BITS Pilani
K K Birla Goa Campus

innovate achieve lead

08- Web Search Basics

Information Retrieval

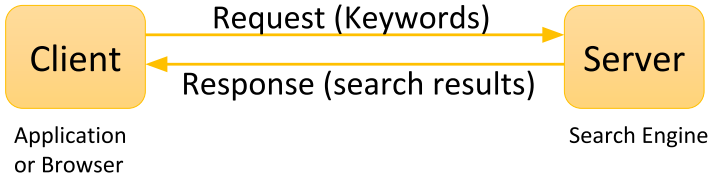
Topics

1. Web Search
2. Web characteristics
   i. Web graph
   ii. Spam
3. The search user experience
4. Index size and estimation
5. Spam page detection
6. Duplicate page detection

1. Web Search (non-technical)

- Web is unprecedented in many ways:
  - scale
  - lack of coordination in creation
  - motives of participants
  - diversity in backgrounds
- Makes web search different and generally harder.

Basic Structure



- http://www.abc.com/path/page.html

Protocol | Domain | Directory | HTML page consisting of information

- Hyperlinks, text- all available in HTML format

Web Search

- Designers of first browsers made it easy to view HTML tags on the content of a URL
- Allowed new users to learn from those example tags and create their own content
- Rapidly led to the formation of numerous incompatible dialects of HTML>> browsers ignoring what they don't understand
- **Promotion**: allowing new developers (and not only the trained programmers) to learn from examples and create their own web pages>> eventually leading to not a few hundreds but millions of individuals

Web Search

- Mass publishing of web pages and information is useless unless no information is being discovered or consumed by other users
- Early attempts in making information discoverable:
  - Full-text index search engines
    - Altavista, Excite, Infoseek
  - Taxonomy populated with web pages in categories
    - Yahoo

Web Search

- Full-text Index Search
  - A keyword based search interface available to the users
  - Mechanism: inverted indexes and ranking
- Taxonomy-based Search
  - Browse through a hierarchy tree of category labels
  - Convenient and intuitive way to find web pages

Web Search

- Drawbacks of Taxonomy based Search:
  - correctly classifying web pages into taxonomies
  - manual editorial process
  - Practically overwhelming to scale the size of web
  - User's idea of a web page (sub-tree) to search for a topic should match with that of the editor
- During initial stage itself, Yahoo's taxonomy surpassed 1000 distinct nodes and eventually stopped using the method
- About.com and Open Directory Project- experts collect web pages and annotate them with categories

2. Web Characteristics

- Decentralized content publishing
  - No central control of authorship
  - Essential Feature that led to explosive growth of the web
  - Also, turned out to be the biggest challenge for web search engines (indexing and retrieval)
- Dozens of natural languages- demanding many different forms of stemming and other linguistic operations

Web Characteristics

- Publication of web pages is no more limited to editorially trained writer
  - Tremendous democratization of content creation
  - Variation in grammar and style
  - Leading to best and worst publication
  - Variation in colors, fonts, structure
  - Consisting of clickable images- redirecting you to richer textual content
    - HOW TO INDEX SUCH PAGES WHEN NO TEXT IS AVAILABLE?

Web Characteristics

- Democratization of content creation:
  - New level of granularity in opinion on any topic
  - Web contained truths, lies, contradictions, and suppositions (hypotheses, beliefs)

Which web page should one trust?

Web Characteristics

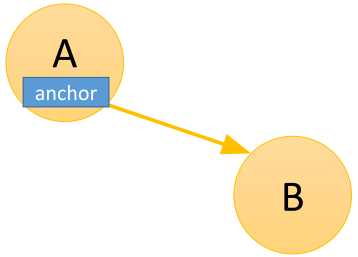
- Some publishers are trustworthy and others are not
- How is a search engine to assign such a **measure of trust** to each website? (later modules)
- No defined notion of trust
  - One website might be trustworthy to one user but may not be so to other
  - In non-web publishing, it's user self selection of sources they find trustworthy- *The New York Times* or *The Wall Street Journal*
  - This challenge is significant when search engines are the viable means for a user

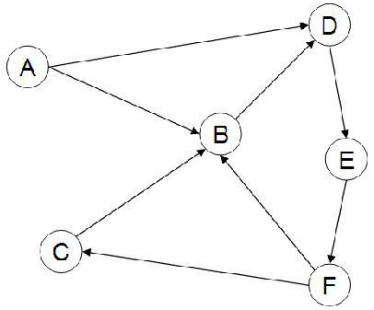
Web Characteristics

- "How big is the Web?"- No easy answer
- "How many pages are in a search engine's index"- more precise
- By the end of 1995, Altavista reported having 30 million *static web pages* crawled and indexed in it's database

2.1 Web Graph

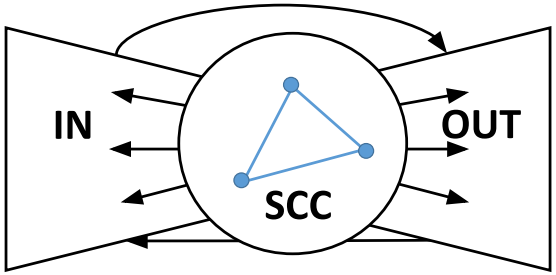
- **Anchor** text- encapsulated in href attribute
- In- Link and Out-Link
- **Strongly Connected Components** (SCC)- the user can surf from any page in SCC to any page in SCC





There is a path from A to F through D and E But no path from F to A

Similarly, there is a path from A to C but not from C to A



2.2 Spam

- Web search engines- an important means for connecting advertisers to the buyers
- Searching for a "golf real estate" is not merely for seeking information but the user might be interested in buying the property
- Sellers of such properties or agents would have high incentive to create pages that rank highly in the search results
- If the ranking is based on term frequency, then a web page with numerous repetition of term "golf real estate" would rank high

First generation of Spam
(in the context of web search)

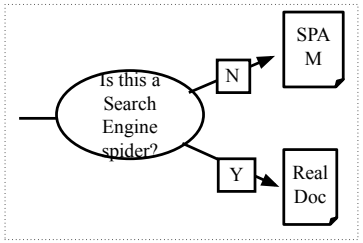
Manipulation of web pages content for the purpose of appearing high up in the search results for selected keywords

Spam

- Heterogeneity in the motives of spamming
  - Commercial purpose- an economically motivated activity
- In many search engines, it is possible to pay to have one's web page included in the search engines' index>> "PAID INCLUSION"
- Search Engine Optimization
  - Tuning your web page to rank highly in the algorithmic search results for selected keywords
  - Alternative to paying for placement
  - Thus, intrinsically a marketing function
  - Performed by companies, webmasters and consultants ("Search engine optimizers") for their clients

Spamming Techniques

1. **Cloaking**
   - serve fake content to the search engine
   - For human user, altogether different content than that indexed by the search engine



Spamming Techniques

2. **Doorway pages**
   - Text and metadata carefully chosen to rank highly on selected search results
   - When browser requests to doorway page, it is redirected to a page consisting of more commercial content
3. **Link spamming**
   - Manipulation of metadata related to a page including links
   - Domain flooding: numerous domains that point or re-direct to a target page

3. The Search User Experience

- crucial to understand the *users* of web search as well
- **Traditional IR**:
  - users were typically professionals with some training in the art of phrasing queries over a well-authored collection of documents
  - Understood the structure and style of collection

The Search User Experience

- **Web Search Users**
  - Not knowing the heterogeneity of web content, syntax of query language, art of phrasing queries
  - A mainstream tool like a search engine should not place such demand on the billions of people
  - Research studies conclude that an average number of keywords in a web search query is somewhere between 2 or 3

The Search User Experience

- More user traffic on a search engine>> more revenue it stands to earn from sponsored search
- How do search engines grow their traffic?
- Google identified two principles:
  - Focus on relevance, specifically, precision rather than recall in first few results
  - A light weighted user experience

The Search User Experience

- **Focus on relevance:**
  - Save users' time in locating the information they need
- **User Experience:**
  - Search query page and search results page is uncluttered
  - Almost entirely textual and with very few graphical elements

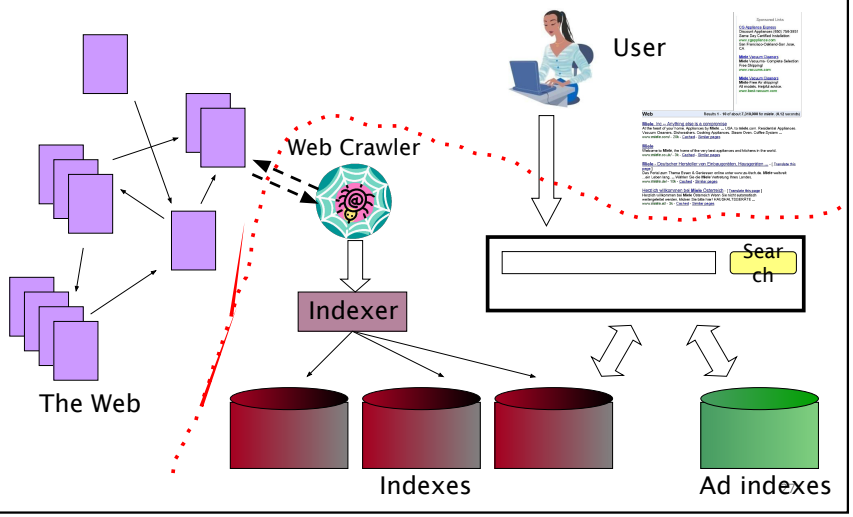
3.1 User Query Needs

- **Informational**
  - General information on a broad topic.
    - Country name, a research area
  - Not one single web page that contains all the information
  - Typically try to get information from multiple web pages
- **Navigational**
  - Website or homepage of a single entity
    - An airline company- AirIndia
    - The very first search result should be the homepage of AirIndia
- **Transactional**
  - User performing a transaction on the web
    - Buying a product or service, making a reservation, bill payment
  - Search engine should return results listing services that provide interfaces for such transactions.

User Query Needs

- Important to identify which category the query belong to
  - The category not only governs the algorithmic search results but also the suitability for sponsored search results (Is there an intent to purchase)
- Arguments
  - One search result for navigational queries?
  - Re-directed to the official or target website of entity?
  - Users don't care
  - However, for informational queries, users does care about the comprehensiveness of search engine.
  - Search engines do have to pay attention to how their index size compare to other search engines

The Search User Experience



4. Index size and estimation

Index Size and Estimation

- Comprehensiveness of the search results grows with index size
  - Though it also matters which pages a search engine indexes
  - Some pages are more informative than others
- Difficult to claim the fraction of web indexed by search engine
  - Infinite number of dynamic pages

Index Size and Estimation

- http://www.yahoo.com/any_string will return a valid HTML page rather than error
  - "there is no such page."
- One website can generate infinite number of such valid pages
  - Some of them are malicious spider traps

Index Size and Estimation

- ~~What is the size of the web?~~
- ~~What is the size of the search engine index~~
- a better defined question:
  - Given two search engines what are the relative sizes of their indexes?
- Again, is it precise? No.
  1. In response to a query, a search engine can result a webpage whose content it has not indexed (fully or partially)
     1. Indexes only a first few thousands words
  2. Indexes are organized in tiers and partitions.
     1. Website search or global search

Index Size and Estimation

- Search engine indexes include multiple classes of indexed pages, so no single measure of index size.
- Several techniques to determine the ratio of indexes of two search engines
  - **Underlying hypothesis**
    - Each search engine indexes a fraction of web chosen independently and uniformly at random.
  - **Assumptions**
    - There is a finite set of web from which a search engine chooses a subset
    - Each search engine indexes an independent uniformly chosen subset

4.1 Capture- Recapture Method

- Relative size from overlap- given two engines A and B
- Pick a random page from the index of A and test whether it is present in the index of B and similarly, pick a random page from B is in A.
- Result: a fraction x of engine A and a fraction y of B that overlap.

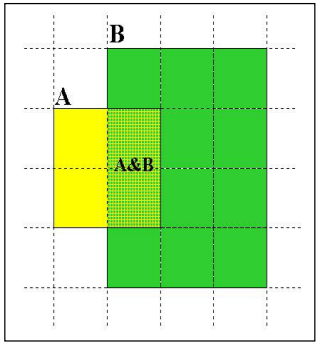
Capture- Recapture Method

- $A \cap B = \frac{2}{4} * |A|$

  $A \cap B = \frac{2}{12} * |B|$

  $\frac{1}{2}|A| = \frac{1}{6}|B|$

  $\frac{|A|}{|B|} = \frac{1}{3}$



Capture- Recapture Method

- In order to get an unbiased estimate of $\frac{|A|}{|B|}$:
  - Previous assumptions must be true
  - Sampling process is unbiased
- Two possible Scenarios:
  - Measurement is performed by someone with access to search engine indexes- Simple
  - Independent party sending queries from outside the search engines- Challenging
    - Generating a random page on the web and test it's presence in the indexes of both A and B search engines
    - Picking a uniformly at random page on web is a difficult problem

4.2 Sampling techniques

4.2.1 Random Searches

- Search log of web searches
- Send a random search from the log to *A* and select a page from the search results
- Hard to find web searches log
  - Trap all queries going out from of a work group
  - Limitation: work group bias queries

4.2.2 Random IP Addresses

- Generate random IP addresses
- Send requests to the web server residing at that random IP address
- Collecting all pages from that web server
- Choose a page at random
- Challenges:
  - Multiple hosts can share the same IP address (virtual hosting)
  - Webserver not accepting requests from the host
  - Domain flooding
  - Might hit a website with a very few pages
    - Can be fixed only be knowing the distribution of pages on the website

4.2.3 Random Walk

- View the web as a strongly connected directed graph
- Take an arbitrary web page and start a random walk
  - Includes various "jump" rules back to visited sites
  - Does not get stuck in spider traps!
  - Must consider that the graph is finite
- **Challenges**:
  - Web is not strongly connected
  - List of seed web pages is a problem
  - Time to convergence not really known

4.2.4 Random Queries

- Successfully built upon for a series of estimates
- Carelessly implemented leading to misleading measurements

- Generate a random query: How?
  - Take a lexicon of words (extracted from web)
  - Take some words from the lexicon and perform conjunction (w1 AND w2)
  - Get top *k* results from search engine *E1*
  - Choose a random url *U* as a candidate
  - Search "w1 AND w2" on search engine *E2*
  - Check if *U* is present in the search results

4.2.4 Random Queries

- Strong Query to check whether an engine B has a document D:
  - Download D. Get list of words.
  - Use 8 low frequency words as AND query to B
  - Check if D is present in result set.
- Problems:
  - Biased towards content rich documents
  - Ranking bias as taking top k results
  - Engine time-outs
  - Is 8-word query good enough?
  - Pages omitted

Summary

- No sampling solution is perfect.
- Lots of new ideas ...
- ....but the problem is getting harder
- Quantitative studies are fascinating and a good research problem

5. Near duplicate detection

Duplicate Documents

- The web is full of duplicated content
- Strict duplicate detection = exact match
  - Not as common
  - Two strict duplicate documents have the same *fingerprint*
- But many, many cases of near duplicates
  - E.g., last-modified date the only difference between two copies of a page
  - Comparing all pairs of web pages is exhaustive and infeasible

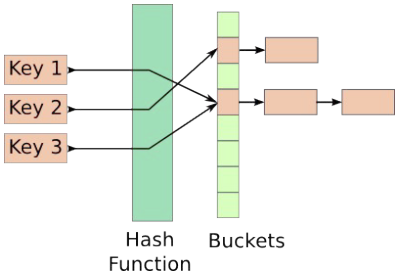
5.1 Shingling Method

- Shingles: Word n-grams
- Given a positive integer *k* and a sequence of terms in a document *d*, define the k-shingles of *d* to be a set of all consecutive sequences of *k* terms in *d*
- **Text: a rose is a rose is a rose (k=4)**
  - a_rose_is_a
  - rose_is_a_rose
  - is_a_rose_is
  - a_rose_is_a
  - rose_is_a_rose
- Similarity measure (degree of overlap) between two documents (document is now represented as a set of shingles)
- Jaccard Coefficient between Sets $S(d_1)$ and $S(d_2)$:
  - $J(S(d_1), S(d_2)) = \frac{|S(d_1) \cap S(d_2)|}{|S(d_1) \cup S(d_2)|}$

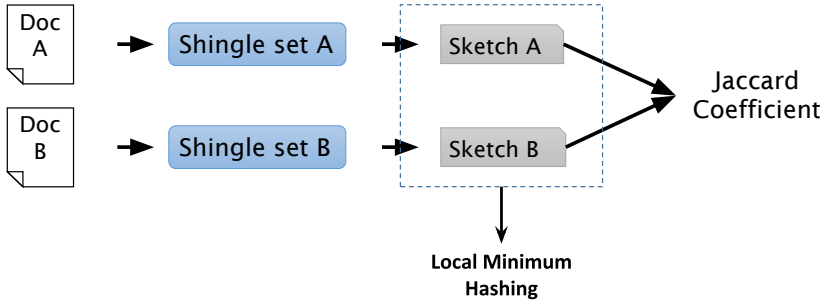
Score> threshold => near duplicates
need to compute Jaccard coefficient pairwise

Hashing Method

- For a given document d_j, create a set of shingles $S(d_j)$, $|s|=k$, $s \in S(d_j)$
- Apply a hash function H on $S(d_j)$ and store the result in a Hash Table.
- If shingles of two documents share the same key then they seem to be duplicate



Efficient Similarity Computing



From Sets to Boolean Matrix

- Rows= elements of the universal sets
  - Example: sets of all k-shingles
- Columns= sets
- If and if only if element e_i is a part of set S_j then $a_{ij}=1$
- Column similarity is the Jaccard similarity of the sets of their rows with 1
- Typical matrix is sparse (most of the values are 0)

Example- Column Similarity

C1	C2	
0	1	*
1	0	*
1	1	*
0	0	
1	1	*
0	1	*

Jaccard similarity
Sim (C1, C2)= 2/5= 0.4

Four types of Rows

- Our goal is to describe how min hashing of sets works and how it can be used to compute the similarity by looking at the signatures of the columns
- Given two sets or matrix columns C1 and C2, rows may be classified as>>

	C1	C2
A	1	1
B	1	0
C	0	1
D	0	0

A= # rows of type A

Sim (C1, C2)= $\frac{|A|}{|A|+|B|+|C|}$

Minhashing

- Each minhashing hash function is associated with the permutation of rows with the matrix (Imagine the rows are permuted)
- Define Minhashing function h(C)= index of the first row (in permuted order) in which column c has 1
- Use several hash functions (~100) to create a signature for each column
- For each column the signature is the sequence of row numbers we get after applying the minhashing function on the column
- We select only one hash function for the matrix and apply the same on each column.
- Signatures can be displayed in another matrix where column represents sets and rows represent minimum hash value for that column

Minhashing Example

1	4	3	1	0	1	0
3	2	4	1	0	0	1
7	1	7	0	1	0	1
6	3	6	0	1	0	1
2	6	1	0	1	0	1
5	7	2	1	0	1	0
4	5	5	1	0	1	0

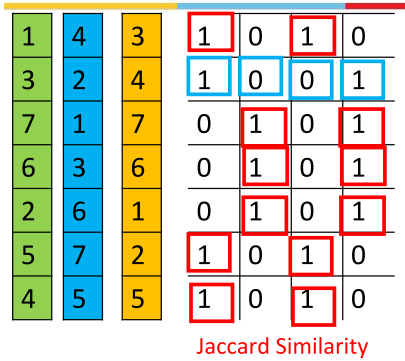
Signature Matrix M

2	1	2	1
2	1	4	1
1	2	1	2

Minhashing Property

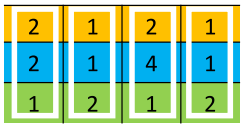
- The probability (over all permutations of the rows) that h(c1)=h(c2) is the same as Jaccard sim(c1, c2)
- why?
- look down the permuted columns c1 and c2 until we see a 1
- if it's a type *A* row, then h(c1)=h(c2). if type b or type c then not
- the probability that two columns have same min hash value is the probability that the first row that isn't a type d is a type a row

Minhashing Example



Fraction of components in which two signatures agree

Signature Matrix M



	1-3	2-4	1-2
Col/Col	3/4= 0.75	3/4= 0.75	0
Sig/Sig	2/3= 0.67	3/3= 1.00	0

Implementation of Minhashing

- Minhashing is done once we actually permuted the rows but not very feasible to do so
- 1 billion rows
- Hard to pick a random permutation of 1 billion rows
- Representing a permutation will require 1 billion entries (4GB)
- 100 permutation will cost 0.4 TB
- Many disk access are required to access the rows from permuted order

Implementation

- Simulating permutation without doing permuting rows
- For each main hash function select a normal hash function
- We pretend that the order of row r in permutation is the h(r)
- So, for each column, we look for that row r for which the column has a 1 and for which h(r) is minimum

Implementation

- For more approximation, pick a large number of hash functions (~100)
- We simulate for each hash function
- For each column c and each hash function h_i, keep a slot M(i, c)
- For 100 hash functions size of slot is 100*#columns
- **Intent**: M(i, c) will be the smallest value of $h_i(r)$ for which column has a value 1 in row r

Implementation

For each row r **do begin**
  for each hash function h_i **do**
    compute h_i (r);
  for each column c
    if c has 1 in row r
      for each hash function h_i **do**
        if h_i (r) < M(i, c) **then**
          M(i, c):= h_i (r);
end

Example

Row	C1	C2
1	1	0
2	0	1
3	1	1
4	1	0
5	0	1

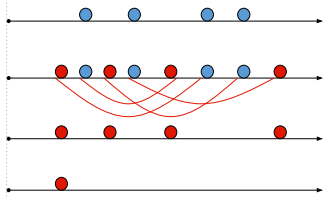
h(x)= x mod 5
g(x)= 2x+1 mod 5

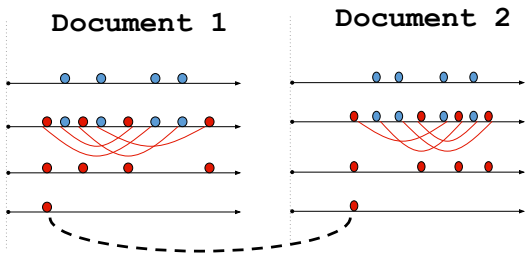
Jaccard Similarity should be closer to 0

Column wise similarity =1/5= 0.2

H(r)	Value	Sig 1	Sig 2
		∞	∞
h(1)	1	1	∞
g(1)	3	3	∞
h(2)	2	1	2
g(2)	0	3	0
h(3)	3	1	2
g(3)	2	2	0
h(4)	4	1	2
g(4)	4	2	0
h(5)	0	1	0
g(5)	1	2	0

Document 1





Sketch of a Document

1. For a given document d_j,
2. Create a set of all shingles $S(d_j)$, $|s|=k$, $s \in S(d_j)$
3. Apply a hash function H (64 bit) on shingles $S(d_j)$ where each $h \in H(d_j)$ represents a hash of $s \in S(d_j)$
4. Apply permutation $\prod$ on hash $H(d_j)$ where each $\pi(h) \in \prod(d_j)$
5. Select the minimum $\pi_i \in \prod(d_j)$
6. **Repeat** steps from 3 to 5 for a large number of hash (~200 unique hash functions)
7. If $\pi_i \in \prod(d_{j1}) = \pi_i \in \prod(d_{j2})$, then the documents are similar