CS6120: Intelligent Media Systems

Dr. Derek Bridge
School of Computer Science & Information Technology
UCC

Web Search

Web Search

Web Search is Special

- Size of web
- Decentralized content creation
- Images and other media
- Dynamic pages
- Naïve end-users
  - rarely type more than 2 or 3 search terms
  - rarely use Advanced Search features
  - typing and spelling errors
  - different query needs
- Revenue through advertising

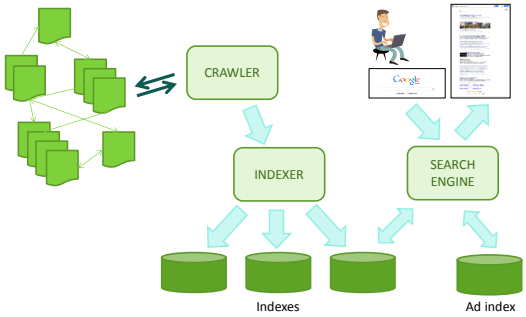
User Query Needs

- Informational
  - Users seek general info on a broad topic, e.g. Napoleon Waterloo, Greece
    - They'll often need to consult multiple pages
  - Or users seek an answer to a specific question
    - They'll need one or two authoritative single sources
- Navigational
  - Users seek the main page for an entity, e.g. UCC
    - They want that to be the first result
- Transactional
  - The search is a prelude to a transaction, e.g. a purchase, a download, a reservation
    - Users want the one page or a small choice of pages where they can perform this transaction

Web Search Results

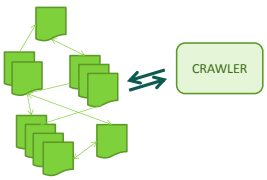
- Possible criteria
  - relevance, i.e. match between query terms and document content
  - authoritativeness
  - personalization and contextualization

Overview



Web Crawlers

- Web crawler (or spider)
  - software that 'gathers' web pages, in order to index them
  - like a human who is browsing the web
    - repeatedly sends out requests for pages
- Crawlers have a list called the *URL frontier*
  - keeps track of URLs of pages that it has not yet indexed
  - initialise the URL frontier with a *seed set* of URLs



Crawlers: simplified

URL frontier = the seed set

while *URL frontier* is not empty

Remove highest priority URL from the *URL frontier*

Fetch the web page

Index the web page

Extract URLs that the web page links to

Add the URLs to the *URL frontier* in order of priority

Crawlers: a few niceties

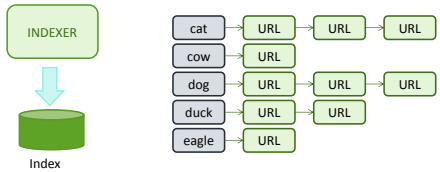
- Avoid too much duplication
  - check that a page hasn't been recently fetched and indexed
- Continuous crawling
  - because web pages change, URLs may be added back into the frontier so they get visited again
- Priority
  - based on change-rate and quality

Crawlers: a few niceties

- Being considerate
  - e.g. wait a few seconds before successive requests to a server
  - e.g. do not index pages mentioned in a site's robots.txt files
- Robustness
  - not falling into spider traps
- Efficiency, scalability
  - use multiple threads, processes, machines (but this complicates some of the other criteria)

Indexes

- The indexer builds an *inverted index*
  - a list of *terms* (usually sorted)
  - for each term, a *document list*, i.e. a list of URLs in which the term occurs



Index construction: simplified

- Given the URL and text of a document
  - Tokenize the text, i.e. break it into a list of tokens (e.g. words)
  - Linguistic pre-processing of tokens (e.g. stem them) to give terms
  - For each term,
    - if it's not among the list of terms
      - insert it into the list
    - if the URL is not in that term's document list
      - insert it into the list

Index construction: niceties

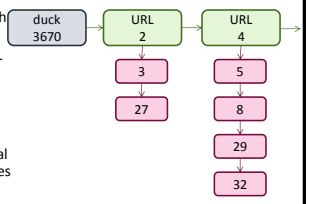
- Taking the text of a document
  - e.g. convert to a standard character encoding
  - e.g. strip off mark-up
- Tokenization
  - e.g. break on spaces (but sometimes you need to be cleverer, e.g. "New York")
  - e.g. throw away some but not all of the punctuation
    - e.g. what are the tokens for "O'Callaghan", "didn't", "C++" and words with hyphens such as "mis-represent"
  - e.g. possibly throw away stop words (very common words)
    - but this can cause problems, e.g. "To be or not to be", "The Who", "flights to Cork", "flights from Cork"

Index construction: niceties

- Linguistic pre-processing to increase matches
  - e.g. (for English) remove diacritics ("naïve" → "naive")
  - e.g. change to lowercase, perhaps
  - e.g. stemming, perhaps
- Efficiency
  - keeping the index small so as much as possible can be in main memory
  - organizing it for efficient update and querying
  - distributing its construction over multiple machines

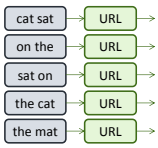
Index construction: niceties

- You might record *snippets*
- You might record frequencies
  - *document frequency* for each term
  - *term frequency* for each URL in a document list
- You could also build a positional index
  - records positions of terms within documents for phrasal queries and proximity queries
  - you could then also record markup



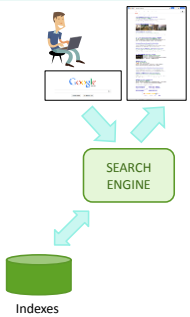
Index construction: niceties

- You could also build a *phrasal index*
  - e.g. a biword index would treat each adjacent pair of words as a term
  - but better might be to index phrases that look useful
    - e.g. each word is capitalized ("New York")
    - e.g. the phrase was within quotation marks or <i> tags…
    - e.g. crops up in queries



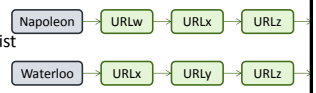
Search Engines

- Receive user's query
  - tokenize and pre-process as before
  - find user's query terms and their URLs in index(es)
  - rank the relevant URLs
- Send back results to user



Search Engines

- Consider a user who searched for "Napoleon Waterloo"
- Locate "Napoleon" in the index; gets its document list
- Locate "Waterloo" in the index; get its document
- Intersect the two lists
- Other things being equal, results in the intersection should be ranked higher than results not in the intersection



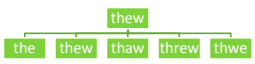
1. URLx
2. URLz
3. URLw
4. URLy

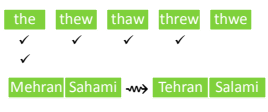
Advanced Queries

- Search engines often offer advanced search with
  - boolean operators: AND, OR, NOT
  - proximity search: X NEAR Y
  - phrasal search: in quotes
  - other, see Google Advanced Search
- Rarely used by the average user
  - except phrasal search: 10% of all queries?

Spelling and Typing Errors

- For each query word, create a set of candidate corrections
  - minor edits such as transpositions, insertions, deletions
  - including none
- Choose a candidate
  - dictionary-based
    - accept a candidate if it is in a dictionary and common
    - but person names, product names,…
  - corpus-based





Spelling and Typing Errors

- Corpus-based

w	c	$P(w\|c)$	$P(c)$	$10^9 P(w\|c)P(c)$
thew	the	.000007	.02	144.0
thew	thew	.95	.00000009	90.0
thew	thaw	.001	.0000007	0.7
thew	threw	.000008	.000004	0.03
thew	thwe	.000003	.00000004	0.0001

The Unreasonable Effectiveness of Data, Peter Norvig, http://www.youtube.com/watch?v=yvDCzhbjYWs (from about 27 mins)

Ranking

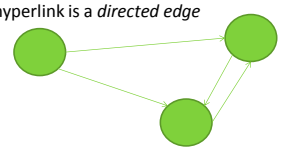
- For web search, the number of documents that match the terms in a query is typically very large
  - ranking the results is essential
- Ranking is based on a score that might combine, e.g.:
  - term frequency
  - inverse document frequency
  - the position or markup
    - e.g.if the query term matches document text within h1, strong, or em tags, the document might get a higher score
- The score may be a weighted combination:
  - weights may be specified by an 'expert'
  - but nowadays they are often learned automatically

Term Spam

- Add hidden terms to your page so that it is highly-ranked in searches for those terms
  - e.g. add multiple occurrences of "movie" to your shoe shop page
- Google's solutions
  - index a page by terms from pages that link to it, especially terms near to the links
  - PageRank: an 'importance' score for web pages based on the web graph

The Web Graph

- Web pages can be thought of as a *directed graph*
  - each page is a *node*
  - each hyperlink is a *directed edge*



  - links into a page are called *in-links*
  - the number of in-links a page has is its *in-degree*
  - similarly *out-links* and *out-degree*

Link Analysis

- Analysis of the web graph
  - used to compute PageRank
  - one factor in prioritising pages in the crawler's URL frontier
- Inspired by bibliometrics
  - analysis of citations to a scholar's publications
  - e.g. a researcher's H-index
- Analogy
  - a citation confers authority from one publication to another
  - a link in the web confers authority from one web page to another
- But not all citations/links are endorsements
  - e.g. an author citing his/her own works
  - e.g. all web pages in a site linking to the copyright notice

PageRank

- A web page's PageRank
  - a numerical score between 0 and 1
  - how often a random surfer would encounter that page, divided by the number of pages s/he visits
    - s/he visits some pages more often than others
    - intuitively, these are pages with many links coming in from other frequently visited pages
    - the idea of PageRank is these are the more important pages

PageRank: Random Surfing

- Start at some randomly-chosen page
- Repeatedly
  - decide (randomly) whether to
    - teleport to a new page (randomly chosen), or
    - follow one of the out-links (randomly-chosen), if any
  - move to the new page and add 1 to its count
- This has to repeat until a steady-state is reached
  - where the PageRanks don't change anymore

PageRank: Calculating it

- We don't need to build a random surfer
- Suppose there are n pages
  - use a vector $\boldsymbol{v}$ of n elements, one per page, containing the probability that a surfer is on that page
  - initially, each element in $\boldsymbol{v}$ is set to $1/n$

$$\begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix}$$

  - uses a matrix M of probabilities, the transition matrix
  - if page i has an out-link to page $j$and page i's out-degree is d, then $M_{ij} = 1/d$

$$\begin{pmatrix} 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1/1 \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/2 & 0 & 1/2 & 0 \end{pmatrix}$$

PageRank: Calculating it

- Calculate PageRank iteratively
  - $\boldsymbol{v}$ is the initial probability distribution
  - $M\boldsymbol{v}$ is the distribution after one step
  - $M(M\boldsymbol{v})$ is the distribution after two steps
  - …and so on
- Compute this until you find a $\boldsymbol{v}$ such that $\boldsymbol{v} = M\boldsymbol{v}$
  - for the web, about 50-75 iterations
  - this is the matrix's principle eigenvector
- This ignores teleportation, dead-ends and spider-traps (to simplify)

PageRank: Why Teleportation?

- Teleportation solves two problems
  - Dead-ends
    - a page with no out-links
    - without teleportation, in the limit any page that can take you to a dead-end will have PageRank of zero
  - Spider-traps
    - a group of pages with no dead-ends but their out-links do not link to pages outside the group
    - may be intentional (created by spammer) or unintentional
    - without teleportation, in the limit the PageRank calculation will place all the PageRank within the trap

Ranking, again

- Ranking is based on a score that might combine, e.g.:
  - term frequency
  - inverse document frequency
  - the position or markup
- But also
  - PageRank
- And a couple of hundred more *signals*

Ranking, again

http://www.searchenginejournal.com/infographic-googles-200-ranking-factors/

Link Spam

- Link spam is an attempt to manipulate PageRank
  - just like some 'scholars' create fake publications that cite their own work
- People build spam farms
  - fake pages with links to and from your shoe shop page
  - use terms such as "movies" near to these links

Link Spam

- But that's not good enough
  - the fake pages won't have high PageRank because they are not the destinations of other pages with high PageRank, hence your shoe shop page won't get high PageRank
- Must get some links from other people's real pages to your shoe shop page
  - to bring in some real PageRank
  - e.g. post comments on the real pages where the comments contain links back to your target page

Counteracting Link Spam

- Watch for and eliminate web pages that appear in structures that look like spam farms
  - a page that has links in from comments on other pages and links from other pages that don't do much
- Use TrustRank in place of PageRank
  - teleports only to trustworthy pages (e.g. from reliable domains)

Advertising

Display ads

- E.g.banner ads
- Typically a company pays
  - per impression (i.e. per download), or
  - per click on the ad

Search ads

- Alongside organic search results, there are sponsored search results

Search Ads

- You set a budget and place a bid with Google:
  - advert, query term, amount
- The user enters a query
  - any ad whose query term is part of the user's query might be shown depending on
    - relevance
    - budget remaining,
    - amount of bid
    - click rate so far
- If the user clicks on the ad, you pay Google the amount of the bid
- Click spam
  - Your competitor writes software to 'click' on your ads in order to exhaust your budget

NEW DIRECTIONS IN WEB SEARCH

Real Time Search

- How to include 'breaking stories' in results?
  - barely compatible with traditional web crawling
- Requires
  - frequent indexing of social networking sites, blogs, forums and news web sites
  - ranking based partly on recency and 'buzz'
- State of play
  - Bing has an agreement with Twitter
  - But Google's agreement ended mid-2011

Personalized Search

- Personalization
  - usually more signals involved in ranking
- E.g. Google
  - Web History
    - if you are logged-in, queries and clicks on results are recorded in your Web History (server-side)
    - frequently visited sites get 'bumped up' in search results
  - Google+
    - sites you follow in Google+ get 'bumped up'
- This may help query disambiguation
  - E.g. "jaguar" for a car fanatic vs. an Apple fan boy
- But it is controversial!

Knowledge Graph

- The biggest change
  - from keywords to entities

- Google's Knowledge Graph
  - A semantic network of entities and relationships, capturing 'facts'
  - 600 million entities; 18 billion facts; and growing
  - Compiled from Freebase, CIA World Factbook, Wikipedia

Knowledge Graph

- Search results can include
  - knowledge panels
  - carousels
- Works well for
  - actors, directors, movies, artists, art works, museums, cities, countries, rivers, planets, pieces of music, bands, sports teams,…

Knowledge Graph

- More important is the *answer box*
  - tries to answer a question
- Try these:
  - Charles Dickens wife
  - Where is Dickens buried
  - How tall is Tom Cruise
  - Rooney's shirt number
  - United scores
  - Who built St Paul's
  - Population of Cork
  - Cork weather
  - Ireland dialling code
  - How old is the Earth
  - 70 Fahrenheit to Celsius
  - Euros to dollars
  - sin(x)
  - Calories in ham and cheese sandwich
  - Google stock
  - What is the answer to life, the universe, and everything

Answering Questions

- How does Google do it?
  - some natural language processing
  - the knowledge graph
  - some specialised web services
- Wolfram Alpha is similar
  - more emphasis on specialised reasoning for calculations and computations
- This moves web search
  - from finding all relevant results
    - many results
  - to answering your question
    - one primary result

What's Next for Search?

- Better natural language understanding
- Better speech recognition
  - e.g. Google Voice Search, Apple Siri
- More conversational features
  - e.g. with Google Voice Search try
    - How tall is Tom Cruise
    - Hold old is he
- Increasing contextualization
  - e.g. Google Now

But…

- With *search*, you still have to formulate a need
  - a sense of what you're after
- The focus in the rest of CS6120 is *discovery*