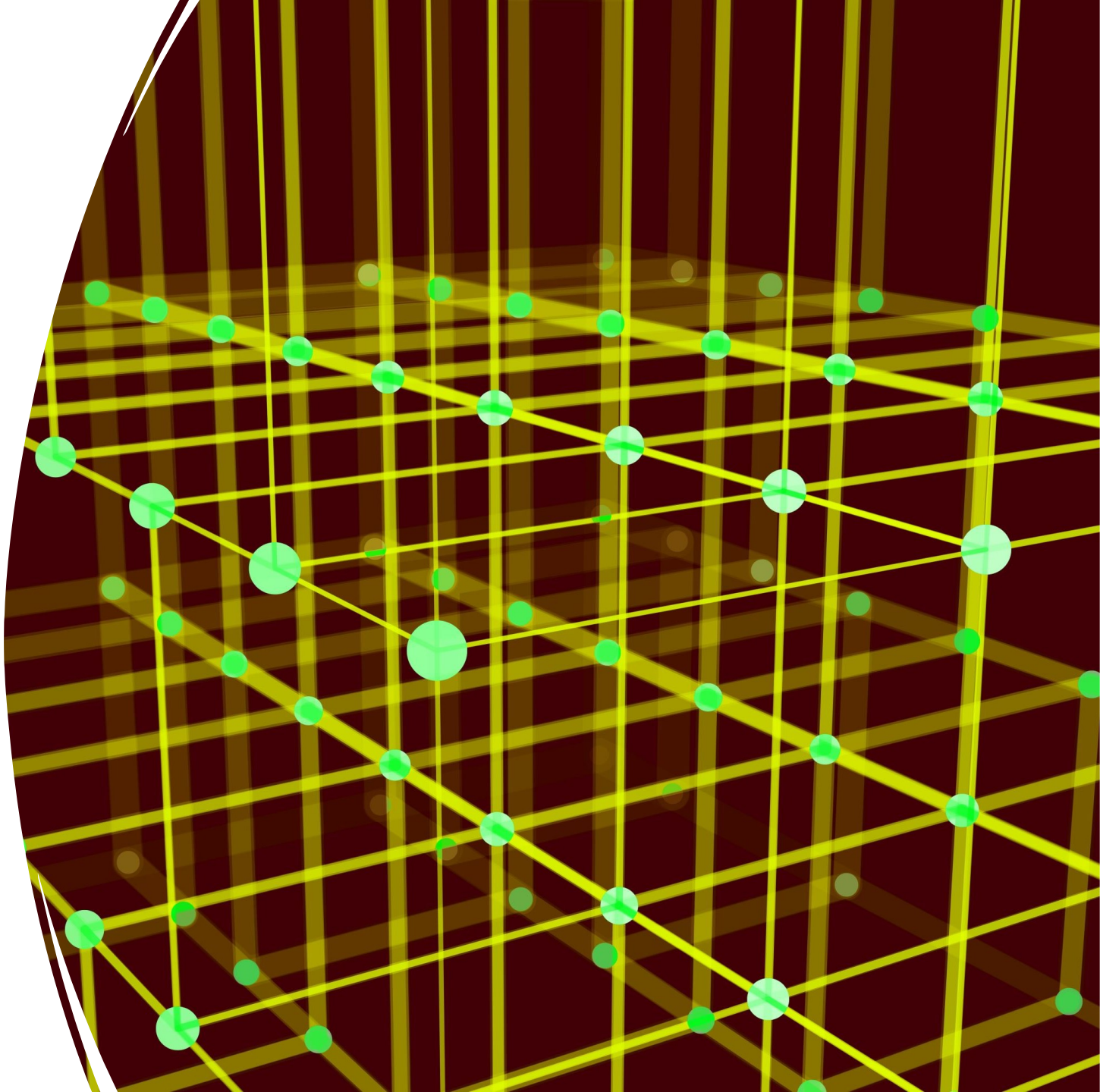
Link Analysis

Ravi Kumar Gupta

https://kravigupta.in



WWW, Search Engines and Spammers

- WWW is a groundbreaking innovation of this century.
- Web users and content creators are diverse with varied motivations.
- Searching the Web became challenging
- Rise of Web Search Engine addressed this issue
- Spammers started manipulating the search engine results
- Search engines continuously evolve to counteract spam.
- Search engines use a technique – Link Analysis – to avoid spam
- Link Analysis – Analysis of hyperlinks and the graph structure of the Web for ranking of web search results.
- One of the many factors considered by Search Engines

WWW, Search Engines and Spammers

- Google introduced the "PageRank" measure, revolutionizing web page ranking.
- PageRank evaluates web pages based on user queries.
- Spammers found ways to manipulate PageRank, leading to "Link Spam."
- "TrustRank" was developed to detect and counteract Link Spam.
- Multiple variants of PageRank have been developed for refined web page evaluation.

Search Engines

Early search engines were of two types:

- Full-text index search engines
  - (e.g., AltaVista, Lycos) with keyword search interfaces.
  - Indexing the growing Web was challenging.
- Taxonomy-based search engines
  - (e.g., Yahoo!) organized Web pages hierarchically by categories.
  - Accurate classification was challenging due to Web's size and growth.

Full-Text Index Search Engines

- Search engines that scan and index the complete textual content of web pages. Users input keywords, and the engine returns pages where those keywords appear.

Key Features:

- **Automated Crawling:** Uses bots or spiders to continuously crawl the web and update the index.
- **Keyword-Based:** Users search by entering keywords or phrases.
- **Relevance Ranking:** Algorithms rank pages based on relevance to the search query, considering factors like keyword frequency, backlinks, and site authority.
- **Dynamic Updates:** Constantly updates its index to include new pages and reflect changes to existing ones.

Examples: AltaVista, Lycos

Taxonomy-Based Search Engines

- Search engines that organize web content into a hierarchical structure based on predefined categories or topics.

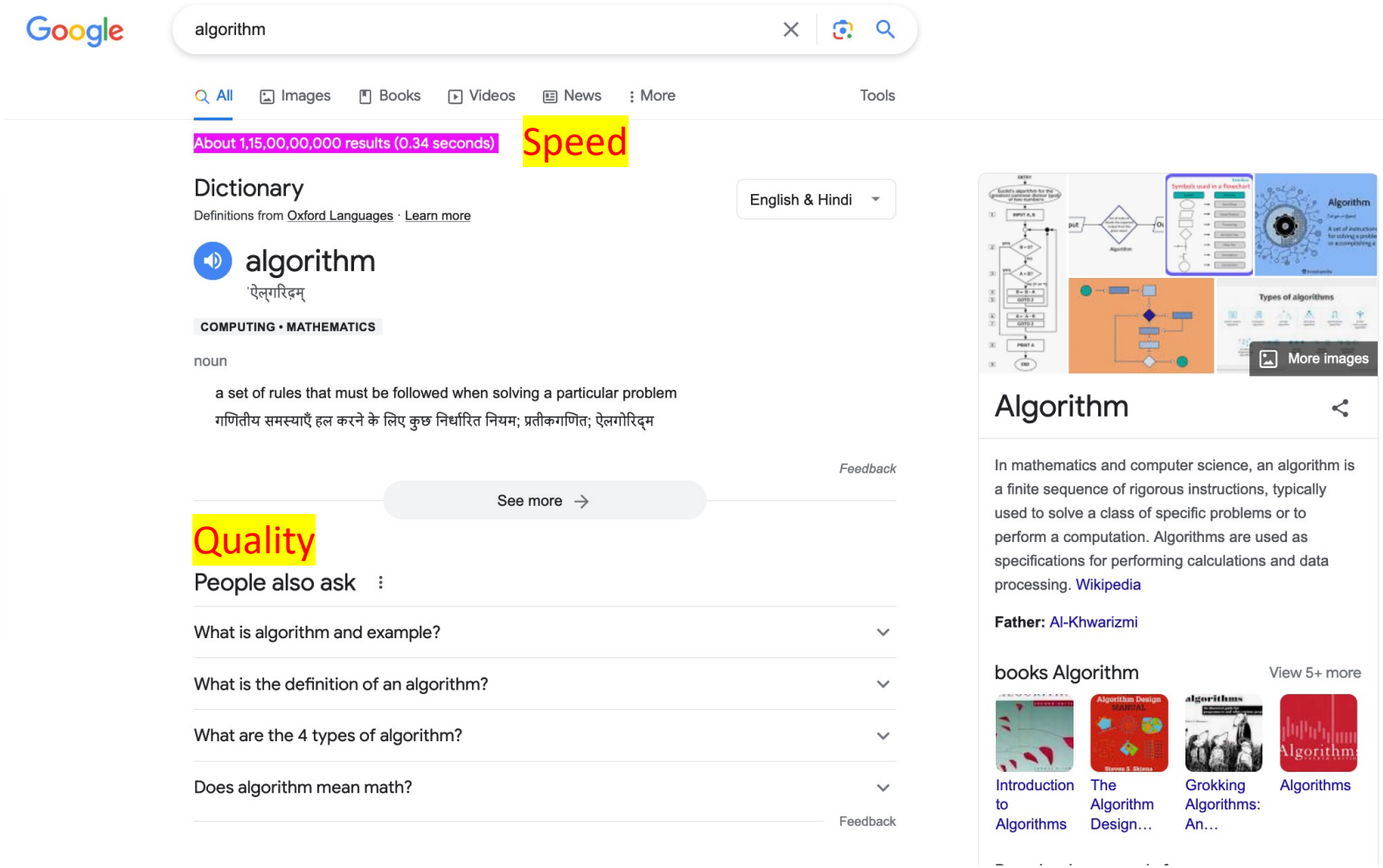
Key Features:

- **Hierarchical Structure:** Content is organized in a tree-like structure with broad categories and nested subcategories.
- **Manual Curation:** Human editors often decide the categorization of web pages.
- **User Navigation:** Users navigate through categories and subcategories instead of relying solely on keyword searches.
- **Clear Organization:** Provides an organized view of web content by topic or category.

Examples: Yahoo! Directory, Open Directory Project (DMOZ)

History

- Web became more about
  - E-selling platforms
  - Opinion forming sites
  - Information dissemination
- Search engines
  - Connect users to relevant information.
  - Prioritize speed and quality of search results.
- Web owners motive –
  - Strong desire to rank high in search results.
  - Attract more visitors and gain visibility.



History contd..

- All this led to first generation of Spam

Spam : Manipulation of web page content to rank higher in search results for specific keywords.

- First generation of spam targeted early search engines.
- Search engines developed techniques to detect and counteract spam.
- Spammers evolved, introducing more sophisticated spamming techniques.

Spamdexing and SEO

Spamdexing

- Practice of search engine spamming
- Combination of "Spamming" and "Indexing".
- Aimed at illegitimately achieving high search engine rankings.

Search Engine Optimization (SEO):

- Industry focused on optimizing websites for search engines.
- Goal: Improve website ranking legitimately.

Misuse of SEO:

- Some SEO providers resort to spamdexing.
- Aim: Achieve high rankings through illegitimate means.

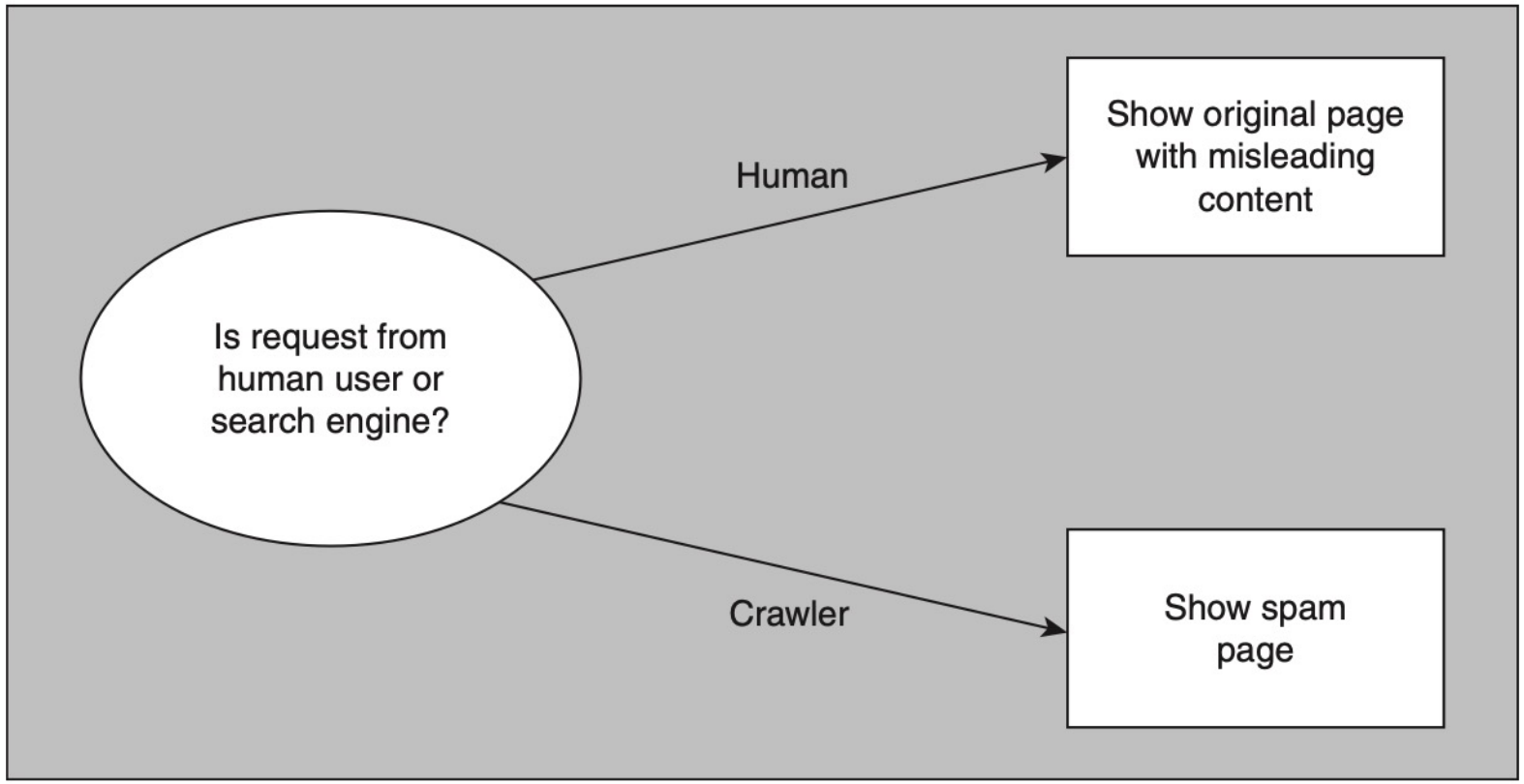
Techniques of Spamdexing

- There are many techniques of Spamdexing

Two popular are -

- Cloaking
- Doorways pages

Cloaking


Is request from human user or search engine?
Human
Crawler
Show original page with misleading content
Show spam page

Cloaking

- **Definition:**
  - Cloaking is the practice of presenting different content or URLs to human users and search engine crawlers.
- **Purpose:**
  - Deceive search engines to achieve higher rankings.
  - Get indexed under specific, often unrelated, terms.
- **How It Works:**
  - Server identifies the user-agent (browser or bot).
  - Human users see the regular page.
  - Search engine bots see an optimized version.

Cloaking contd..

- **Misleading Indexing:**
  - Users might see pages listed for unrelated search terms.
- **Targeting Multiple Search Engines:**
  - Different versions for each search engine's ranking algorithm.
- **Ethical Implications:**
  - Considered a "black hat" SEO tactic.
  - Violates most search engines' Webmaster Guidelines.
- **Risks:**
  - Websites can face penalties, including de-ranking or removal.

Cloaking - Example

- **Scenario:**
  - A website selling writing instruments.
- **For Search Engines:**
  - Page displays text about the history of writing instruments, paper, etc.
  - Ensures high ranking for these topics.
- **For Human Users:**
  - Page displays images or Flash ads of pens, pencils, and other writing tools.
  - Visually appealing but lacks the textual content shown to search engines.

Cloaking Example

- **Objective:**
  - Boost ranking in unrelated topics.
- **Technique:**
  - Keyword stuffing.
- **For Search Engines:**
  - Page contains the term "music" repeated thousands of times.
  - Elevates the page's ranking for music-related searches.
- **For Human Users:**
  - The repeated term might be hidden or not visible.
  - Users are misled to visit a site unrelated to their search.

Doorway Pages

- **Definition:**
  - Low-quality pages with minimal content, stuffed with similar keywords/phrases.
  - Designed to rank high in search results but offer no real value to visitors.
- **Functionality:**
  - When accessed, they redirect users to a more commercial page.
  - Can be generated in bulk using software for specific keywords.
- **Appearance:**
  - Often visually unappealing and wouldn't pass human scrutiny.

Doorway Pages - Example

- **Scenario:** A company sells handmade leather shoes and wants to target customers in various cities.
- **Doorway Page Creation:**
  - Separate pages are created for each city:
    - "Handmade Leather Shoes in New York"
    - "Handmade Leather Shoes in Los Angeles"
    - "Handmade Leather Shoes in Chicago"
  - Each page has nearly identical content but swaps out the city's name.
- **Purpose:** When users search for leather shoes in their city, they land on the respective doorway page, which then redirects them to the main website.

Doorway Pages - Example

- **Scenario:** An online electronics store sells various brands of headphones.
- **Doorway Page Creation:**
  - Separate pages are made for each brand or type:
    - "Best Sony Headphones"
    - "Top-rated Bose Headphones"
    - "Affordable JBL Headphones"
  - Each page might have sparse content but is filled with keywords related to the brand.
- **Purpose:** To capture users searching for specific headphone brands and then redirect them to the main product listing or another related page.

Term Spam

Manipulation of web page text to deceive search engines is termed "Term Spam".

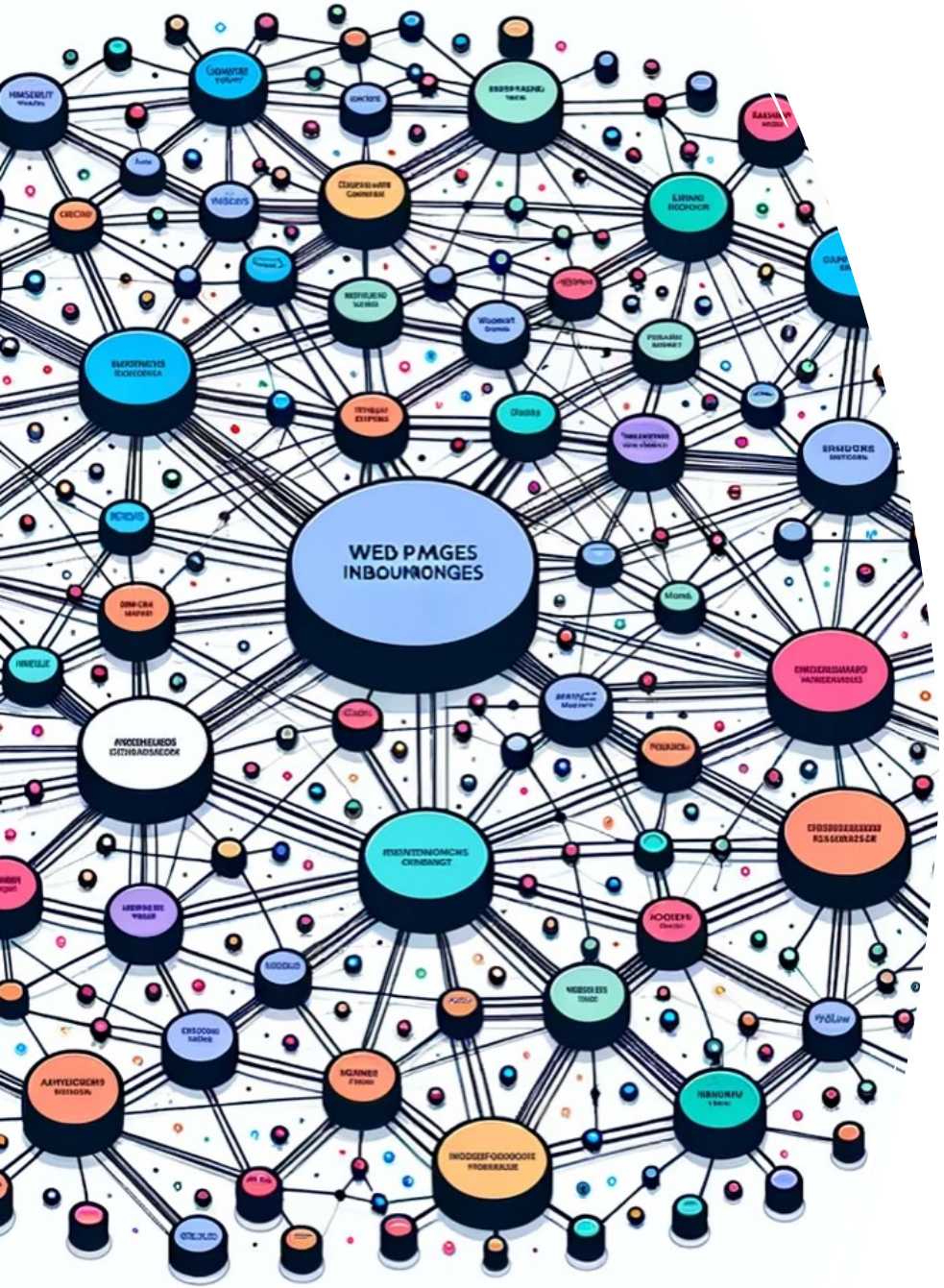
Techniques

- **Meta-Tag Stuffing:**
  - *Overloading meta-tags with keywords to manipulate rankings.*
- **Scraper Sites:**
  - *Websites that copy content from other sites to attract more traffic.*
- **Article Spinning:**
  - *Rewriting existing articles to appear as unique content.*

Search Engines vs Spammers

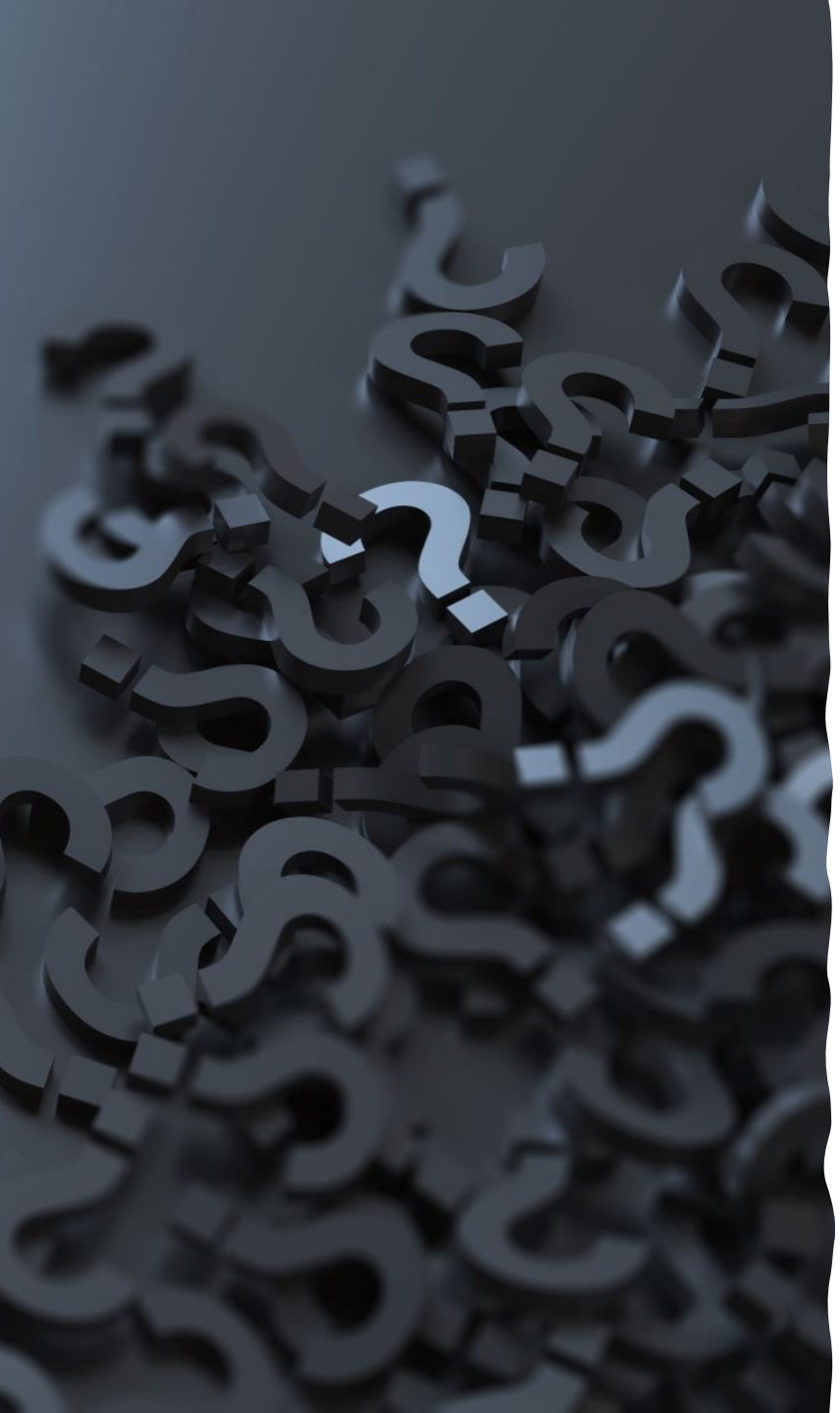
- **Link Analysis:**
  - *Exploiting the link structure of the web to rank pages.*
  - *First majorly implemented by Google.*
- **Link Spam:**
  - *Spammers' response to link analysis.*
  - *Efforts to manipulate the link structure of the web.*
- **Ongoing Battle:**
  - *The war between search engines and spammers continues as techniques evolve on both sides.*

PageRank



Link-Based Analysis

- Web can be treated as a giant graph.
- Web pages are nodes; hyperlinks are edges.
- Number of inbound links indicates a page's importance.
- Aims to combat term spam and improve search results.



Google’s innovations

- **Random Surfer Model**
  - Imagine a surfer randomly navigating the web.
  - Pages visited more often are deemed more important.
  - Basis for the PageRank algorithm.
- **Link Context Analysis**
  - Ranking considers terms near links to a page.
  - Counters term spam as altering external links is challenging.

Random Surfer Model

- Concept based on a user randomly navigating the web.
- At each step, the surfer moves to a randomly chosen linked page.
- Pages visited more frequently are deemed more important.
- Forms the foundational idea behind Google's PageRank algorithm.

- *The idea of PageRank is that pages with large number of visits are more important than those with few visits.*

Example

- Think of a mall with many stores.
- If most people end up visiting a particular store more often, it's probably an important or popular store.
- Similarly, websites that get more "visits" or "clicks" are seen as important in the web world.

Link Context Analysis

- Analyzes terms appearing on or near links to a page.
- Helps in determining the context and relevance of the linked page.
- Aims to counter term spam by considering external link context.
- Makes spamming harder as altering terms on external links is challenging.

- Example
- Imagine a blog post about healthy foods that links to an article on "avocado benefits."
- If the words around the link mention "nutritious," "healthy fats," or "vitamins," it gives search engines a clue
- Clue that the linked article is indeed about the health benefits of avocados.

PageRank Algorithm

- Considers both number and quality of links.
- Spammers can't just increase in-links with low-quality pages.

Google’s own words -

- *PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the Website is.*
- *The underlying assumption is that more important Websites are likely to receive more links from other Websites.*

PageRank Definition

- Definition:
  - PageRank is a system developed by Google to rank web pages in their search engine results. It's a way of measuring the importance of website pages.
- How it Works:
  - Every time a webpage links to another page, it's casting a vote for that page. The more votes a page gets, the more important it is.
  - However, not all votes are equal. Votes from a high PageRank page mean more than votes from low PageRank pages. So, links from important pages have more weight.

Illustration as Web Graph Model

- Web Graph: Think of the internet as a giant web, where each page is a node, and the links between them are arcs.
- Example: There are five pages (1 to 5). Some pages link to others, creating a web of connections.
- Backlinks & Outlinks:
  - Backlinks (or inlinks) are incoming links to a page.
  - Outlinks are outgoing links from a page.
  - For instance, page 5 has links
    - coming from other pages (backlinks)
    - also links it sends out to other pages (outlinks).

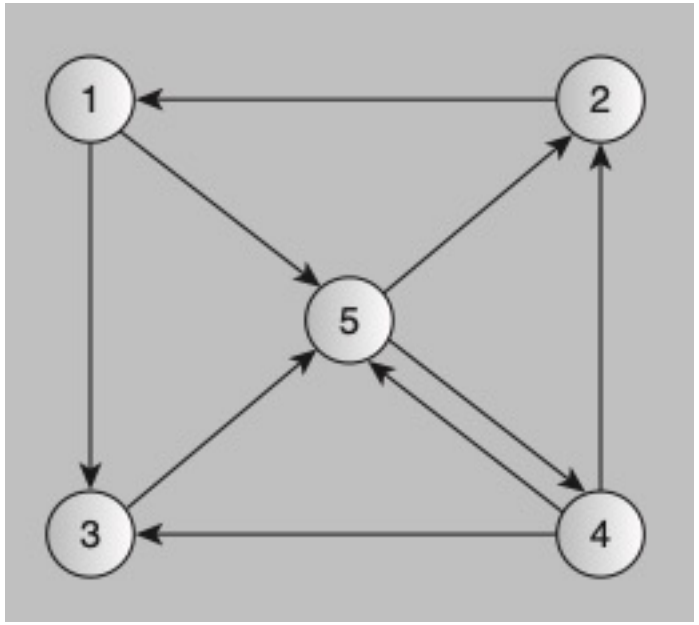


Illustration as Random Surfer Model

- Imagine a user randomly clicking on links on the web.
- If the user is on page 1
- They have two links to choose from: one to page 3 and another to page 5.
- They'll choose either with a 50% chance.
  - Probabilities – ½ for both page 3 and 5
- They can't directly jump to pages 2 or 4 from page 1.
  - Probability - 0

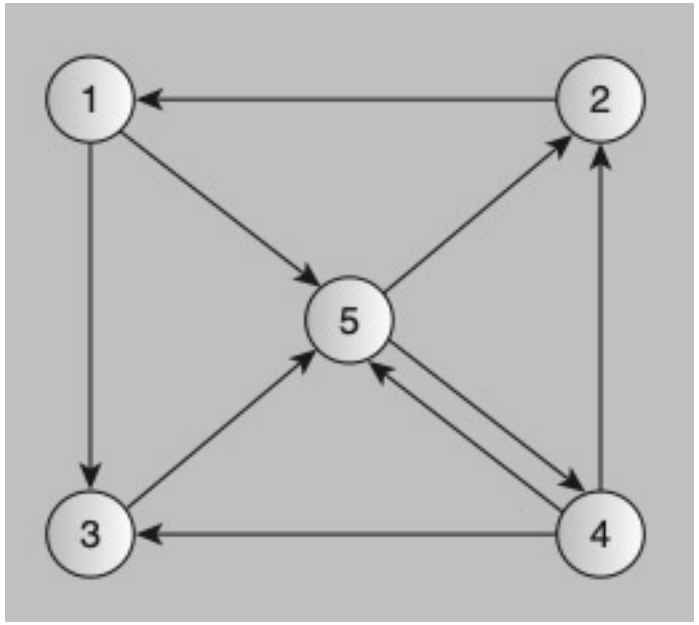


Illustration as Transition Matrix

$$M_5 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 1 & \frac{1}{3} & 0 \end{bmatrix}$$

- Matrix that represents the probabilities of moving from one page to another.
- Each column represents a web page.
- Each entry in the matrix shows the probability of transitioning from one page to another.
- Column 2 represents node 2, only outline to node 1 hence first row element is 1.

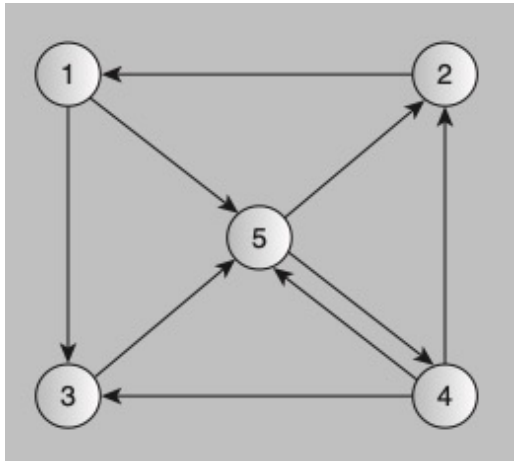


Illustration as Transition Matrix

Matrix Dimensions:

- The matrix M is of size $n \times n$ where n is the total number of web pages.

Matrix Entry Calculation:

For any pair of web pages Pi and Pj:

- If Pj has an outlink to Pi, the entry in the matrix M at row i and column j is:
  $M(i,j) = \frac{1}{k}$
  Where k is the total number of outlinks from page Pj.
- If Pj does not have an outlink to Pi, the entry is:
  $M(i,j) = 0$

PageRank Computation

PR Computation – Random Surfer Model

- Consider a vector v of size n
- n → number of web pages
- J^{th} component is the probability that the surfer is at page j.
- This probability is an indication of PageRank value of that page
- Initially the surfere can be at any of the n pages with prob = 1/n

$$v_0 = \begin{bmatrix} 1/n \\ 1/n \\ \vdots \\ \vdots \\ 1/n \\ 1/n \end{bmatrix}$$

PR Computation – Random Surfer Model

- Recall M_5 - Transition Matrix
- If M Satisfies
  - Sum of entries of any column is always equal to 1
  - All entries have values greater or equal to zero

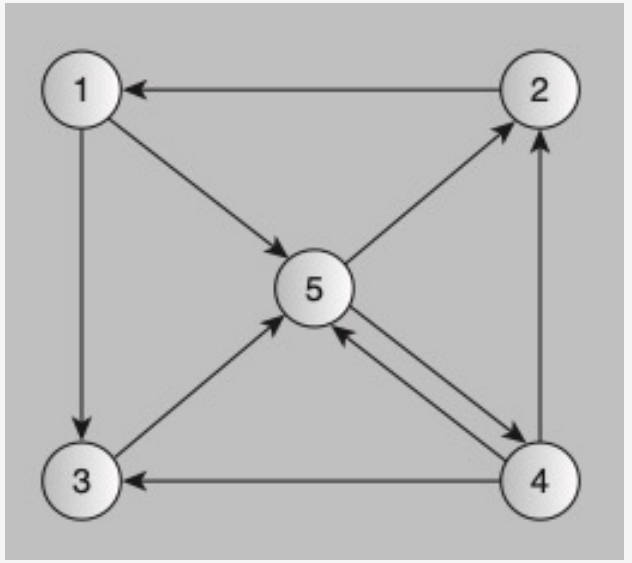
If yes, then it is called – A matrix of a Markov Chain process.

Also called - Markov Transition Matrix

Markov Chain:

- A process where the next state depends only on the current state.
- The web can be viewed as a Markov chain where each page is a state.
- m_{ij} – Probability of next node ***i*** provided current node is ***j***.

$$M_5 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 1 & \frac{1}{3} & 0 \end{bmatrix}$$



PR Computation – Random Surfer Model

- Consider server is at node j.
- Transition probabilities – v_j in M
- vj – column vector giving prob that current location is node j
- Use v and M to get distribution vector of next state x=Mv
- The new distribution $x = M \times v_j = \sum_j m_{ij} \times v_j$

Iterative Transitions:

- After the first step, the distribution is Mv_0, where v_0 is the initial distribution.
- After two steps, the distribution becomes $M(Mv_0)$.
- In general, after k steps, the distribution is $M^k v_0$, which means we multiply the matrix M by itself k times and then multiply the result by the initial distribution v_0.

PR Computation – Random Surfer Model

- This process cannot continue indefinitely.
- After a while, surfer starts to visit certain web pages more often then other pages.
- Slowly the visit frequency converges to fixed, steady-state quantity
- The distribution vector v remains same
- The final equilibrium state value v is the PageRank value of each node.

PR Computation – Random Surfer Model

Reaching Equilibrium

- Equilibrium in a Markov chain requires:
  - Strong connectivity in the graph.
  - No dead ends (each node has at least one outlink).
- Typically true for the World Wide Web.
- When these conditions are met, there's a unique steady-state probability vector, the principal left eigenvector of the Markov chain's matrix.
- In this case, v is the principal eigenvector of matrix M.
- Eigenvector: $v = \beta Mv$ (where β is a constant eigenvalue).
- All columns of matrix M sum to 1, so the associated eigenvalue for the principal eigenvector is also 1.

PR Computation – Random Surfer Model

- PageRank computation involves finding the principal left eigenvector of matrix M with eigenvalue 1.
- Matrix M can be very large for the entire web.
- For entire web M could contain billion rows and columns
- The Power method is a simple iterative algorithm for this task.
- Iterative computation: $x_k = M^k(Mv_0)$
- Values in xk stabilize after many iterations.
- k represents the iteration step
- Stabilized values indicate PageRank values.
- Typically, 60-80 iterations are needed for convergence.

PR Computation – Example

- Consider our example of M as before.

$$M_5 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 1 & \frac{1}{3} & 0 \end{bmatrix} \qquad v_0 = \begin{bmatrix} 1/5 \\ 1/5 \\ 1/5 \\ 1/5 \\ 1/5 \end{bmatrix}$$

$$x_k = M^k(Mv_0)$$

$$\begin{bmatrix} 1/5 \\ 1/5 \\ 1/5 \\ 1/5 \\ 1/5 \end{bmatrix} \begin{bmatrix} 1/5 \\ 1/6 \\ 1/6 \\ 1/10 \\ 11/30 \end{bmatrix} \begin{bmatrix} 1/6 \\ 13/60 \\ 2/15 \\ 11/60 \\ 3/10 \end{bmatrix} \cdots\cdots\cdots \begin{bmatrix} 0.4313 \\ 0.4313 \\ 0.3235 \\ 0.3235 \\ 0.6470 \end{bmatrix}$$

Iteration Count = 60

Page 1 – PR = 0.4313
Page 2 – PR = 0.4313
Page 3 – PR = 0.3235
Page 4 – PR = 0.3235
Page 5 – PR = 0.6470 - Highest

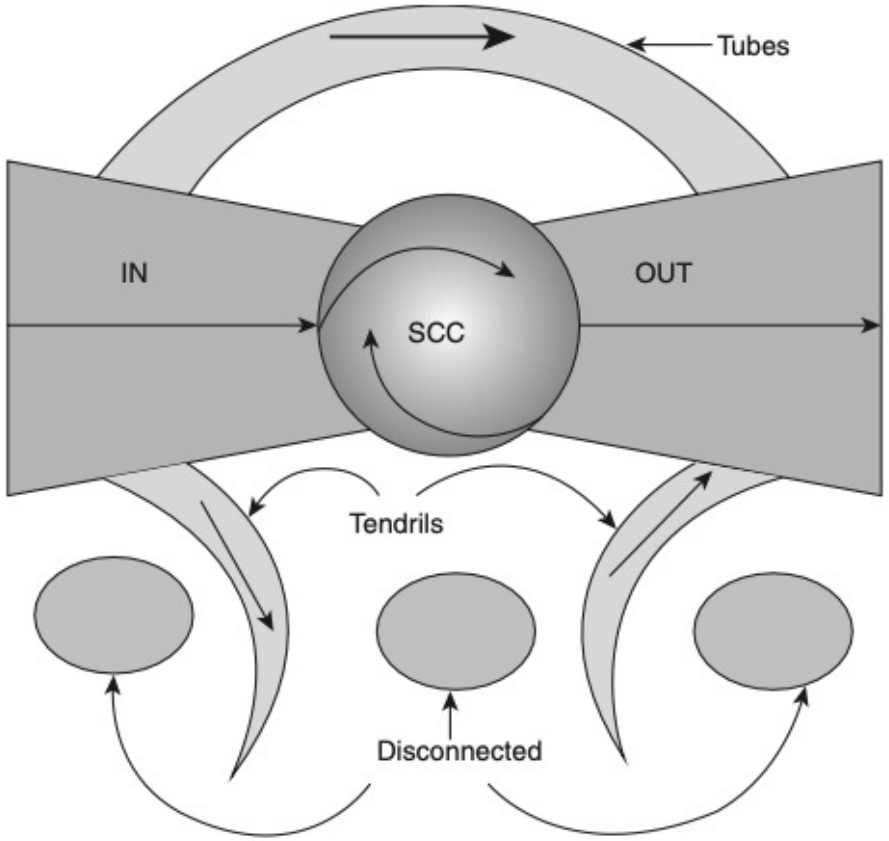
Structure of the Web

Structure of the Web

- PageRank assumes the entire web is one strongly connected entity.
- A 2000 study by IBM, AltaVista, and Compaq Systems challenged this assumption.
- The study analyzed 200 million webpages and 1.5 billion hyperlinks.
- It found the web is divided into three regions:
  - **Core:** A strongly connected portion where any surfer can reach any webpage from any other in the core.
  - **IN-Component:** Contains webpages with links to the core but no paths from the core leading to them.
  - **Out Component:** Includes nodes reachable from the core but lacking links leading back to it.

Bow-tie Structure of the Web

- SCC, IN and OUT
- Tendrils:
  - Tendrils extend from the IN and Out components.
  - Smaller groups of web pages that are indirectly connected to the core, either through the IN or OUT.
  - Can be thought of as sub-communities or clusters of web pages.
- Tubes:
  - Pages that reach from IN to OUT without linking to any page in SCC
- Disconnected components
  - Isolated clusters of web pages
  - Not connected to the core or any other significant part of the Bow-tie
  - Pages that exist independent of the main web structure

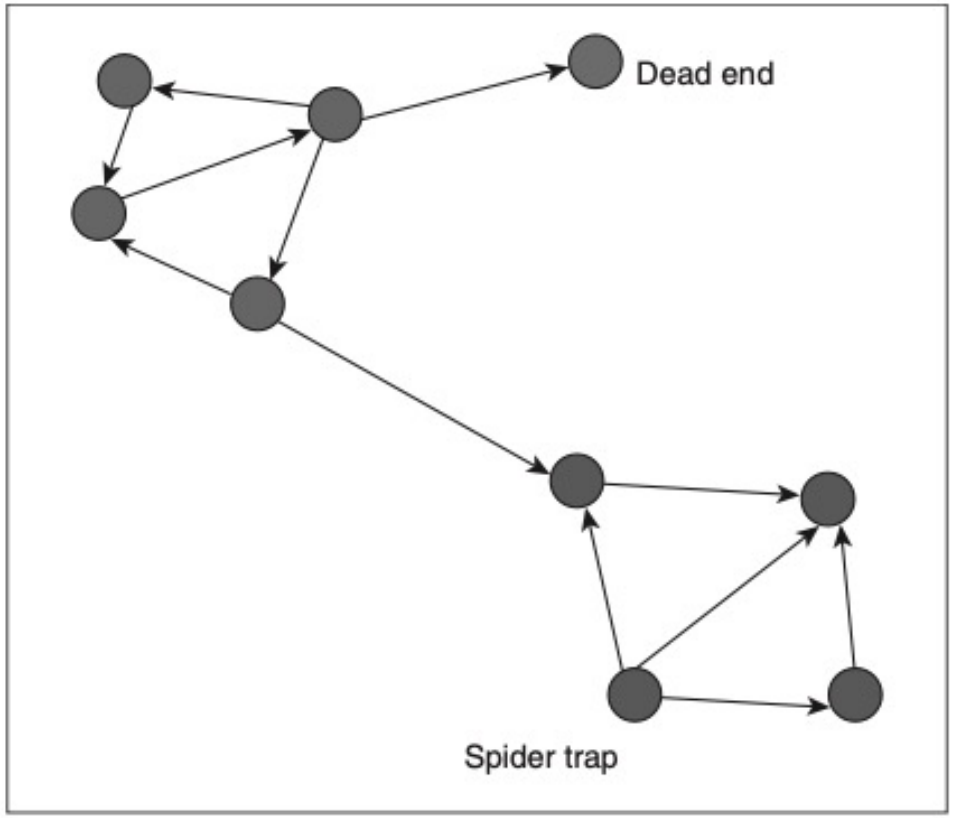


Bow-tie Structure of the Web

- The size of each region in the bow-tie structure was examined.
- Surprisingly, the core, while the largest, makes up only about one-third of the web.
- Origination and termination pages each constitute about a quarter, and disconnected pages about one-fifth.
- The bow-tie structure challenges assumptions used for Markov process convergence in PageRank computation.
- For instance, once a surfer lands in the OUT or out-tendril of the IN, they cannot reach the SCC or IN, resulting in a zero probability.
- This underscores the need for PageRank computation to consider the web's structure.

Structure of the Web

- **Dead Ends**
- Dead ends are web pages with no outlinks.
- They cause a loss of PageRank because surfers can't navigate further.
- **Spider Traps**
- Spider traps are sets of web pages where outlinks only lead to other pages within the same set.
- This causes an imbalance in PageRank distribution.
- **Taxation**
- Taxation allows surfers to leave the web at any step and start randomly at a new page.
- Addresses the Dead Ends and Spider Traps



Modified PageRank

Problem Statement

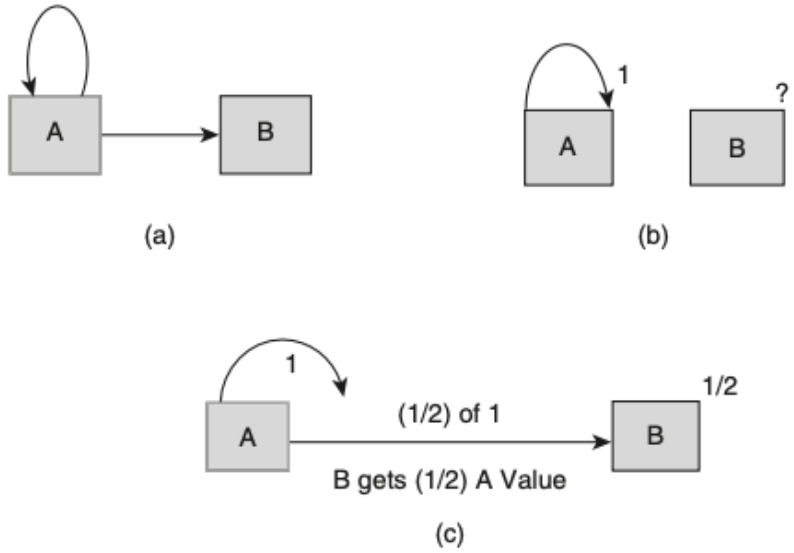
- Dead ends in a web network can create problems in the PageRank algorithm.
- In such cases, some columns in the transition matrix M may not sum to 1, causing components to reach a PageRank of 0.
- To overcome these issues, modifications to the basic PageRank algorithm are done.

Dealing with Dead Ends

- Dead-end pages with no outgoing links and their in-links are removed iteratively until an SCC remains.
- The order and iteration of removal are tracked.
- PageRank for the reduced graph G (SCC) is computed.
- Removed pages and links are restored in reverse order.
- The last set of nodes with in-links directly from the SCC are restored first.
- When restoring a dead-end page, its PageRank is computed as the sum of contributions from in-links within the SCC, adjusted by the number of outgoing links from each source page.

Dealing with Dead Ends

- A and B two nodes are there
- Identify dead ends – B
- Remove dead ends – Remove B
- Compute PR (A) – 1
- Restore dead Ends – Restore B
- Give half of PR(A) to B because there was only A pointed to B before removal.
- PR(B) = ½ PR(A)



Avoiding Spider traps

- Using Teleport Operation
- The surfer jumps from their current node to any other random node in the Web graph.
- If a node has no outlinks, teleport with some probability.
- If a node has outlinks, choose an outlink with probability β or teleport with probability 1–β
- Typical values for β might be 0.8 – 0.9
- Modified PageRank Formula

$$v' = \beta M v + (1 - \beta) e/n$$

v': The new PageRank vector after one iteration.
β: A probability factor.
M: Transition matrix.
v: Current PageRank vector.
e: Vector of all ones.
n: Total number of nodes in the graph.

Using PR in Search Engines

Using PageRank In a Search Engine

- PageRank is a critical factor used in determining the ranking of a web page
- PR relies on three main factors
- **Page-Specific Factors**
  - Things like body text, title tags, URL of the web pages
  - Google calculates Information Retrieval (IR) Score based on these page specific factors
- **Anchor Text of Inbound Links**
  - Google considers the text used in links pointing to a web page
  - Helps to establish the context or topic of the linked page
- **PageRank**
  - Measures a page's significance

Using PageRank In a Search Engine

- Ranking Process
  - Combines IR score and PageRank.
  - PageRank more influential for single-word queries.
  - Content-related factors key for multi-word queries.
- Algorithm Refinements:
  - Google continuously updates and refines its ranking algorithm.
  - Approximately 250 page-specific properties considered.
  - Specific details of Google's algorithm are closely guarded.

Effective Computation of PageRank

Effective Computation of PageRank

- Analyzing web graph structure for search engines involves billions of pages and their links.
- Average pages have 10–15 outlinks; doorway pages may have 25–30.
- Transition matrices for PageRank are vast and mostly sparse.
- Storing such matrices requires quadratic space.
- Efficient storage methods consider only non-zero values.
- Options include listing non-zero outlinks with values (linear space) and storing outlink counts with destination nodes (adjacency list-like structure).
- This will require 4 bytes integers for node value, 8 bytes floating point number for value of the link
- One more optimization can be done.

Effective Computation of PageRank

- Each value in the M is just a fraction
- 1 divided by number of out-links
- Sufficient to store node, number of out-links and destinations

$$M_5 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 & 1/2 \\ 1/2 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 \\ 1/2 & 0 & 1 & 1/3 & 0 \end{bmatrix}$$

Source nodes	Out degree	Destination nodes
1	2	3, 5
2	1	1
3	1	5
4	3	2, 3, 5
5	2	2, 4

Adjacency List

PR Implementation using MapReduce

- The step used to update PageRank in one iteration is given by –

$$v' = \beta M v + (1 - \beta) e/n$$

v': The new PageRank vector after one iteration.
β: A probability factor. = typical value 0.85
M: Transition matrix.
v: Current PageRank vector.
e: Vector of all ones.
n: Total number of nodes in the graph.

- When the number of notes is small, each Map task can store the vector v fully in Main Memory and also the v'.
- After that it is only a simple Matrix vector multiplication

PR Implementation using MapReduce

- Assume we have enough RAM to fit v', plus some working memory
  - Store v and matrix M on disk

Basic Algorithm:

- Initialize: $v = [1/N]_N$
- Iterate:
  - Update: Perform a sequential scan of M and v and update v'
  - Write out v' to disk as v for next iteration
  - Every few iterations, compute $|v - v'|$ and stop if it is below threshold

- The threshold used in PageRank computations can vary based on the application and desired accuracy.
- A common threshold value is a small positive number, often set to 0.001 or 0.0001.
- The threshold signifies that iterations continue until the difference between PageRank values in two consecutive iterations is less than the chosen threshold.