Unit 5 Web searching

* The world wide web (WWW) has experienced expoential expansion since the end of 1980. Massive amounts of data exists in the form of many media, including text, photos, audio, video etc.

* one terabyte or more is thought to be the approximate size of the textual data alone.

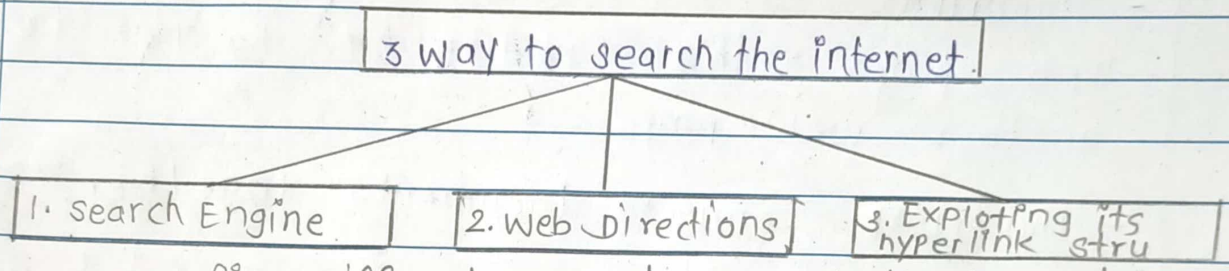


fig : Different ways to search the internet.

* **Search Engine.**

- search Engines allow you to search documents or web pages on the world wide web from a remote machine.

- It uses keyword search.

- searche Engines use web crawlers to visit websites & gather data for indexation.

- Be aware that the results you receive from one search engine might not be the those from another.

✲ Web Directories.

- A web directory is a list of websites that organized by category.
- web directories are populated through appl and approval processes as opposed to searc engine which utilise web crawlers to visit website and gather data for indexation.
- Different submission procedure are used by various search directories.

✲ Hyperlink structure.

- Hyperlink is a clickable thing on a web page that redirect ~~to~~ you to a another web page.
- Hyperlink connects documents together.

✲ comparision between web Engine & web Directories for web search. ✲

SrNo	Topic.	search Engine	web Directories.
1	Used for	Used to locate content online	Used to locate content online.
2.	Useful for finding	Find websites through keyword.	Find website by subject.
3	Result contains	List of web pages.	List of websites.
4.	Type of Result got.	Result of the search engine is not biased	Depending on the own beliefs web director can even be biased.
5.	Human intervation	A computer program decides which websites should be listed.	A human decides whic website should be liste hence the complete directory is selected

challenges in web searching.

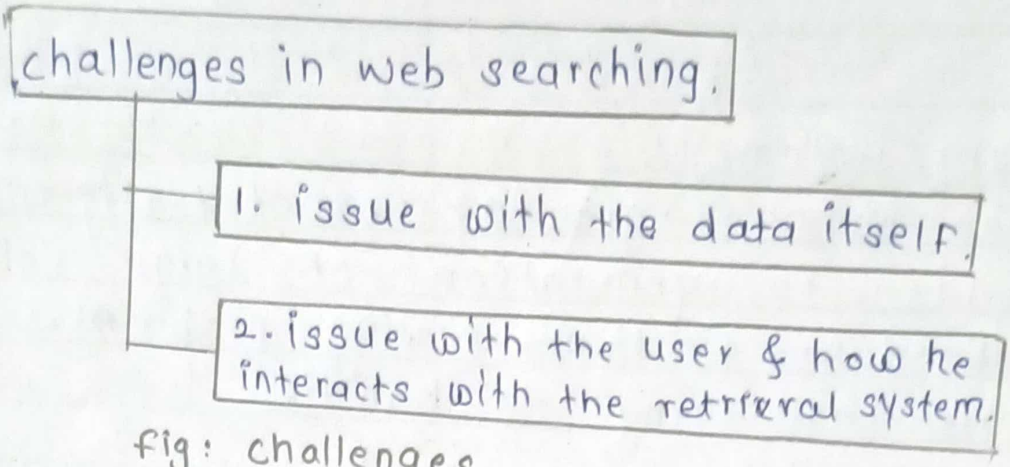


fig: challenges.

* Problems Related to Data.

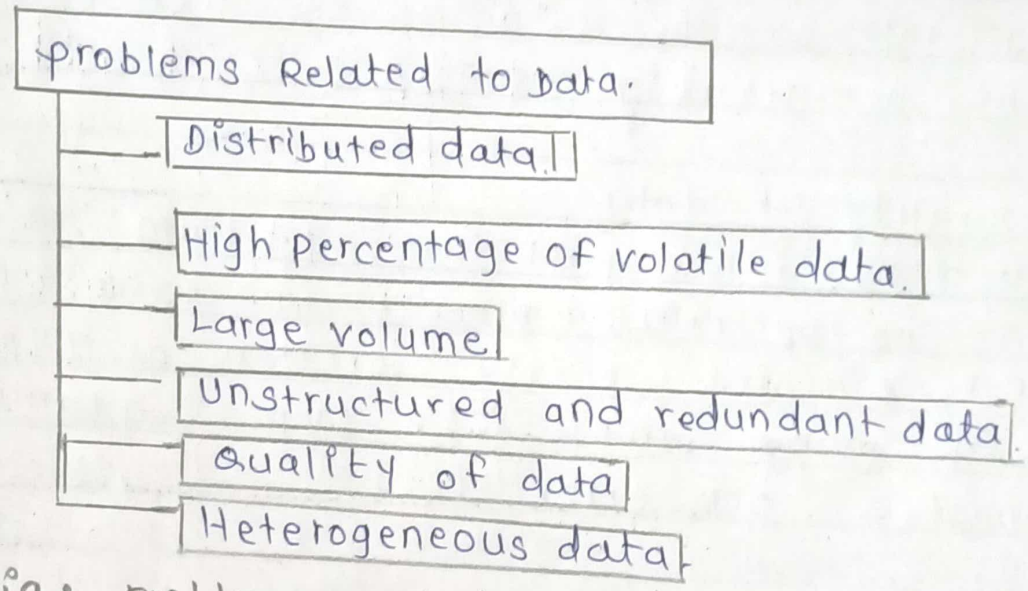


Fig: Problems related to Data.

Distributed Data.

Data is spread across numerous machine & platforms The computers connected have no predefined topology.

High percentage of volatile data.

Due to internet dynamics it is simple to add and remove new computer, websites & pages some percentage of web changes every month.

3) Large volume
The web is growing expoentrally which leads to generation of large volume of data & thus scaling problems that are challenging to handle.

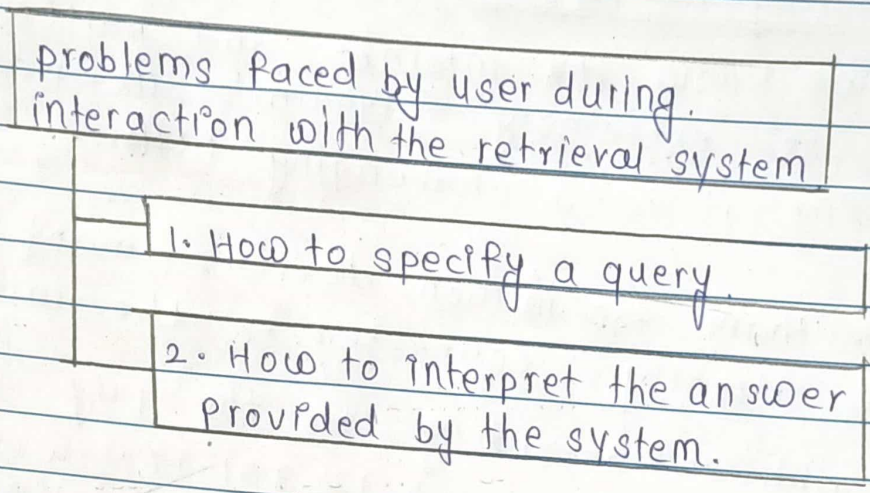
4) Unstructured and Redundant Data.
Each HTML page is not well structured and a lot of the data on the web is duplicated. or remarkably similar

5) Quality of data.
web is a new publishing medium where there is no editorial process so data can be false, invalid, poorly written or typically with many errors like types grammetical mistak, OCR error etc.

6) Heterogenous data.
There exist a variety of media types, multiple formats for their reprsentation and different language with different alphabets are used to reprsent them.

☆ problems faced by user during interaction with the retrieval system.

- The second class of problems are those in which users face some problems while interacting system.



- Defining relatively short queries short queries is ok but if the queries are long and complicated it becomes challenging to define it properly without taking the document's semantic content into account.

- Therefore, the main challenges in submitting an effective search query and getting a manageable and relevant answer.

Web characterstics:

- Web characterstics.
  - 1. Measuring the web.
  - 2. Modeling the web.

Fig. characterising the web.

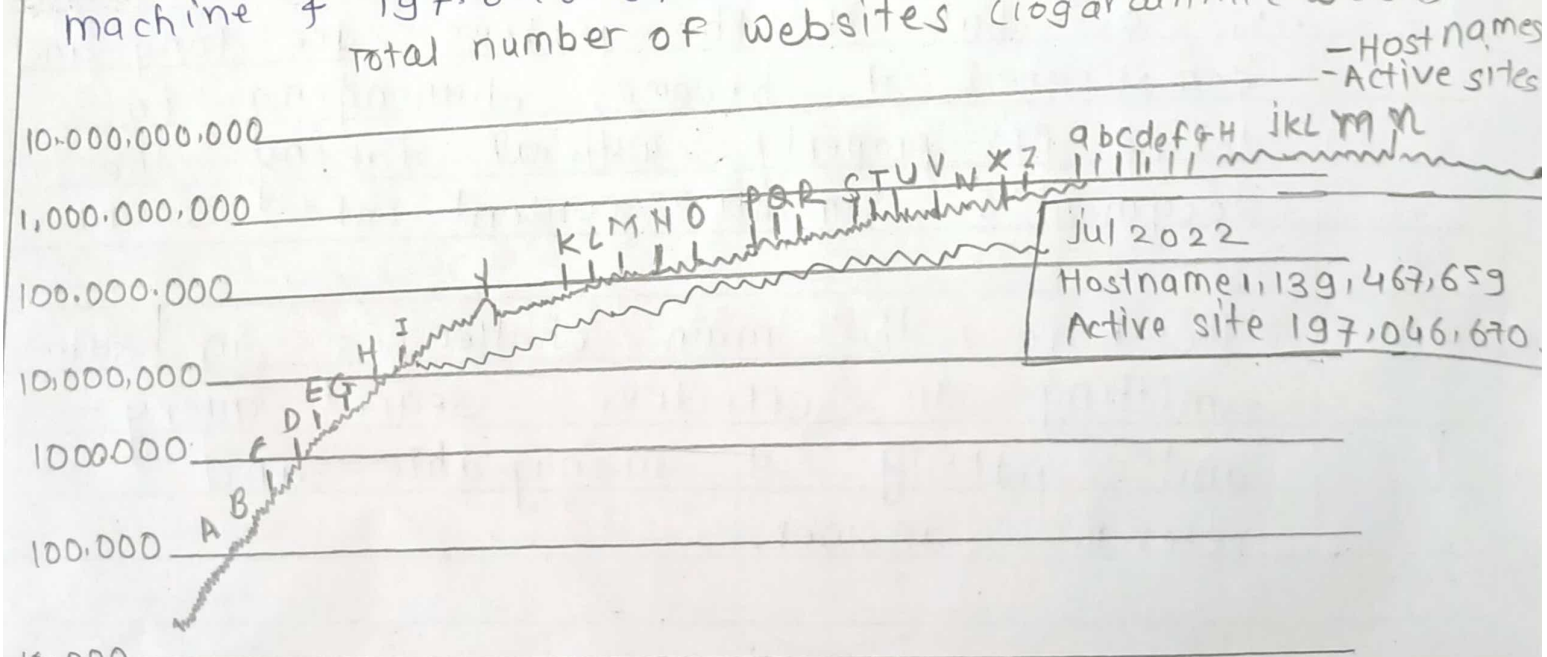
Measuring the Web:

- As we know the internet and particularity the web is extremely dynamic in nature, measuring it is a challenging task.

- more thane 750 million devices, many of which were directly connected to the internet in 2010.

In addition according to the july 2022 Netcraft web survey. there are 12,341,172 web facing machine & 197,046,670253 active sites.

Total number of websites (logarathmic scale)



Net craft survey of July 2022 on total no of website.

HTML is the most widely used format for web content followed by ASCII text, GIF & JPG both pictures) then PDF in that order.

* There are many interseting traits and data of HTML pages, some of which are included here.

- first off, the majority of html pages are not standards, meaning that they do not adhere to all HTML requirments.

- In reality, many pages would not be rendered if browsers behave strictly as HTML compilers.

* Modeling the Job.

- web modeling addresses the specific issues related to design & development of large scale web application.

- All characterizations of the web show that the quantitative properties of the CORE compoent follow a power law distribution.

- This invariance is also called self similarity. If F(x) is probability distribution a must be set such that the sum of all probability is 1.

Search Engine:

- This section discusses various architecture of retriv system that use the the web as a full text data source.

- Standard IR systems and the web vary sfm primarily in that the web requires that all query processing and ranking the done only using indice where accessing the test is not needed.

- so we need not store the copy of web pages locally, also we need not have to access pages remotely at query time.

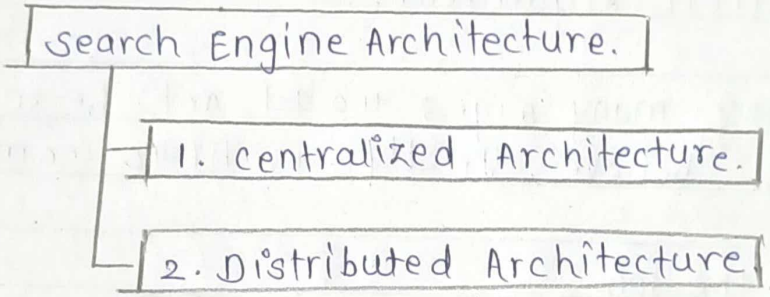


Fig: Types of search Engine Architecture.

Centralized Architecture.:-

- A centralized crawler-indexer architecture is used by most search engine.

- crawlers are software agents that provide fresh or updated web pages to a central server for indexing.

- A crawlers is not actually sent to and executed on remote machine., it operate locally instead making request to distant web servers.

- The created index is used to respond to inquiries are other names for crawlers.

- Inverted index consists of a lists of terms vacabwary, where each term is associated with list of pointers to the pages in which it occur.

- In ~~ider~~ idexing procedure, normalization operations are carried out, which may involve removing punctuation, replacing several space between words with a single space, and changing capital to lowercase letters.

- A centralized crawler-indexer architecture is divided into two section.

section 1 manages user requests.

section 2 deals with data

contains:
1. User interface and
2. Query Engine.

contains:
1. crawler and
2. Indexer.

fig: centralized crawler - Indexer Modules.

- The schematic software architecture of an early search engine like Alta vista.

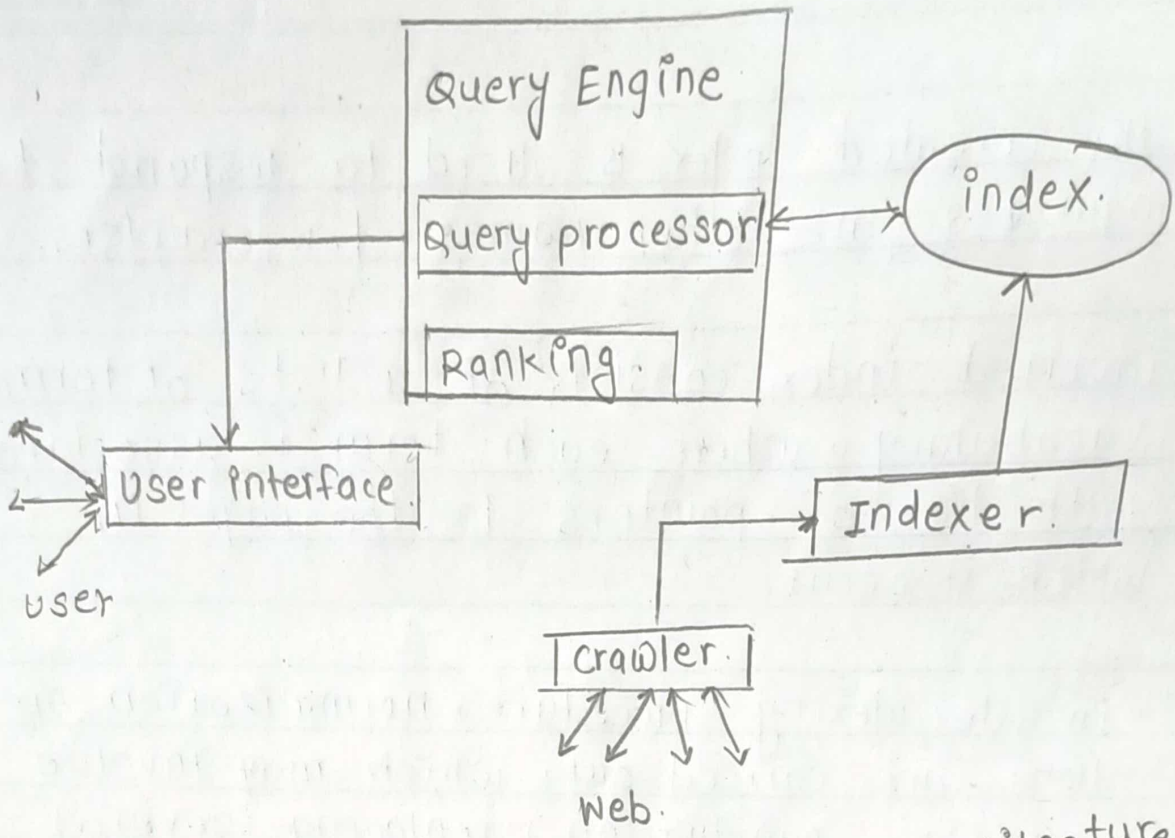


fig: centralized Crawler-Indexer Architecture.

1) User interface - Through user interface, user submits the query for searching. It is a mediator between the user and retrieval system.

2) Query Engine - Here the user query is refined and processed using the query processor. and using the ranking module, the results that are the most relevant are shown first at the time of display.

3) crawler -. It is a software programs which runs locally and send request to remote servers for new tab Page.

4) Indexer - It extracts the words from every page & records the URL where each word occured it creates an index of a document.

✡ Limitations of crawler Indexer Architecture.

i) Difficult to cope up with the growth of job.
ii) High load on servers.
iii) Difficult to gather data because of the dynamicity of the web.

✡ Distributed Architecture:

- The crawler - indexer architecture has different variants.

- One of them is Harvest which is an early example.

- Harvest gathers and distributes data using a distributes data using a distributed fashion, which is more effective than the traditional web crawler arch.

- There are two main compoents in the architecture gathers and brokers.

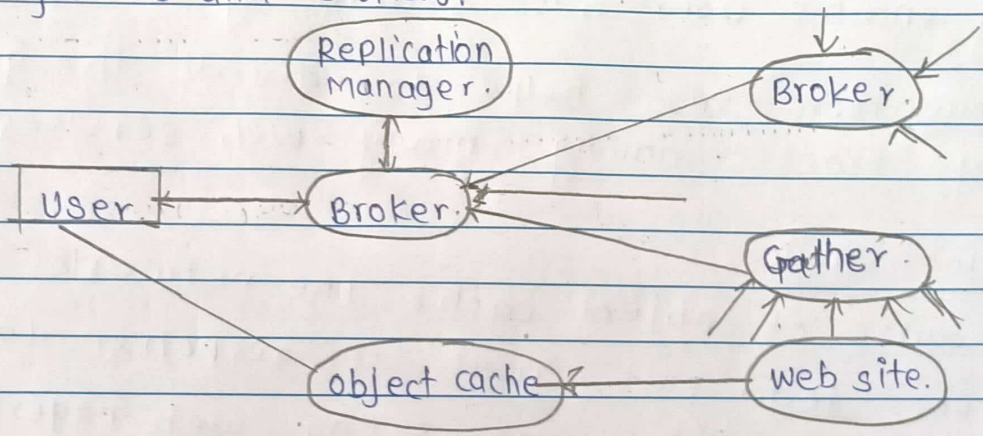


Fig: Distributed Harvest Architecture.

1) Gatherer:

- A gatherer collects and extracts the indexing informati from one or more web servers.

- The system assigns the time to gatherer, for gathering the information & it is periodic in nature.

- Gatherer can run on a web server, generating no external traffic for that server.

2] Broker:

- Task of the broker is to provide the indexing mechanism & they query interface to the allowed data.

- Brokers get data from gatherers or other brokers and update incrementtly their indexes.

- Broker can also filter information & send it to to other broker.

3) Replicator -

1) The replicators are used to replicate servers. they enhance user base scalability.

- Replication also helps to divide the gathe- - ring process among many web services.

4) object cache.

- Because of object cache, the network and server load, as well as response latency is reduced while accesseng web pages.

✡ Advantage of Distributed / Harvest Architecture.

1) Increased server load due to simultaneous requests from various crawlers.

2) Increased web traffic as a result of crawlers retriving entire objects, even though most content eventually lost ; and.

3) Lack of coordination between engines as data is gathered indpendently by each crawler.

✡ Limitations of Distributed / Harvest Architecture.

Sr No	centralized Architecture	Distributed Architecture.
1.	consists of crawler & indexer	consists of gatherer & broker.
2.	Does not provide replication and objects cache.	provide replication & object cache.
3.	Difficult to cope up with the growth of the web.	cope up with the growth of theweb efficiently by distributing the work.
4.	Crawlers provide new web pages to central server for indexing.	Indexing is done by brokers in distributed fashion.
5.	Increased web traffic as a result of crawlers retrieving entire objects.	Gatherer can run on a web server, generating no external traffic for that server

✱ User interfaces.

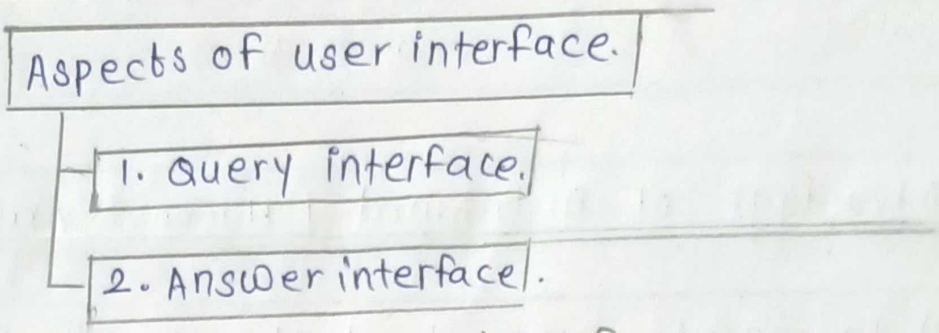


Fig: Aspects of User interface

✱ Query interface.

Although a user ~~interface~~ may believe that a given sequence of words reprsents the same query in all search engines, it is not the case.

- For example in altavista a sequence of words in a user query is a reference to the web pages having all the words while in Hotbot it is a reference to the web page having all the words.

- Another problem is that some search engines use stopwords, some do stemming and some are not case sensitive.

- All search engines also provide a query interface & complex queries as well as a command language including boolean operators and other features such as phrase search proximity search & wild cards

* Answer interface:

- The answer usually consists of a list of the ten top ranked web pages.
- Each entry in the list includes some information about the document it represents.
- Typically the information includes the url and a couple of lines with it content.
- The order of the list is typically by relevance.
- The web page retreived by the search engine in response to a user query are ranked usually using statistics related to the terms in the query.
- some search engine also take into account term included in the metatags or the title or the popularity of the webpage improve the ranking.

The answer interface is for one of the most popular search engine is Google. The result includes top 10 web pages with URL, the order of the list is by relevance.

Ranking :

- Ranking means position of web page in the result generated by search engines.

- The Higher the ranking, the more relevant the web page is.

- The rankeng is important as highly ranked website get more vistors.

- Most search engines use variatfons of the boolean and vector model for ranking.

The number of links pointing to a page indicates its popularity and quality in link base ranking.

In addition to the traditional TF-IDF technique TF-IDF technique, yuwano and Lee offered three additional ranking algorithms, boolean spread, Vector, ~~vec~~ and most cited.

The first two are extensions of the standard boolean & vector space model ranking algorithms that take into account page that are pointed to or referenced by page in the response.

- only the terms mentioned in pages with links to the page in the answer are used in the third algorithm, "most cited".

- webquery is another early illustration that enable visual of web pages.

- Web query rates a collection of web pages according to how related they are to one another additionally it brodens. the set by identifying web pages that have a strong relationship to the starting test.

- The set of pages s that point to or pointed by pages in the answer are taken intoaccount by the Hits (Hypertext induced topic search ranking method, which is query-dependent.

- pages have many links pointing to it in s are called authorities.

- pages that have many outgoing links are called authorities.

- Pages that have many outgoing links are called hubs.

- Better authority pages arise from good hubs. and better hub pages comes from outgoing edges to good authorities, creating a positive two way feedback loop.

- Let H(b) and A(p) be the hub and authority value of page p. these values are defined such that the following eqn are satisfied.

$H(p) = \sum_{v:p \to v} A(v), \ldots A(p) = \sum_{v:p \to v} H(v)$

- These values can be determined through an iterative algorithm & they coverage to the principle eigenvector of the link matrix of S.

- In the case of the web to avoid an ~~explanati~~ explosion on the size of s, a maximal number of page pointing to the answer can be defined.

- one solution is to weigh each link is based on the surrounding content.

- The best known link based weight is pageRank.

- This technique does not work with non existent, repeated, or automatically generated links.

✱ Challenges in Ranking :

i) Ranking is the hardest & most important function search engine have to execute & it has some challenges.

1) The first challenge is a creating a suitable d'evaluation process that enables assesing a ranking's effectiveness in terms of its relevance to uses.

2) finding a good quality content on the web is the second challenge.

3) A third difficulty has been introduced by the present advertising based business model used by search engine preventing web spam.

4) The fourth issue lies in defining the ranking function & computing it.

Page rank :

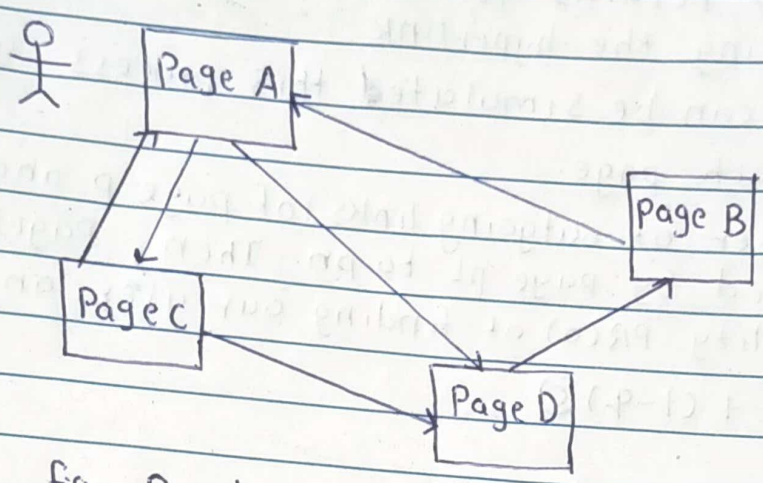


fig. Random walk in Page Rank

- Page Rank is a ranking algorithm originally used by google
- Page Rank simulates a user navigating randomly on the Web, as shown in fig
- Consider that our user is currently at page a. Following, she moves to one of the pages pointed by page a by randomly selecting one of the hyperlinks Next, she repeats the process for the page she moved to and so on. After large number of such moves, we can compute the probability with which our user visited each page. This probability is a property of graph, which was referred to as PageRank in the context of the Web.
- Real Web graphs includes self links and dead ends or page without external links. In order to prevent the user from becoming stuck on these sites, a second situation is taken into account in which she can navigate to any other page with a low probability q.

As a result, our user either follows a hyperlink on the page with probability 1-q or jumps to a random page probability q.

According to the rules of this random walk on the web graph this users never returns to the website they recently viewed by clicking the hyperlink.

A Markov chain can be simulated this process, giving stationary probability of each page.

- Let L(p) be number of outgoing links of page p and suppose the page a is pointed by page p1 to pn. Then, PageRank of page a is given by probability PR(a) of finding our user and is defined by

$$PR(a) = + (1-q) \text{©}$$

where,

PR(a) = Page Rank of page a

q - parameter that must be set by system (typical value 0.15)

T - total number of pages

PR(Pi) - Pank rank of incoming pages to page a in last iteration

L(Pi) - outbound link of page pointing to page a.

- Some technical problems arise while calculating PageRank. The biggest one is how to handle dead ends or "sink nodes" in the Markov chain i.e pages that do not have outgoing links
- Using q=1 for these pages is one option.
- Another option is to remove them and compute only their PageRank
- Another formulat for finding page rank is

$$PR(A) = (1-d) + d \text{©}$$

where d = damping factor

- Damping factor is used when page gets stuck in specific area.
- If damping factor is not provided it is taken as 0.85 by default
- Page rank algo uses number of iterations to get best page

Crawling the web:

- Web crawling is the process of indexing data on web pages by using automated scripts.
- Crawlers begin their crawling by downloading the robot.txt file from website.
- Robot.txt file has sitemaps i.e the list of URLs that the search engine can crawl
- Web crawler use those links to find new pages once they begin crawling a page.
- To crawl newly found URLs later, these crawlers add them to crawl queue.
- When we retrieve a page, we obtain its actual content
- How recently updated are the web pages? The age of page ranges from one day to two months.
- Because of this, the date that page was indexed is displayed in search engine's response.
- There are 2 and 9% links that are retained by search engine.
- Some search engine navigate entire website, while others pick just a sample of page
- Every page that is linked to another website can be indexed by web crawler.
- The current fastest crawlers are able to transverse upto 10 million web pages per day.

Several Techniques ~~of~~ to Crawl the Web

- The easiest method is to begin with a set of URLs and then, either in breadth-first ~~and~~ or deepth-first approach extract further URLs

- A variation is to start with a set of popular URLS, because except that they have information frequently requested.
- Both seenarios work well for single crawler, but it is challenging to coordinate multiple crawler.
- Another method is to divide the web into segments using web addresses, assign one or more robots to each segment and exam each segment.

Web Crawler :

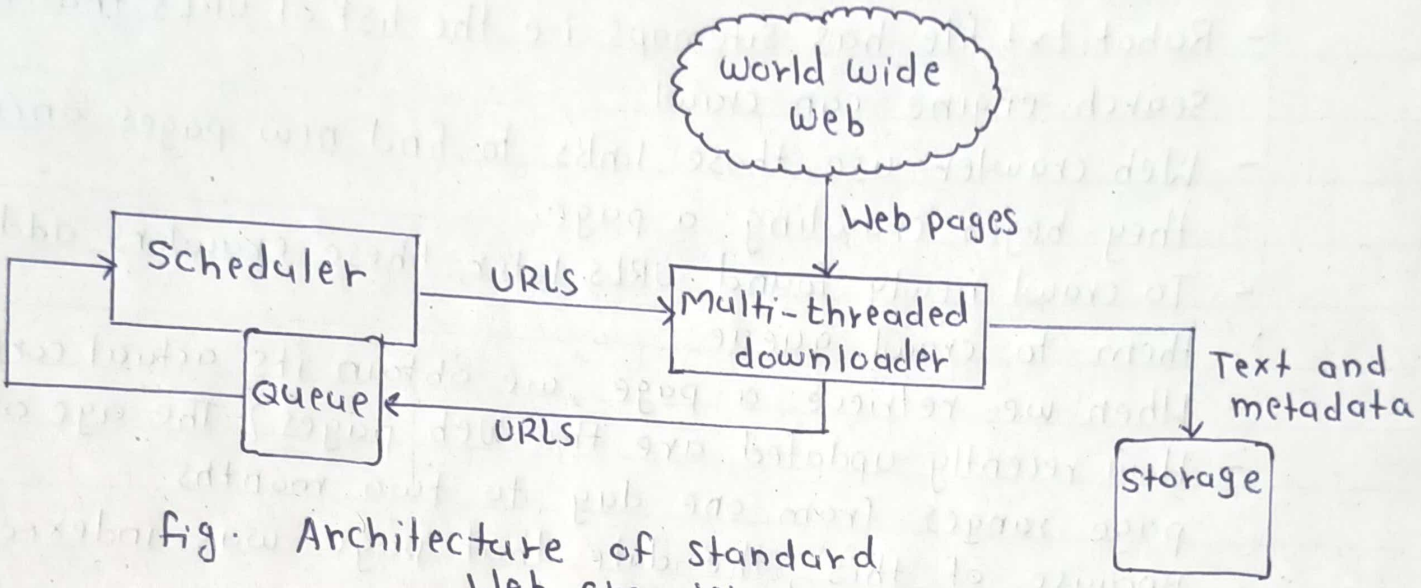


fig. Architecture of standard Web crawler

Following fig is Architecture of web crawler, where web page are transversed and text and metadata is saved to storage, while link or URLS from that page are added to queue for next transversal, scheduler works like a frontier from which next URL is taken to sear

Components of web Crawler

Robots.txt file

- Robot.txt file is just a text file with no HTML markup code.
- Many websites have robot.txt files in their source code file.
- This file has a predefined set of instructions for bot.
- The robot.txt file for any given website can typically be viewed by typing the full URL for homepage and adding /robots.txt
- For ex if you want to find Robots.txt file of Amazon onlin shopping site, just type its URL and add /robots.txt to it

- A robots.txt file can only provide guidelines for bots.
- A decent bot will attempt to visit the robots.txt file before examining any other page on site and will do in accordance with rules, such as web crawler
- A bad bot will either disregard the robots.txt file or process it to identify the websites

1. Spider: spider are software programs which download robots.txt file and other pages requested by crawler manager for extracting and processing URLs.
2. Link Extractor: From page downloaded by spider it extracts the URLs from the links present in those pages and send those URLs to crawler manager.
3. Crawler Manager: It sends the set of URL's obtained from Link extractor to DNS resolver to obtain its corresponding logical address.

Indices:

- Most indices make use of variants of inverted file
- Inverted file is list of words that has been sorted, and each word has a collection of pointers to pages where it appears.
- Some search engine removes stopwords to reduce index size
- It's crucial to remember that content is indexed in logical order.
- The normalization process involves changing uppercas to lowercase letters, removing punctuations and goi from numerous spaces to just one betn each word
- A query is answered by doing binary search on sorted

- If we are searching multiple words, the result have to be combined to generate final answer.
- Another possibility is to compute the complete answer while user requests more web pages, using lazy evaluation scheme.
- Inverted files can also point to the actual occurrences of a word within document.
- Finding words ~~with~~ which start with a given prefix requires two binary searches in sorted list.
- The index can be less dense if we point to logical blocks.

Browsing :

This section discusses web-based browsing and searching, for Web directories. Less than 1% of all web pages are covered by directories, but despite this, user gets more relevant response

Web Directories:

- Web Directories are also called catalogs, yellow pages, or subject directories.
- Directories are hierarchical taxonomies that classify human knowle
- The first level of taxonomies used by Web directories can be

1) Arts and Humanities
2) Automotive
3) Business and Economy
4) Computer and Internet Education
5) Employment
6) Entertainment
7) Games
8) Government
9) Health and Fitness
10) Hobbies and Interests
11) Home Investing
12) Kids and Family
13) Life and Style
14) Living
15) Local News
16) oddities
17) People
18) Philosophy and Religion
19) Politics Recreations
20) Reference
21) Regional
22) Science and Technology
23) Shopping and services
24) Social Science
25) Society and culture
26) Sports
27) Travel and Tourism World

- In most cases, pages have to be submitted to the Web directories, where they are reviewed, and, if accepted, classified in one or more categories of hierarchy
- Although the taxonomy can be seen as a tree, there are cross references, so it is really a directed acyclic graph
- Advantage of this technique is that if we find what we are looking for, the answer will be useful in most cases.

Combining Searching with Browsing:
- Usually, users either browse following links or they search a website.
- currently, in web directories, a search can be reduced to a subtree of taxonomy.
- However, the search may miss related page that are not in that part of taxonomy.
- Some search engines find similar pages using common words, but often this is not effective.
- WebGlimpse is a tool that tries to solve these problems by combining browsing with searching.

Metasearchers:
- Metasearchers are web servers that send a given query to several search engines, Web directories and other databases, collect the answers and unify them.

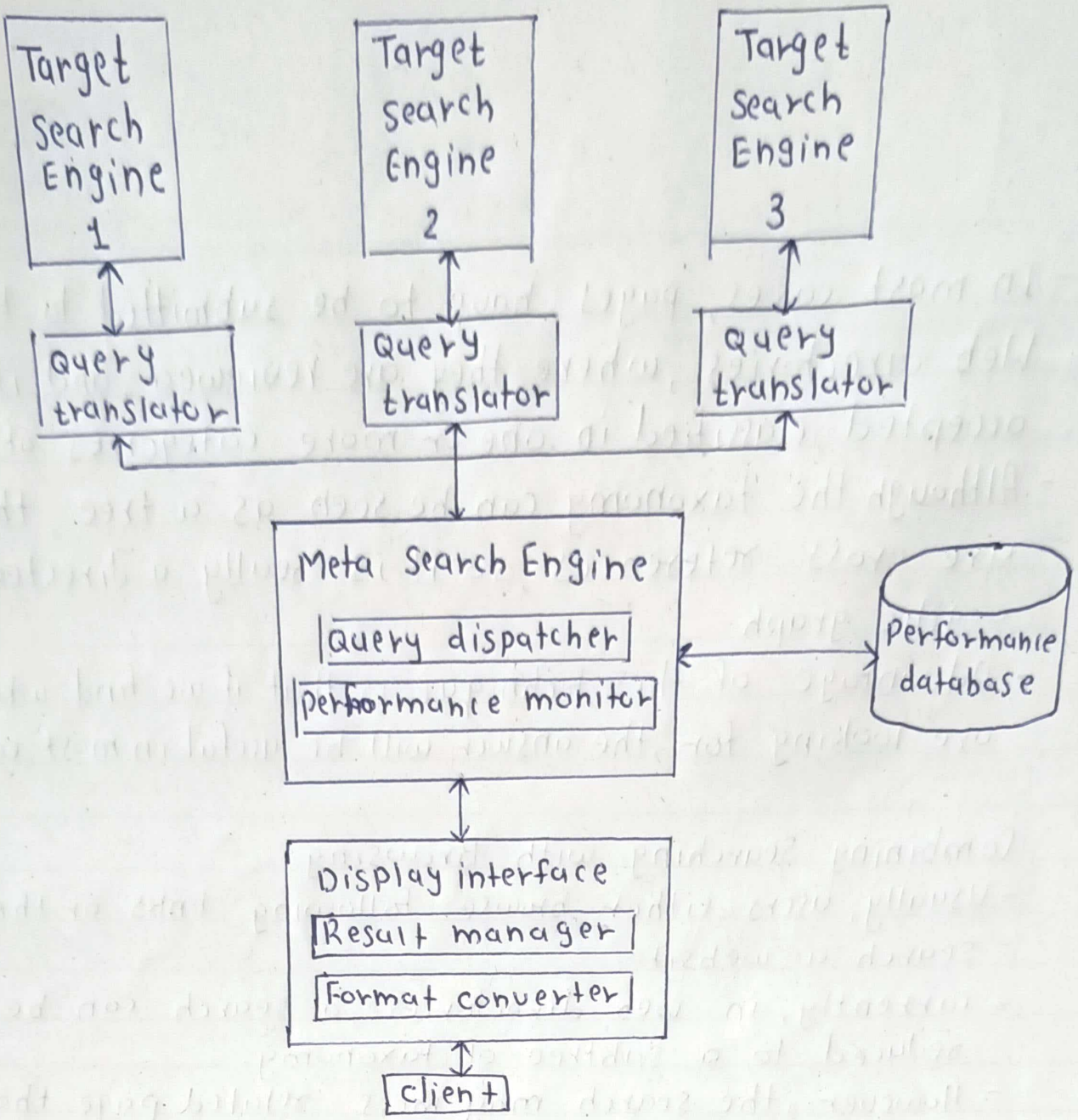


fig. Metasearchers

- In the architecture the client submits the query through interface.
- After receiving the user query, the meta search engine translate it into a source specific query. It also translates the query result from underlying source to common date format.
- old ex of metadata searchers are Metacrawler and Savvysearch
- Metasearchers differ from each other in how ranking is performed in the unified result, and how well they translate the user query to the specific query language of each search engine.
- Metasearchers can also ran on the client
- Lis of Metasearch Engines:

1) AOL Search
2) ApocalX
3) All the Internet
4) carrot2
5) ceeKyP
6) curry Guide
7) Dogpile
8) Draze
9) Fazzle
10) Entireweb

Advantages of Metasearchers

- By sending multiple queries to several other se engines this extends the coverage data of the top and allows more information to be found.
- The results can be sorted by different attributes such as host, keyword, datae, etc, which can be more informative than the output of single search engine.
- Browsing the results should be simpler
- Pages returned by more than one search engine are more relevant.

Disadvantages of Metasearchers

- Metasearch engines are not capable of parsing query forms or able to fully translate query syntax.
- The number of results per search engine retrieval by the metasearchers is limited.
- The result of metasearcher is not necessarily all the Web pages matching the query

Trends and Research Issues:

- The future of the Web might surprise us, considering its massive use.
- There are numerous diverse trends, and each one creates fresh, unique study questions.
- Modeling, Querying, Distributed Architectures, Ranking Indexing, Dyamic pages, Duplicated data, Multimedia, User interfaces, Browsing all these trends need to do even better.

An important issue to be settled in the future is a stand protocol to query search engine.

- Hyperlinks can also be used to infer information about t
- Although this is not exactly searching the web this is an important trend called Web mining.
- Other trends are intranet applications, Web document cluster Building models to see Web sites as databases and/or infor tion system.

Introduction To Web Scraping:

- Web Scraping is automated gathering of data from interne
- The term "scraping" refers to obtaining the information from another source and saving it into a local file.
- To transform unstructured data from a webpage into struct- ured data, web scraping is helpful.
- Web scrapping is not a new term introduced, previously it has so many flavours like screen scraping, data mining, web traves- ting.
- Web scrapers do ~~not~~ an excellent task of gathering and processing large amounts of data quickly.
- Using web scraper, you can view databases spanning thousands or even millions of page at once.
- In addition, web scrapers can go places that traditional search engines cannot.
- If you want to buy any product from online selling sites, you will visit multiple sites, check availability of product, do comparasion of prices and then choose site that will sell it at reasonable price. But these task need a lot of time and effort.
- A well-developed web scraper can chart the cost of a item, across a variety of websites and tell you the best site to choose.
- some sites allow scraping when used legally, web scraping is illegal if ~~someth~~ someone tries to scrap the nonpublic data.

Python for Web Scraping

- Many languages support web scraping, but python has its own advantage like, it has vast collection of libraries it is dynamically typed, it is open source community.
- So in this section we will see how to use python to request information from web server, how to perform basic handling of the server's response and how to begin interacting with a website in an automated fashion.

•Steps for web scraping

1) Find the URL that you want to scrape
2) Inspect the Page
3) Write the code
4) Store the data in required standard format

Before performing actual web scraping, you must first install the Requests module and beautifulsoup package using following commands:

1. pip install beautiful soup
2. pip install requests

1. Request

- Requests module is used to make a http request to a specific URL.
- This module provides inbuilt functionalities for managing request and response.
- Get method is used to retrieve information from the given server using the given URL

Python code for printing content

```
# import request module
import request

# make request to web server using get()
req = requests.get("http://snjb.org/jain-gurukul/")

# print content
print(req.content)
```

Response:

If you want to check response from web server then just print req.status_code or simply req, if the code printed is 200 then you got a success.

2. Beautiful Soup

- Beautiful soup is a Python package for parsing HTML and XML documents
- It creats parse tree that are ~~related~~ helpful to extract the data easily.
- It helps format and organise the messy web by fixing bad HTML and presenting us with easily transuersable Python objects reprsenting XML structures.
- It works with the parser to provide a natural way of navigating, searching and modifing the parse tree.

Installation of Beautiful Soup

```
pip install bs4
```

- Write a python code in any well known python IDE:

1) Import beautiful soup and request module
2) Request for a particular website using requests.get()
3) Create instance of beautiful soup including parser name
4) Check and print the contents from the soup.
5) Run python file.

- If you want to display only text then instead of printing soup print soup.get_text()

```
import requests
from bs4 import BeautifulSoup
req=requests.get("http://snjb.org/jain-gurukul/")
soup=BeautifulSoup(req.content,"html.parser")
print(soup.get_text())
```

- It dislays text on particular webpage

HTML Parsing

- Parsing means to divide something into its components and then describe their synatic roles.
- Once we get raw HTML code, we can parse it to produce some valuable information.
- In order to ~~see~~ use a particular parser, we must first construct a BeautifulSoups object.
- BeautifulSoup supports HTML parser and several third party Python parsers. You can install any of them according to your needs. Following is the list of Beautiful Soup's parser:
  1) Python's html.parser
  2) lxml's XML parser
  3) lxml's HTML parser
  4) HTML5lib
- on the Flipkart website, an HTML parser is used.