
<td>Ad hoc search</td>
</tr>
<tr>
<td>Images</td>
<td>Vertical search</td>
<td>Filtering</td>
</tr>
<tr>
<td>Video</td>
<td>Enterprise search</td>
<td>Classification</td>
</tr>
<tr>
<td>Scanned docs</td>
<td>Desktop search</td>
<td>Question answering</td>
</tr>
<tr>
<td>Audio</td>
<td>Forum search</td>
<td></td>
</tr>
<tr>
<td>Music</td>
<td>P2P search</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Literature search</td>
<td></td>
</tr>
</tbody>
</table>

IR Tasks

- Ad-hoc search
  - Find relevant documents for an arbitrary text query
- Filtering
  - Identify relevant user profiles for a new document
- Classification
  - Identify relevant labels for documents
- Question answering
  - Give a specific answer to a question
Big Issues in IR

• Relevance
  – What is it?
  – Simple (and simplistic) definition: A relevant document contains the information that a person was looking for when they submitted a query to the search engine
  – Many factors influence a person’s decision about what is relevant: e.g., task, context, novelty, style
  – Topical relevance (same topic) vs. user relevance (everything else)

Big Issues in IR

• Relevance
  – Retrieval models define a view of relevance
  – Ranking algorithms used in search engines are based on retrieval models
  – Most models describe statistical properties of text rather than linguistic
    • i.e. counting simple text features such as words instead of parsing and analyzing the sentences
    • Statistical approach to text processing started with Luhn in the 50s
    • Linguistic features can be part of a statistical model
Big Issues in IR

• Evaluation
  – Experimental procedures and measures for comparing system output with user expectations
    • Originated in Cranfield experiments in the 60s
  – IR evaluation methods now used in many fields
  – Typically use test collection of documents, queries, and relevance judgments
    • Most commonly used are TREC collections
  – Recall and precision are two examples of effectiveness measures

Big Issues in IR

• Users and Information Needs
  – Search evaluation is user-centered
  – Keyword queries are often poor descriptions of actual information needs
  – Interaction and context are important for understanding user intent
  – Query refinement techniques such as query expansion, query suggestion, relevance feedback improve ranking
Search and Information Retrieval

• Search on the Web is a daily activity for many people throughout the world
• Search and communication are most popular uses of the computer
• Applications involving search are everywhere
• The field of computer science that is most involved with R&D for search is information retrieval (IR)

IR and Search Engines

• A search engine is the practical application of information retrieval techniques to large scale text collections
• Web search engines are best-known examples, but many others
  – Open source search engines are important for research and development
    • e.g., Lucene, Lemur/Indri, Galago
• Big issues include main IR issues but also some others
IR and Search Engines

Information Retrieval

- Relevance
  - Effective ranking
- Evaluation
  - Testing and measuring
- Information needs
  - User interaction

Search Engines

- Performance
  - Efficient search and indexing
- Incorporating new data
  - Coverage and freshness
- Scalability
  - Growing with data and users
- Adaptability
  - Tuning for applications
- Specific problems
  - e.g. Spam

Search Engine Issues

- Performance
  - Measuring and improving the efficiency of search
    - e.g., reducing response time, increasing query throughput, increasing indexing speed
  - Indexes are data structures designed to improve search efficiency
    - designing and implementing them are major issues for search engines
Search Engine Issues

• Dynamic data
  – The “collection” for most real applications is constantly changing in terms of updates, additions, deletions
    • e.g., web pages
  – Acquiring or “crawling” the documents is a major task
    • Typical measures are coverage (how much has been indexed) and freshness (how recently was it indexed)
  – Updating the indexes while processing queries is also a design issue

Search Engine Issues

• Scalability
  – Making everything work with millions of users every day, and many terabytes of documents
  – Distributed processing is essential

• Adaptability
  – Changing and tuning search engine components such as ranking algorithm, indexing strategy, interface for different applications
Spam

• For Web search, spam in all its forms is one of the major issues
• Affects the efficiency of search engines and, more seriously, the effectiveness of the results
• Many types of spam
  – e.g. spamdexing or term spam, link spam, “optimization”
• New subfield called adversarial IR, since spammers are “adversaries” with different goals

Search Engine Architecture

• A software architecture consists of software components, the interfaces provided by those components, and the relationships between them
  – describes a system at a particular level of abstraction
• Architecture of a search engine determined by 2 requirements
  – effectiveness (quality of results) and efficiency (response time and throughput)
Indexing Process

- **Text acquisition**
  - identifies and stores documents for indexing
- **Text transformation**
  - transforms documents into *index terms* or *features*
- **Index creation**
  - takes index terms and creates data structures (*indexes*) to support fast searching
Query Process

- User interaction
  - supports creation of results
- Ranking
  - uses query and indexes to generate ranked list of documents
- Evaluation
  - monitors and measures effectiveness and efficiency (primarily offline)
Indexing Process

Details: 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
    • e.g., web feeds for news, blogs, video, radio, tv
  – RSS is common standard
    • RSS “reader” can provide new XML documents to search engine

• Conversion
  – Convert variety of documents into a consistent text plus metadata format
    • e.g. HTML, XML, Word, PDF, etc. → XML
  – Convert text encoding for different languages
    • Using a Unicode standard like UTF-8

Text Acquisition

• Document data store
  – Stores text, metadata, and other related content for documents
    • Metadata is information about document such as type and creation date
    • Other content includes links, anchor text
  – Provides fast access to document contents for search engine components
    • e.g. result list generation
  – Could use relational database system
    • More typically, a simpler, more efficient storage system is used due to huge numbers of documents
Text Transformation

• Parser
  – Processing the sequence of text tokens in the document to recognize structural elements
    • e.g., titles, links, headings, etc.
  – *Tokenizer* recognizes “words” in the text
    • must consider issues like capitalization, hyphens, apostrophes, non-alpha characters, separators
  – *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

Text Transformation

• Stopping
  – Remove common words
    • e.g., “and”, “or”, “the”, “in”
  – Some impact on efficiency and effectiveness
  – Can be a problem for some queries

• Stemming
  – Group words derived from a common stem
    • e.g., “computer”, “computers”, “computing”, “compute”
  – Usually effective, but not for all queries
  – Benefits vary for different languages
Text Transformation

- **Link Analysis**
  - Makes use of *links* and *anchor text* in web pages
  - 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

- **Information Extraction**
  - Identify classes of index terms that are important for some applications
    - e.g., *named entity recognizers* identify classes such as *people, locations, companies, dates*, etc.

- **Classifier**
  - Identifies class-related metadata for documents
    - i.e., assigns labels to documents
    - e.g., topics, reading levels, sentiment, genre
  - Use depends on application
Index Creation

• Document Statistics
  – Gathers counts and positions of words and other features
  – Used in ranking algorithm

• Weighting
  – Computes weights for index terms
  – Used in ranking algorithm
  – e.g., \textit{tf.idf} weight

  • Combination of \textit{term frequency} in document and \textit{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
  - 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
User Interaction

• Query input
  – Provides interface and parser for *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
    • 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
  – *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 $\sum q_i \cdot d_i$
    - $q_i$ 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 click through data 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