Introduction to Search Engine Technology
CS 236620, Technion, Winter 2015/16

What this Course is All About

- Will try to sample the science that drives Web search engines, highlighting several areas of ongoing world-wide (academic and industrial) research
- Algorithms, data structures, system issues, theoretical problems and monetization
- Just the tip of the iceberg
Administration

- Reception hour: by prior appointment, after class
- Email: searchenginecourse@yahoo.com
- Homework grader: TBD
- Grading:
  - 3-4 homework exercises, worth 40 points
  - Dry assignments will be submitted individually; wet assignments will be submitted in pairs
  - (60 points) Final exam

Mission Impossible?

Search engines:

- Crawl and index **tens of billions** of documents
- Answer **billions** of queries per day (G over 3B/day)
- Devote less than **1 second** of processing time per query execution

Users:

- Submit **very short** queries (averaging about 2.6 terms)
- Expect to receive **the most relevant** results on the Web
- In a **blink of an eye**

In terms of 1990 Information Retrieval research - almost unimaginable!
Web Searchers - Observations

- Make ill defined queries
  - Short (2.54 terms average, 80% contain less than 3 words)
  - Use imprecise and often misspelled terms
  - Unfamiliar with query syntax (80% queries without operator)
- Wide variance in information needs, expectations, education/knowledge, screen sizes, IP bandwidth, patience
  - Different modalities (mobile, desktop) = different needs and expectations, even with same person
- Specific behavior (desktop and laptop)
  - 85% look over one result screen only (mostly “above the fold”)
  - 78% of queries are not modified
- Overall, we as users are investing low cognitive effort per query (formulating and looking at results)

The Information Need Behind the Query
[Broder, SIGIR Forum 36(2), 2002]

- Informational
  - I want to learn more about the “roman empire”
  - Many possible fine results
- Navigational
  - Find me the home page of “el al”
  - In principle, a single correct result
  - In practice, correct result may depend on language/locale
- Transactional
  - I want to buy a “digital camera”
  - Good results: e-tailors that sell digital cameras
### Evolution of Search Engines

- Pre-search: human edited directories
  - High quality, authoritative, reliable
  - But inherently non-scalable and doesn’t cater to niches
  - 1994, the *Yahoo* directory

- First generation – use only “on page”, text data
  - Information Retrieval techniques such as TF/IDF, Vector Space model adapted to HTML
  - Scalable, automatic
  - But ranking is poor and easy to spam
  - 1995-1997 AltaVista, Excite, Lycos, etc

---

### Evolution of Search Engines (2)

Second generation - use off-page, web-specific data

- Link analysis
- Anchor-text (how authors refer to a page when linking to it)
- Click-through data (what results people click on)

- From 1998; made popular by Google and used by everyone since

- Ranking is much improved, especially for “navigational” queries
- But popularity rules and results are very homogenous
- Spam evolving into a cat-and-mouse game
Evolution of Search Engines (3)

Third generation – answer the “need behind the query”, focusing on the need rather than on the keywords
- Simple context determination
  - Spatial (user location, target location)
  - Query stream (previous queries)
  - Shortcuts via regular expressions
- Integrate multiple sources of data
  - Maps, videos, images, news
- Search assistance tools that help users create good queries and streamline search efforts
  - Spell correction, expand/narrow suggestions, facets
  - From 2006 and on: some of this happens as you type, before the query is actually evaluated (AJAX)

First half of last decade: Google, Yahoo!, Ask

Evolution of Search Engines (3.5)

More recent efforts (2005-2011):
- Integration of search and text analysis, e.g. understanding concepts in documents and matching those to query intent:
  - Objects (named entities)
  - Directives/modifiers (“download”, “reviews” etc.)
- “Social search” – making use of UGC (user-generated content), social networks, etc.
- Optimizing the experience delivered by the search result page as a whole rather than as a collection of 10 independent “blue links”
  - Page real-estate is scarce, need to make every pixel count
  - Emphasis on diversity of results
- Apps – actionable widgets associated with search results
Evolution of Search Engines (4)

More recent efforts (2012-now):

- Knowledge graphs provide rich results
- More elaborate personalization
- Divergence of desktop and mobile searches
  - Mobile searches emphasize location-based results, "cards" interface
  - Mobile searches within apps on device
  - Mobile can be voice triggered
- Digital assistants (Siri, Cortana, Google Now) rely heavily on search back-ends

Anatomy of a Search Engine

Image taken from "Searching the Web", Arasu et al., TOIT 1(1), 2001
Weeks 1-2: Introduction to Information Retrieval

“Classic” Information Retrieval

Given a query, the system retrieves a set B of documents.

Every retrieved document is either relevant or irrelevant to the query.

Quality metrics:
- **Recall**: \( \frac{A \cap B}{A} \)
- **Precision**: \( \frac{A \cap B}{B} \)

18 October 2015
CS236620 Search Engine Technology
Recall and Precision on the Web

- Relevance of document to queries is not binary – there are many shades of gray
- Broad-topic queries: an abundance problem
  - Precision dominates: a few good results will satisfy users
- Narrow-topic queries: needle in a huge haystack problem
  - Recall demands engines cover significant portions of the Web
- Common measures:
  - Precision@10 and positional precision (NDCG)
  - Diversity: positive recall for many aspects of the query

Classic and Modern IR Issues

- Polysemy (ambiguity): what is the intent behind the query “jordan”?
  - Should the SERP* will represent all aspects?
- Synonymy: elevator/lift, movie/film
- Short queries require the addition of proximity considerations (query term "hits" should be in proximity to each other) – how does this fit with IR theory?
- Information Retrieval merits a course of its own!

* SERP – Search Results Page
Weeks 3-5: Link Structure Analysis

Navigational Queries & Anchor Text

- Anchor text – the highlighted clickable text of a link
- Physically in page A, but actually describes the content of page B
- Many times, concisely defines page B
  - Often, better than the text present on page B itself
- Anchor text is the most important factor in treating navigational queries
- Famous pop-culture examples: “French Military Victories”, “miserable failure”
The Web as a Graph

Pages as graph nodes, hyperlinks as edges
- Sometimes sites are taken as the nodes

Some natural questions:
1. Distribution of the number of in-links to a page
2. Distribution of the number of out-links from a page
3. Distribution of the number of pages in a site
4. Connectivity: is it possible to reach most pages from most pages?
5. Are there models that explain how the Web graph evolved to its current shape?

Link-Structure Analysis Principles

- The connectivity patterns between Web pages contain a gold mine of information
- A link from page \( a \) to page \( b \) can often be interpreted as:
  1. A recommendation, by \( a \)'s author, of the contents of \( b \).
  2. Evidence that pages \( a, b \) share some topic of interest.
- A co-citation of \( a \) and \( b \) (by a third page \( c \)) may also constitute evidence that \( a, b \) share some topic of interest
- Course will cover: HITS, PageRank, SALSA
Weeks 6-8: The Inverted Index

Inverted Index Example

Doc #1

Lexicon

<table>
<thead>
<tr>
<th>Term</th>
<th>DF</th>
<th>Doc #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>1</td>
<td>(1; 1,3,6)</td>
</tr>
<tr>
<td>Good</td>
<td>2</td>
<td>(1; 2) – (2; 2)</td>
</tr>
<tr>
<td>Bad</td>
<td>2</td>
<td>(1; 4) – (3; 5)</td>
</tr>
<tr>
<td>And</td>
<td>3</td>
<td>(1; 5) – (2; 6) – (3; 4,8)</td>
</tr>
<tr>
<td>Ugly</td>
<td>2</td>
<td>(1; 7) – (3; 3)</td>
</tr>
<tr>
<td>As</td>
<td>1</td>
<td>(2; 1,3)</td>
</tr>
<tr>
<td>It</td>
<td>2</td>
<td>(2; 4) – (3; 2,6)</td>
</tr>
<tr>
<td>Gets</td>
<td>1</td>
<td>(2; 5)</td>
</tr>
<tr>
<td>More</td>
<td>2</td>
<td>(2; 7) – (3; 9)</td>
</tr>
<tr>
<td>Is</td>
<td>1</td>
<td>(3; 1,7)</td>
</tr>
</tbody>
</table>

Doc #2

As good as it gets, and more

1 2 3 4 5 6 7

Doc #3

Is it ugly and bad? It is, and more!

1 2 3 4 5 6 7 8 9

Course will cover:

- Index representation and batch construction
- Query evaluation schemes
- Distributed indexing and query evaluation
Weeks 9-10: Crawlers and Caches

Crawlers

- The World Wide Web can be viewed as a directed graph
- Nodes are web objects, such as html pages, image files, pdf files, forms, etc...
- Nodes are uniquely identified by URLs
- Edges are html hyperlinks, pointing from an html page to its embedded objects or to other web pages
- Writing a simple crawler is a fairly easy task – basically, BFS/DFS code that handles HTTP downloads and HTML parsing
- However, there are many issues that need to be considered when writing a robust distributed Web-scale crawler
Query Result Caching

Exploits locality of reference in the query streams that are submitted to search engines:

- Query popularities vary widely, from the extremely popular to the very rare
- Although a minority, significant number of users still view multiple *result pages* per query (motivates prefetching)

Successful caching and prefetching of results can:

- Lower the number/cost of query executions
- Shorten the engine’s response time and lower its hardware requirements

---

Week 11: Search Assistance
Search Assistance

- Help users complete their online tasks quickly and easily
- Examples on next slides, from multiple search engines, show:
  - Task-oriented mashups
  - Query completion tools
  - Suggestions of related concepts and queries
  - Federation of multiple sources of information
  - Structured “web of objects”/“knowledge graph” results
- Challenges:
  - Which experience to trigger for which query/user?
  - Which data items to trigger given an experience?
  - Lots(!!!) of data mining – including usage mining - behind the scenes
- The bigger picture: optimize which (layout), what (content), where (on the page) and how (to render)
Weeks 12-13: Comp. Advertising, Long-Tail Economics and RecSys
The Online Advertising Market

- Internet advertising is a $10^{10}$ business that is growing faster than old media advertising (radio, TV, newspapers & magazines, mail, outdoors)
- Traditional advertising:
  - Few, expensive opportunities
  - Targeting en-masse, by immediate context only
  - Difficult to measure effectiveness
- Internet advertising:
  - Tens of Billions of opportunities daily
  - Open to personalization via rich context of impression
  - Effectiveness is measurable: can measure click-through rates as % of impressions, and conversions as % of clicks

Course Will Cover

- Online advertising types: sponsored search, contextual advertising, display advertising, content marketing, native
- Pricing models: CPM, CPC, CPA
- Auction mechanisms
- A glimpse into a very complex ecosystem
The Long Tail

How the infinite choice available in e-commerce and the ease of publishing online have redefined the business of media & entertainment

Mottos:
- “The biggest money is in the smallest sales”
- “Embrace niches”
Long Tail Economics and Recommender Systems

- People cannot intelligently navigate product or media catalogs consisting of millions of items
  - e-Inventories of books, songs, movies, albums are practically infinite
  - In particular, niche audiences cannot find niche content
- Recommender systems are key to Long Tail economics
  - Find an item you like, and they’ll point you to other items you may like
- Pioneered by Amazon; today, used “everywhere”

Customers who bought items in your Recent History also bought:

Information Retrieval Primer

Many of the following slides are courtesy of Aya Soffer, IBM Haifa Research Lab, and David Carmel, Yahoo Labs
Indexing/Retrieval – Basics

2 Main Stages
- Indexing offline process - involves pre-processing and storing of information into a repository – an index
- Retrieval/runtime online process – upon receiving a query, access the index to find & rank documents relevant to the query

Basic Concepts:
- Document – any piece of information (book, article, database record, Web page, image, video, song)
- Query – some text representing the user’s information need
- Relevance – a binary relation (a predicate) between documents and queries \( R(d,q) \)
  - Obviously a simplification of subjective quality with many shades of gray

Defining Document Relevance to a Query

Although Relevance is the basic concept of IR, it lacks a precise definition

Difficulties:
- User/intent dependent
- Time/place dependent
- In practice, is relative to other documents

Simplifying assumption for now:
- \( D \) - the set of all documents in the collection (corpus),
- \( Q \) - the set of all possible queries,
- \( R: D \times Q \to \{0,1\} \) is well defined
- \( d \) is “relevant” to \( q \) \iff \( R(d,q) = 1 \)
Index Building: Text Profiling

- Documents/Queries are profiled to generate a canonical representation.
- The profile is usually based on the set of indexing units (terms) in the text.
- Indexing units (terms) are generally representative words in the text.
  - How to select representative units?
  - For the moment, let’s take all the words in the given document/query.

In the beginning, God created the heaven and the earth. And the earth was without form and void.

...and, beginning, created, earth, form, god, heaven, in, the, void, was, without
...3 ...1 ...1 ...1 ...2 ...1 ...1 ...1 ...1 ...1 ...1 ...1 ...1 ...1 ...

Document Profiling, More Formally

Given a collection of documents (a corpus)
- All the terms in the collection can be labeled $t_1, t_2, \ldots, t_N$.
- The profile of document $d_j$ is an N-dimensional vector,
  $$d_j \rightarrow (w_{1j}, w_{2j}, \ldots, w_{Nj})$$
  - where
    - $w_{ij} = 0$ if $t_i$ does not appear in $d_j$
    - $w_{ij} > 0$ otherwise
- The N-dimensional vector space is conceptual – implementations will not actually manipulate such large and sparse vectors.
Matrix Representation of an Index

<table>
<thead>
<tr>
<th></th>
<th>d₁</th>
<th>d₂</th>
<th>...</th>
<th>dₘ</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>w₁₁</td>
<td>w₁₂</td>
<td></td>
<td>w₁ₘ</td>
</tr>
<tr>
<td>t₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tₙ</td>
<td>wₙ₁</td>
<td></td>
<td></td>
<td>wₙₘ</td>
</tr>
</tbody>
</table>

Most entries are zero – certainly for large corpora

Retrieval Process

- Assumption: a document not containing any query term is not relevant
- Given a simple query of one term q={tᵢ}
- Use the index for retrieval:
  - 1. Retrieve all documents dⱼ with wᵢⱼ > 0
  - 2. Sort them in decreasing order
  - 3. Return the ranked list of “relevant” documents to the user
- In general: given a user’s query q={t₁…tₖ}:
  - Disjunctive queries: return a (ranked) list of documents containing at least one of the query terms
  - Conjunctive queries: return a (ranked) list of documents containing all of the query terms
The Boolean Model

- Simple model based on Set Theory
- Queries are specified as Boolean expressions
  - indexing units are words
  - Boolean Operator: OR, AND, NOT
- Example:
  \[ q = \text{"java" AND "compilers" AND ("unix" OR "linux")} \]
- Relevance: a document is relevant to the query if it satisfies the query Boolean expression

### Boolean Model - Example

<table>
<thead>
<tr>
<th></th>
<th>(d_1)</th>
<th>(d_2)</th>
<th>(d_3)</th>
<th>(d_4)</th>
<th>(d_5)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>a</strong></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td><strong>b</strong></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>c</strong></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ q = a \text{ AND (} b \text{ OR (NOT } c\text{))} \]
**Boolean Model – Pros & Cons**

- **Pros:**
  - Fast (bitmap vector operations)
  - Binary decision (doc is “relevant” or not)
  - Some extensions are easy (e.g. synonym support)
- **Cons:**
  - Binary decision - what about ranking?
  - Who speaks Boolean?

**Vector Space Model**

- Documents are represented as vectors in a (huge) N-dimensional space
  - N is the number of terms in the corpus, i.e. size of the lexicon/dictionary
- Query is a document like any other document
- Relevance is measured by similarity:
  - A document is relevant to the query if its representative vector is similar to the query’s representative vector
**Documents as Vectors**

- Star
  - Doc about astronomy
  - Doc about movie stars
- Diet
  - Doc about mammal behavior

---

**Vector-space Model**

- “Relevance” is measured by similarity - the cosine of the angle between doc-vectors and the query vector
- Need to represent the query as a vector in the same vector-space as the documents

\[
Sim(q, d) = \frac{q \cdot d}{|q| \cdot |d|} = \frac{\sum w_{iq} w_{id}}{\sqrt{\sum w_{iq}^2 \sum w_{id}^2}}
\]
Example

\[
Sim(q,d) = \frac{q \cdot d}{|q| \cdot |d|} = \frac{\sum w_{iq} w_{id}}{\sqrt{\sum w_{iq}^2 \sum w_{id}^2}}
\]

\[
sim(q, d_2) = \frac{(0.4 \cdot 0.2) + (0.8 \cdot 0.7)}{\sqrt{(0.4)^2 + (0.8)^2} \cdot \sqrt{(0.2)^2 + (0.7)^2}} = \frac{0.64}{0.42} = 0.98
\]

\[
sim(q, d_1) = \frac{0.56}{\sqrt{0.58}} = 0.74
\]

How to Determine Term Weights?

- Binary weights:
  - \( w_{ij} = 1 \) iff document \( d_i \) contains term \( t_j \), otherwise 0.
  - (e.g. the Boolean model)

- Term frequency (tf):
  - \( w_{ij} = \text{(number of occurrences of } t_j \text{ in } d_i) \)

- What about term importance?
  - E.g. \( q = \text{“galaxy in space”} \).
  - Should an occurrence of the query term “in” in a document contribute the same as an occurrence of the query term “galaxy”? 

18 October 2015 CS236620 Search Engine Technology

51
Determining Term Weights (cont)

- **tf x idf** weighting scheme (Salton 73)
  - **tf** – a monotonic function of the term frequency in the document,
    - e.g. $tf(t,d) = \log(freq(t,d) + 1)$
  - **idf** – the inverse document frequency of a term – a decreasing function of the term total freq $N_t$
    - e.g: $idf(t) = \log(N / N_t)$ (for terms appearing at least once)
    - Intuition: query terms that are rare in the corpus better distinguish the relevant documents from the irrelevant ones
    - $W_{i,j} = tf(t_i,d_j) \times idf(t_i)$

Vector Space Model Pros & Cons

- **Pros**
  - Terms weighting scheme improves retrieval effectiveness
  - Allows for approximate query matching
  - Cosine similarity is a good ranking measure
  - Simple and elegant, with a solid mathematical foundation

- **Cons**
  - Terms are not really orthogonal dimensions due to strong term relationships and dependencies
  - Ranking does not respect conjunctive semantics
  - Term weighting schemes sometimes difficult to maintain in incremental settings, e.g. idf values and document norms frequently change
Practical Considerations

- Considering the proximity of query terms’ occurrences in result documents in ranking formula
- Stop-word elimination
  - Stop-word examples: and, the, or, of, in, a, an, to, …
- Linguistic processing of terms (stemming, lemmatization, synonym expansion, compounds) and their effects on recall/precision