Web Search
Structure of the Web

- The Web is a complex network (graph) of nodes & links that has the appearance of a self-organizing structure.

- The figure shows a partial map of the Internet (the physical network over which the Web is built), which traces the paths of packets through the Internet from a host.
# Web Search vs. Data(base) Retrieval

<table>
<thead>
<tr>
<th></th>
<th>Data Retrieval</th>
<th>Web Search</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Matching</strong></td>
<td>Exact Match</td>
<td>Partial (Best) Match</td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td>Deterministic</td>
<td>Probabilistic</td>
</tr>
<tr>
<td><strong>Classification</strong></td>
<td>Monotonic</td>
<td>Polytechnic</td>
</tr>
<tr>
<td><strong>Query language</strong></td>
<td>Artificial</td>
<td>Natural</td>
</tr>
<tr>
<td><strong>Query specification</strong></td>
<td>Complete</td>
<td>Incomplete</td>
</tr>
<tr>
<td><strong>Items wanted</strong></td>
<td>Matching</td>
<td>Relevant</td>
</tr>
<tr>
<td><strong>Error response</strong></td>
<td>Sensitive</td>
<td>Insensitive</td>
</tr>
<tr>
<td><strong>Data representation</strong></td>
<td>Schema</td>
<td>(Mostly) Index terms</td>
</tr>
</tbody>
</table>
Search Taxonomy (Types)

- **Informational (Topical) Search**
  - Finding information about some topic which may be on one or more web pages, such as “sports cars” or “London”

- **Navigational Search**
  - Finding a particular web page, e.g., BYU CS Department homepage, that the user has seen before or is assumed to exist

- **Transactional Search**
  - Finding a site where a task, such as shopping/downloading music/online library browsing, can be performed

- Each type of search is associated w/ different info. need
Search Taxonomy (Types)

- **Navigational Search**

  - The *result* of a navigational query is the homepage of an institution or organization most relevant to the query.

  - When a navigational query is *accurately specified*, a feature, such as Google’s “I’m Feeling Lucky”, will bring the user directly to the required homepage.
<table>
<thead>
<tr>
<th></th>
<th>Information Retrieval</th>
<th>Web Search</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of Information</strong></td>
<td>Mainly Text-Based</td>
<td>Multimedia</td>
</tr>
<tr>
<td><strong>Archived Data Size</strong></td>
<td>Relatively Small</td>
<td>Billions/Trillions of Pages</td>
</tr>
<tr>
<td><strong>Rate of Change</strong></td>
<td>Static</td>
<td>Dynamic</td>
</tr>
<tr>
<td><strong>Replication</strong></td>
<td>Few</td>
<td>Enormous ($\geq 30%$)</td>
</tr>
<tr>
<td><strong>Quality of Pages</strong></td>
<td>High</td>
<td>Vary Dramatically</td>
</tr>
<tr>
<td><strong>Range of Topics</strong></td>
<td>Closed Collection</td>
<td>Wide Open</td>
</tr>
<tr>
<td><strong>Type of Queries</strong></td>
<td>More Complete</td>
<td>Short</td>
</tr>
<tr>
<td><strong>User Expertise Level</strong></td>
<td>Similar</td>
<td>Vary Significantly</td>
</tr>
</tbody>
</table>
Search Engine Optimization (SEO)

- **SEO**: understanding the relative importance of *features* used in search and how they can be manipulated to obtain *better search rankings* for a web page, e.g.,
  - Improve the text used in the **title** tag
  - Improve the text in **heading tags**
  - Make sure that the **domain name** & **URL** contain important keywords
  - Improve the **anchor text** & link structure to the page
Anchor Text

- Anchor text tends to be short, descriptive, and similar to query text.

- Used as a description of the content of the destination page.
  - i.e., collection of anchor text in all links pointing to a page used as an additional text field.

- Incorporating the collection of all the anchor text (as extra text) in links to a page into the ranking algorithm.

- A simple search/ranking algorithm: search through all links in a collection, looking for anchor text that is an exact match for the user’s query.

- Retrieval experiments have shown that anchor text has significant impact on effectiveness for some types of queries, i.e., more than PageRank.
Web Search

- Given the size of the Web, many pages will contain all query terms
  - Ranking algorithm focuses on discriminating between these pages based on the occurrence of query terms
  - Word/term proximity (i.e., close proximity) is important
    - The dependency model: query terms are assumed to appear in close proximity to each other in relevant documents,
    - e.g., given the query “Green party political views”, relevant docs should contain the phrases “green party” & “political views” in relatively close proximity
    - N-grams are commonly used in commercial web search
Link Analysis

- Links are a key component of the Web
- Important for *navigation*, but also for *web search*
  
  E.g., `<a href="http://example.com" >Example website</a>`
  
  “Example website” is the *anchor text*
  
  “http://example.com” is the *destination link*
  
  Both are used by web search engines
Billions of web pages, some are more *informative* than the others

Links can be viewed as information about the *popularity* (*authority?*) of a web page, e.g., eBay

- Can be used by ranking algorithm

*Inlink* count could be used as simple measure

Link analysis algorithms like *PageRank* provide more reliable ratings

- Less susceptible to link spam
PageRank: Random Surfer Model

- Browse the Web using the following algorithm:
  - Choose a random number $r$ between 0 and 1
  - If $r < \lambda$:
    - Go to a random page
  - If $r \geq \lambda$:
    - Click a link at random on the current page
  - Start again

- PageRank of a page is the probability that the “random surfer” will be looking at that page
  - Links from popular pages will increase PageRank of pages they point to
Dangling Links

- Random jump prevents getting stuck on pages that
  - Do not have links
  - Contains only links that no longer point to other pages
  - Have links forming a loop

- Links that point to the first two types of pages are called dangling links
  - May also be links to pages that have not yet been crawled
PageRank

- PageRank ($PR$) of page $C = PR(A)/2 + PR(B)/1$
- More generally,

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L_v}$$

- where $B_u$ is the set of pages that point to $u$, and $L_v$ is the number of outgoing links from page $v$ (not counting duplicate links)
PageRank

- Don’t know PageRank values at start

- Assume equal values (1/3 in this case), then iterate:
  - First iteration: \( PR(C) = 0.33/2 + 0.33 = 0.5, \) \( PR(A) = 0.33, \) and \( PR(B) = 0.17 \)
  - Second: \( PR(C) = 0.33/2 + 0.17 = 0.33, \) \( PR(A) = 0.5, \) \( PR(B) = 0.17 \)
  - Third: \( PR(C) = 0.42, \) \( PR(A) = 0.33, \) \( PR(B) = 0.25 \)

- Converges to \( PR(C) = 0.4, \) \( PR(A) = 0.4, \) and \( PR(B) = 0.2 \)
PageRank

- Taking random page jump into account, 1/3 chance of going to any page when $r < \lambda$

- $PR(C) = \lambda/3 + (1 - \lambda) \cdot (PR(A)/2 + PR(B)/1)$

- More generally, 

\[
PR(u) = \frac{\lambda}{N} + (1 - \lambda) \cdot \sum_{v \in B_u} \frac{PR(v)}{L_v}
\]

- where $N$ is the number of pages, $\lambda$ typically 0.15
1: procedure PageRank(G)
2:   \( G \) is the web graph, consisting of vertices (pages) and edges (links).
3:   \((P, L) \leftarrow G\) \quad \triangleright \text{Split graph into pages and links}
4:   \( I \leftarrow \text{a vector of length } |P| \) \quad \triangleright \text{The current PageRank estimate}
5:   \( R \leftarrow \text{a vector of length } |P| \) \quad \triangleright \text{The resulting better PageRank estimate}
6:   \textbf{for all} entries \( I_i \in I \) \textbf{do}
7:       \( I_i \leftarrow 1/|P| \) \quad \triangleright \text{Start with each page being equally likely}
8:   \textbf{end for}
9:   \textbf{while} \( R \) has not converged \textbf{do}
10:      \textbf{for all} entries \( R_i \in R \) \textbf{do}
11:         \( R_i \leftarrow \lambda/|P| \) \quad \triangleright \text{Each page has a } \lambda/|P| \text{ chance of random selection}
12:      \textbf{end for}
13:      \textbf{for all} pages \( p \in P \) \textbf{do}
14:         \( Q \leftarrow \text{the set of pages } p \text{ such that } (p, q) \in L \text{ and } q \in P \)
15:         \textbf{if} \( |Q| > 0 \) \textbf{then}
16:            \textbf{for all} pages \( q \in Q \) \textbf{do}
17:               \( R_q \leftarrow R_q + (1 - \lambda)I_p/|Q| \) \quad \triangleright \text{Probability } I_p \text{ of being at page } p
18:            \textbf{end for}
19:         \textbf{else}
20:            \textbf{for all} pages \( q \in P \) \textbf{do}
21:               \( R_p \leftarrow R_q + (1 - \lambda)I_p/|P| \)
22:            \textbf{end for}
23:         \textbf{end if}
24:      \( I \leftarrow R \) \quad \triangleright \text{Update our current PageRank estimate}
25:   \textbf{end for}
26:   \textbf{end while}
27:   \textbf{return} \( R \)
28: end procedure