CST466
DATA MINING

MODULE-5

Module 5 (Advanced Data Mining Techniques)

Web Mining - Web Content Mining, Web Structure Mining- Page Rank, Clever, Web Usage Mining- Preprocessing, Data structures, Pattern Discovery, Pattern Analysis. Text Mining-Text Data Analysis and information Retrieval, Basic measures for Text retrieval, Text Retrieval methods, Text Indexing Techniques, Query Processing Techniques.

Introduction - Text Mining:

- *Most previous studies of data mining have focused on structured data - such as relational, transactional, and data warehouse data.*
- *In reality, a substantial portion of the available information is stored in text databases (or document databases), which consist of large collections of documents from various sources, such as news articles, research papers, books, digital libraries, e-mail messages, and Web pages.*
- *Text databases are rapidly growing due to the increasing amount of information available in electronic form, such as electronic publications, various kinds of electronic documents, e-mail, and the World Wide Web.*
- *Nowadays most of the information in government, industry, business, and other institutions are stored electronically, in the form of text databases.*

TEXT MINING:

- In reality, a substantial portion of the available information is stored in **text databases** (or document databases).
  - Text databases are rapidly growing due to the increasing amount of information available in electronic form.
  - Text databases consists of large collections of documents from various sources, such as news articles, research papers, books, digital libraries, e-mail messages, and Web pages.
- Text mining is the process of **extracting meaningful information and insights from large collections of textual data.**
- Data stored in most text databases are **semistructured data.**
  - ie, they are neither completely unstructured nor completely structured.
  - For example, a document may contain;
    - Structured fields, such as title, authors, publication_date, category etc.,
    - Unstructured text components, such as abstract and contents.

- **Traditional information retrieval techniques become inadequate** for the increasingly vast amounts of text data.
  - **Only a small fraction of the many available documents will be relevant to a given individual user.**
- Without knowing what could be in the documents, it is difficult to formulate effective queries for analyzing and extracting useful information from the data.
- So, **it is very important to have effective tools to compare different documents, rank the importance and relevance of the documents, or find patterns and trends across multiple documents.**
- Thus, text mining has become an increasingly popular and essential theme in data mining.

Text Data Analysis and Information Retrieval:

Information retrieval systems vs Database systems:

- Information retrieval (IR) is a field that has been developing in parallel with database systems for many years.
- Information retrieval is concerned with the organization and retrieval of information from a large number of text-based documents.
- Database systems are focused on query and transaction processing of structured data.
- Since IR and database systems each handle different kinds of data, some database system problems are usually not present in IR systems and vice versa.

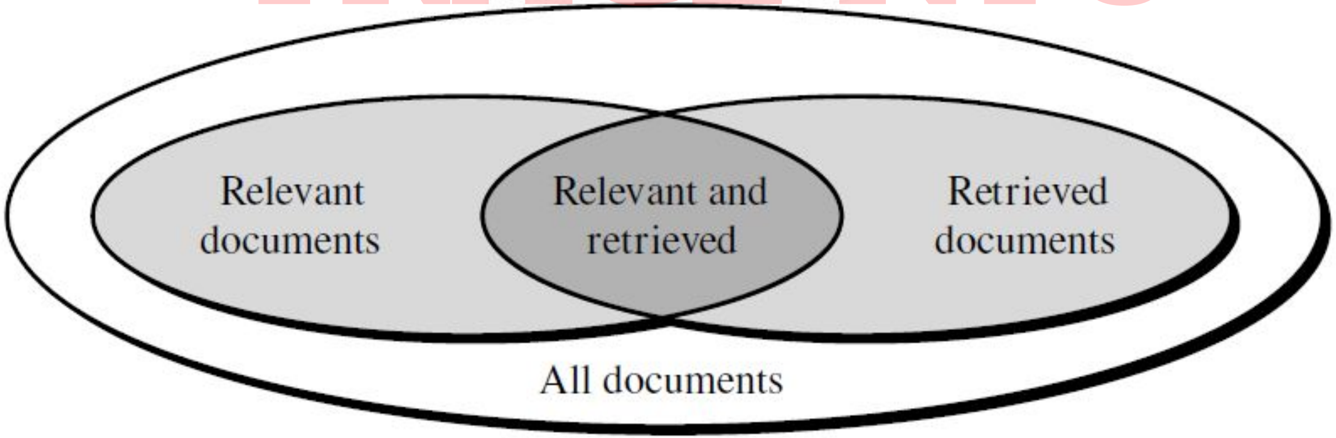
- A typical information retrieval problem is **to locate relevant documents in a document collection based on a user's query**.
  - Here, the user's query is often some keywords describing an information need.
- In such a search problem, a user takes the initiative to "pull" the relevant information out from the collection.
- User mostly have some ad hoc (i.e., short-term) information needs.
  - Eg: Finding information to buy a used car.
- When a user has a long-term information need (e.g., a researcher's interests), a retrieval system takes the initiative to **"push" any newly arrived information item to a user if the item is judged as being relevant to the user's information need.**
- Such an information access process is called **information filtering**, and the corresponding systems are often called **filtering systems or recommender systems.**

Basic Measures for Text Retrieval: Precision and Recall**

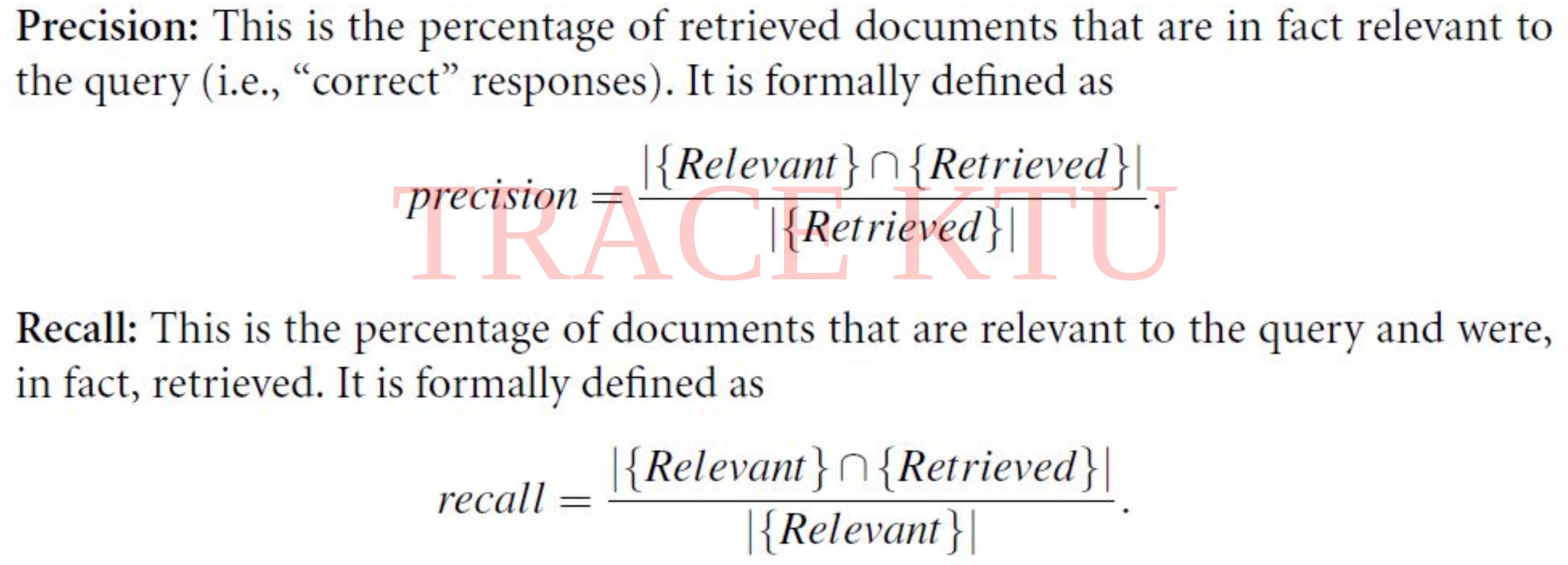
"Suppose that a text retrieval system has just retrieved a number of documents based on the input query given.

How can we assess how accurate or correct the system was?"

- The set of documents relevant to a query - **{Relevant}**
- The set of documents retrieved - **{Retrieved}**
- The set of documents that are both relevant and retrieved - **{Relevant}** ∩**{Retrieved}**



- There are **two basic measures for assessing the quality of text retrieval**:



Precision: This is the percentage of retrieved documents that are in fact relevant to the query (i.e., "correct" responses). It is formally defined as

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|}.$$

Recall: This is the percentage of documents that are relevant to the query and were, in fact, retrieved. It is formally defined as

$$recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|}.$$

- An information retrieval system often needs to trade off recall for precision or vice versa.
- The commonly used trade-off is the **F-score**, which is defined as the harmonic mean of recall and precision:

$$F\_score = \frac{recall \times precision}{(recall + precision)/2}.$$

- The harmonic mean discourages a system that sacrifices one measure for another too drastically.
- Precision, recall, and F-score are the basic measures of a retrieved set of documents.
- Anyway, these three measures are not directly useful for comparing two ranked lists of documents.

TEXT RETRIEVAL METHODS:***

- Information retrieval methods fall into two categories.
- The retrieval methods can be;
  - **A document selection method**

    or

  - **A document ranking method**

Document selection methods:

- In document selection methods, the **query is regarded as specifying constraints for selecting relevant documents**.

- A typical method of this category is the Boolean retrieval model.
  - Here, a document is represented by a set of keywords and a user provides a Boolean expression of keywords, such as “car and repair shops,” “tea or coffee,” or “database systems but not Oracle.”
- The retrieval system would take such a Boolean query and return documents that satisfy the Boolean expression.

Drawbacks:

- Prescribing a user’s information need exactly with a Boolean query is quite difficult.
- So, the Boolean retrieval method generally only works well when the user knows a lot about the document collection and can formulate a good query in this way.

Document ranking methods:

- Document ranking methods **use the query to rank all documents in the order of relevance**.
- For ordinary users and exploratory queries, these methods are more appropriate than document selection methods.
- Most modern information retrieval systems present a ranked list of documents in response to a user’s keyword query.
- There are many different ranking methods based on a large spectrum of mathematical foundations, including algebra, logic, probability, and statistics.
- The common intuition behind all of these methods is that we may **match the keywords in a query with those in the documents and score each document based on how well it matches the query.**
- The goal is to approximate the degree of relevance of a document with a score computed based on information such as the frequency of words in the document and the whole collection.
- It is inherently difficult to provide a precise measure of the degree of relevance between a set of keywords.
- For example, it is difficult to quantify the distance between data mining and data analysis.
- Comprehensive empirical evaluation is thus essential for validating any retrieval method.

Vector Space model:

- One of the most popular text retrieval method is the vector space model.
- We represent a document and a query both as vectors in a high-dimensional space.
- We use an appropriate similarity measure to compute the similarity between the query vector and the document vector.
- The similarity values can then be used for ranking documents.

How do we tokenize text???

- The first step in most retrieval systems is to identify keywords for representing documents, a preprocessing step often called **tokenization.**
- To avoid indexing useless words, a text retrieval system often associates a stop list with a set of documents.
- A **stop list** is a set of words that are deemed "irrelevant."
  - Eg: a, the, of, for, with, and so on are stop words, even though they may appear frequently.
- Stop lists may vary per document set.
  - Eg: Database systems could be an important keyword in a newspaper.
  - it may be considered as a stop word in a set of research papers presented in a database systems conference.

- A group of different words may share the same **word stem.**
- A text retrieval system needs to identify groups of words where the words in a group are small syntactic variants of one another and collect only the common word stem per group.
- Eg: The group of words drug, drugged, and drugs, share a common word stem, drug, and can be viewed as different occurrences of the same word.

How can we model a document to facilitate information retrieval???

- Starting with a set of d documents and a set of t terms, we can model each document as a vector v in the t dimensional space R^t .

ie, Given: Set of **d documents** and a set of **t terms.**

Each document is modeled as a vector v in the t-dimensional space R^t .

Term Frequency:(TF)

- The number of occurrences of term *t* in the document *d*.
- Denoted as **freq(d,t)**

Term-frequency matrix:

- Term-frequency matrix measures the association of a term *t* with respect to the given document *d.*
- It is generally defined as;
  - 0, if the document does not contain the term.
  - Non-zero, otherwise.
- Denoted as **TF(d, t)**
- There are many ways to define the term-weighting for the non-zero entries in such a vector.
  - For example,
    - We can simply set TF(d,t) = 1 if the term t occurs in the document d.
    - The **relative term frequency**, ie, the term frequency versus the total number of occurrences of all the terms in the document, otherwise.

- There are also other ways to normalize the term frequency.
- The Cornell SMART system uses the following formula to compute the (normalized) term frequency:

$$TF(d,t) = \begin{cases} 0 & if\ freq(d,t) = 0 \\ 1 + \log(1 + \log(freq(d,t))) & otherwise. \end{cases}$$

Inverse document frequency: (IDF)

- IDF represents the scaling factor, or the importance, of a term t.
- If a term t occurs in many documents, its importance will be scaled down due to its reduced discriminative power.
- Eg: The term "*database systems*" may likely be less important if it occurs in many research papers in a database system conference.

- According to the same Cornell SMART system, IDF(t) is defined by the following formula:

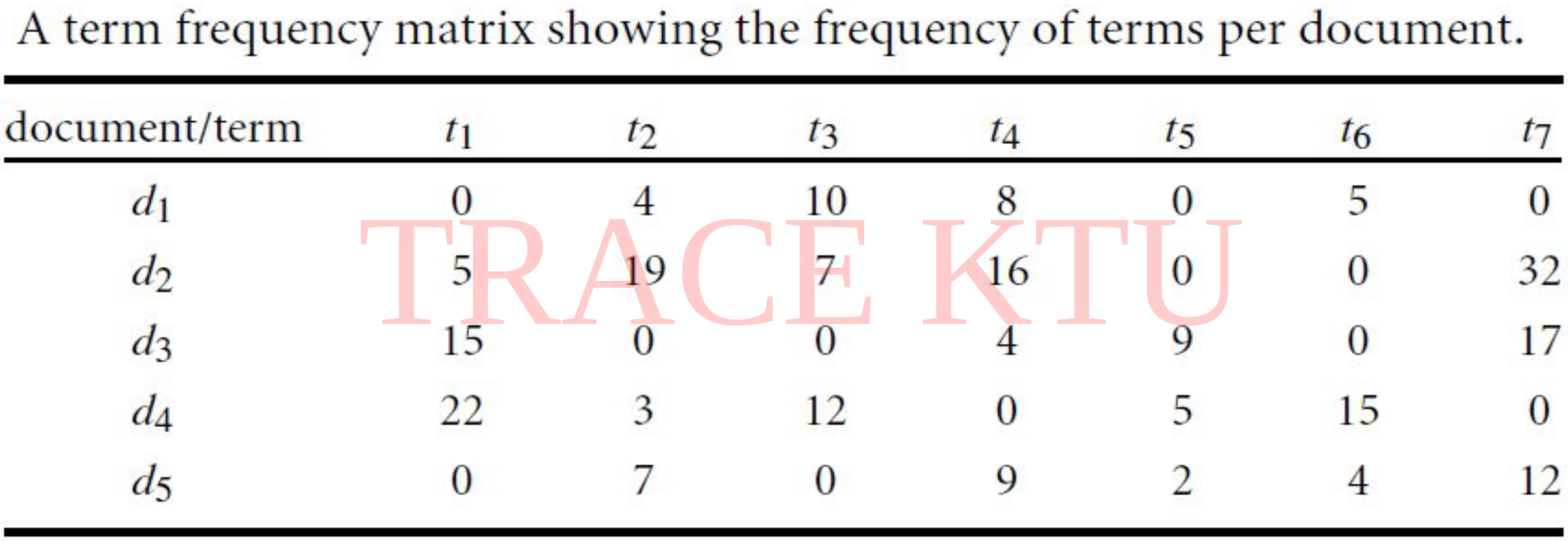
$$IDF(t) = \log \frac{1 + |d|}{|d_t|}$$

  - d is the document collection, and d_t is the set of documents containing term t.
  - If $|d_t| << |d|$, the term t will have a large IDF scaling factor and vice versa.
- In a complete vector-space model, TF and IDF are combined together, which forms the TF-IDF measure:

$$TF\text{-}IDF(d,t) = TF(d,t) \times IDF(t)$$

Sample question:

Q: Given a term-frequency matrix. Based on this, calculate the TF-IDF value of a term t_6 in document d_4.



A term frequency matrix showing the frequency of terms per document.

document/term	t_1	t_2	t_3	t_4	t_5	t_6	t_7
d_1	0	4	10	8	0	5	0
d_2	5	19	7	16	0	0	32
d_3	15	0	0	4	9	0	17
d_4	22	3	12	0	5	15	0
d_5	0	7	0	9	2	4	12

Solution:

$$TF(d_4,t_6) = 1 + \log(1 + \log(15)) = 1.3377$$

$$IDF(t_6) = \log\frac{1+5}{3} = 0.301.$$

$$TF\text{-}IDF(d_4,t_6) = 1.3377 \times 0.301 = 0.403$$

"How can we determine if two documents are similar?"

- **Similar documents are expected to have similar relative term frequencies.**
- So,we can measure the similarity among a set of documents or between a document and a query based on similar relative term occurrences in the frequency table.
- Many metrics have been proposed for measuring document similarity based on relative term occurrences or document vectors.
- A representative metric is the **cosine measure**, defined as follows.
  - *Let v1 and v2 be two document vectors.*
  - *Their **cosine similarity** is defined as;*

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1||v_2|}$$

where the inner product v1. v2 is the standard $|v_1| = \sqrt{v_1 \cdot v_1}$ *t, defined as* $\Sigma_{i=1}^{t} v_{1i} v_{2i}$

and the norm |v1| *in the denominator is defined as*

TEXT INDEXING TECHNIQUES:

- Following are the two popular **text retrieval indexing techniques**;
  - **Inverted indices**
  - **Signature files**

INVERTED INDEX:

- An inverted index is an index structure that **maintains two indexed tables;**
- **Document table:**
    - *document_table consists of a set of document records.*
    - *Each document record contains two fields:*
      - *doc_id and posting_list*
        - *posting_list is a list of terms (or pointers to terms) that occur in the document, sorted according to some relevance measure.*
- **Term table**
    - *term_table consists of a set of term records.*
    - *Each term record contains two fields:*
      - *term_id and posting_list*
        - *posting_list specifies a list of document identifiers in which the term appears.*

- With such an organization, it is easy to answer queries like;

  "Find all of the documents associated with a given set of terms,"

  Or

  "Find all of the terms associated with a given set of documents."

- Eg: To find all of the documents associated with a set of terms,
  - Firstly, check the term table.
  - For each term in the term table, there will be a list associated.
  - This list contains document identifiers of all documents which contains that term.
  - Then, intersect them to obtain the set of relevant documents.

Advantages:

- **Easy to implement.**
- Inverted indices are **widely used** in industry.

Drawbacks:

- The **posting lists could be rather long, making the storage requirement quite large.**
- They are easy to implement, but are **not satisfactory at handling synonymy** (where two very different words can have the same meaning) **and polysemy** (where an individual word may have many meanings).

SIGNATURE FILES:

- A signature file is a **file that stores a signature record for each document in the database.**
- Each signature has a fixed size of *b* bits representing terms.
- A simple encoding scheme goes as follows;
    - Each bit of a document signature is initialized to 0.
    - If the term it represents appears in the document , then the bit is set to 1.
    - If each bit that is set in signature S2 is also set in S1, then the signature S1 matches another signature S2.
    - Multiple terms may be mapped into the same bit.
    - Such multiple-to-one mappings make the search expensive.
        - Because, a document that matches the signature of a query does not necessarily contain the set of keywords of the query.
        - The document has to be retrieved, parsed, stemmed, and checked.
        - Improvements can be made by first performing frequency analysis, stemming, and by filtering stop words, and then using a hashing technique and superimposed coding technique to encode the list of terms into bit representation.

Drawback:

- Multiple-to-one mapping.
- Expensive search.

QUERY PROCESSING TECHNIQUES:

- Once an inverted index is created for a document collection, a retrieval system can answer a keyword query quickly by looking up which documents contain the query keywords.
- Specifically, we will maintain a **score accumulator** for each document and update these accumulators as we go through each query term.
    - For each query term, we will fetch all of the documents that match the term and increase their scores.
- When examples of relevant documents are available, the system can learn from such examples to improve retrieval performance.
    - This is called **relevance feedback** and has proven to be effective in improving retrieval performance.
- When we do not have such relevant examples, a system can assume the top few retrieved documents in some initial retrieval results to be relevant and extract more related keywords to expand a query.
    - Such feedback is called **pseudo-feedback** or **blind feedback**.
    - Pseudo-feedback also often leads to improved retrieval performance.

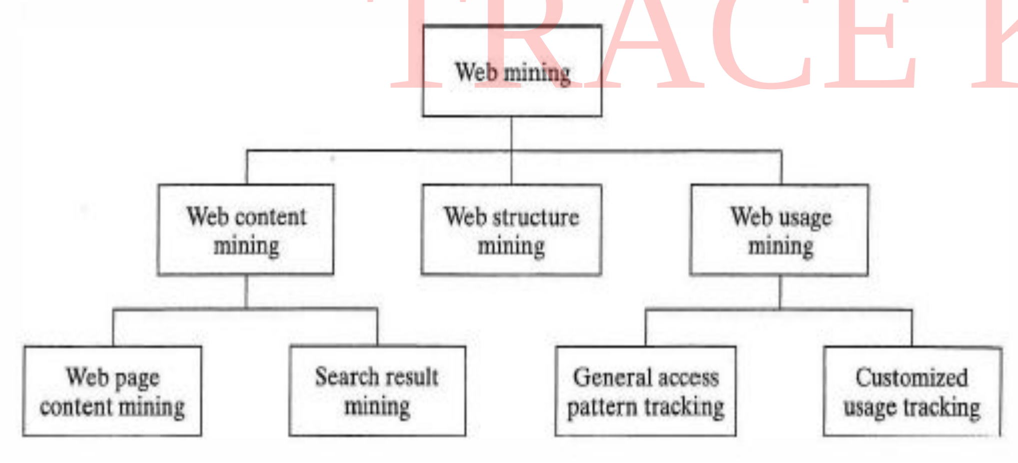
- One major limitation of many existing retrieval methods is that they are based on exact keyword matching.
- However, due to the complexity of natural languages, keyword-based retrieval can encounter two major difficulties.
  - **Synonymy problem:**
    - Two words with identical or similar meanings may have very different surface forms.
    - Eg: A user's query may use the word "automobile," but a relevant document may use "vehicle" instead of "automobile."
  - **Polysemy problem:**
    - The same keyword may mean different things in different contexts.
    - Eg: Word "mining" may mean different things in different contexts.

WEB MINING:

- Web mining is **mining of data related to the World Wide Web.**
- Web mining is the process of **extracting useful information and knowledge from** the;
  - **Content** data
  - **Structure** data
  - **Usage** data

of web pages and websites.

- It involves the application of data mining techniques to discover patterns, relationships, and insights from the vast amount of data available on the web.



Web mining can be classified into three categories;

- **Web content mining.**
- **Web structure mining.**
- **Web usage mining.**

Web content mining:

- Web content mining **examines the content of web pages as well as results of web searching.**
- The content includes text as well as graphics data.
- Web content mining is further divided into;
  - **Web page content mining.**
    - Traditional searching of web pages via content.
  - **Search results mining.**
    - Search of pages found from a previous search.
    - Thus, some mining activities have been built on top of traditional search engines, using their result as the data to be mined.

Web structure mining:

- With web structure mining, **information is obtained from the actual organization of pages on the Web.**
- Content mining is similar to the work performed by basic IR techniques, but it usually goes farther than simply employing keyword searching.
- Eg: Clustering may be applied to Web pages to identify similar pages.
- The intrapage structure includes links within the page as well as the code (HTML, XML) for the page.

Web usage mining:

- Web usage mining **looks at logs of web access.**
- Web usage mining is further divided into;
    - **General access pattern tracking**
    - **Customized usage tracking**
        - These types of usage mining that **looks at a history of Web pages visited.**
        - This usage may be **general or may be targeted** to specific usage or users.
        - Besides identifying what the traffic patterns look like, usage mining also involves the **mining of these sequential patterns.**
            - Eg: Patterns can be clustered based on their similarity.
            - This in turn can be used to cluster users into groups based on similar access behavior.

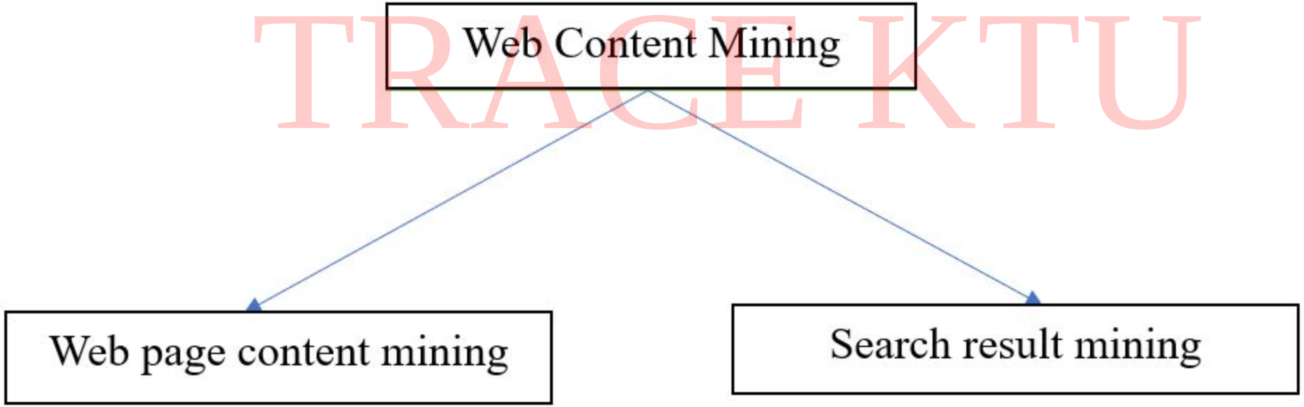
Note:

- *There are many* ***applications for Web mining.***
- *One application is targeted advertising.*
  - *Targeting is any technique that is used to direct business marketing or advertising to the most beneficial subset of the total population.*
  - *The objective is to maximize the results of the advertising.*
    - *ie, send it to all (and only) the set of potential customers who will buy.*
- *In this manner, the cost of sending an advertisement to someone who will not purchase that product can be avoided.*
- *Targeting attempts to send advertisements to people who have not been to a Web site to entice them to visit it.*
- *Thus, a targeted ad is found on a different Web site.*

WEB CONTENT MINING:

- It is the process of **extracting useful information and data from the World Wide Web.**
- This can involve;
  - **Analyzing website content**
  - **Analyzing search engine results, and social media platforms**

} to identify patterns, trends, and insights.



- Web content mining can be thought of as extending the work performed by basic search engines.
    - Most search engines are keyword-based.
    - Web content mining goes beyond this basic IR technology.
    - It can improve on traditional search engines through techniques such as concept hierarchies and synonyms, user profiles, and analyzing the links between pages.
- One taxonomy of Web mining divided Web content mining into;
    - **Agent-based approaches:**
        - Software systems called agents perform the content mining.
        - Search engines belonging to this class, as do intelligent search agents, information filtering, and personalized web agents.
        - Intelligent search agents go beyond the simple search engines and use other techniques besides keyword searching to accomplish a search.
            - Eg: They may use user profiles or knowledge concerning specified domains.
        - Information filtering utilizes IR techniques, knowledge of the link structures, and other approaches to retrieve and categorize documents.
        - Personalized web agents use information about user preferences to direct their search.
    - **Database approaches**
        - The database approaches view the web data as belonging to a database.
        - There have been approaches that view the Web as a multilevel database, and there have been many query languages that target the Web.

Note:

- *One problem associated with **retrieval of data from Web documents** is that they are **not structured** as in traditional databases.*
- *There is **no schema or division into attributes.***
- *Traditionally, Web pages are defined using hypertext markup language (HTML).*
- *Web pages created using **HTML** are only **semistructured**, thus making **querying more difficult** than with well-formed databases containing schemas and attributes with defined domains.*
- *HTML ultimately will be replaced by **extensible markup language (XML)**, which will provide **structured documents** and **facilitate easier mining.***

Introduction to Crawlers:

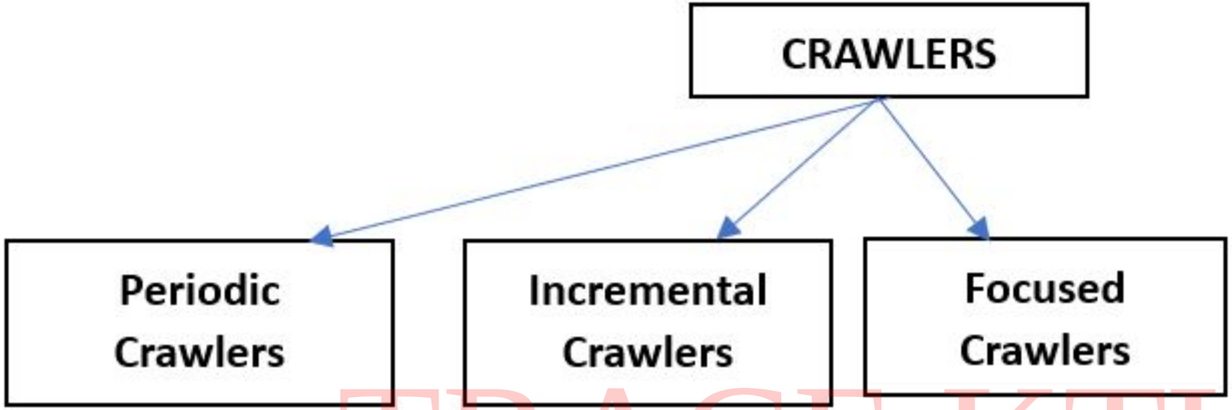
Eg: Suppose we are interested in **collecting data on customer reviews for a particular product.**

- We can build a **crawler that navigates through different e-commerce websites and collects data on customer reviews for that product.**
- The **crawler would start by visiting the first e-commerce website, and then navigating through the pages that contain the customer reviews.**
- It would then **extract the relevant data**, such as the rating, the review text, and the date of the review.
- **Once the crawler has collected data from the first website, it would move on to the next website and repeat the process.**
- **This would continue until the crawler has visited all the relevant websites and collected all the data that is needed.**
- After the crawler has collected the data, we can then **use data mining techniques to analyze the data and extract insights.**
    - Eg: We can use sentiment analysis to identify the overall sentiment of the customer reviews, or use text mining to identify common themes and topics mentioned in the reviews.

1. Crawlers:*** (or spider or robot)

- A crawler is a **program that traverses the hypertext structure in the Web.**
- The page (or set of pages) that the crawler starts with are referred to as the **seed URLs**.
- **By starting at one page, all links from it are recorded and saved in a queue.**
- **These new pages are in turn searched and their links are saved.**
- As crawlers search the Web,
  - They may **collect information about each page**, such as extract keywords and **store in indices for users of the associated search engine.**
- A **crawler may visit a certain number of pages and then stop, build an index, and replace the existing index.**
- Crawlers are used to **facilitate the creation of indices used by search engines.**
- They **allow the indices to be kept relatively up-to-date** with little human intervention.

Types of Crawlers:



CRAWLERS

Periodic Crawlers

Incremental Crawlers

Focused Crawlers

Periodic crawlers:

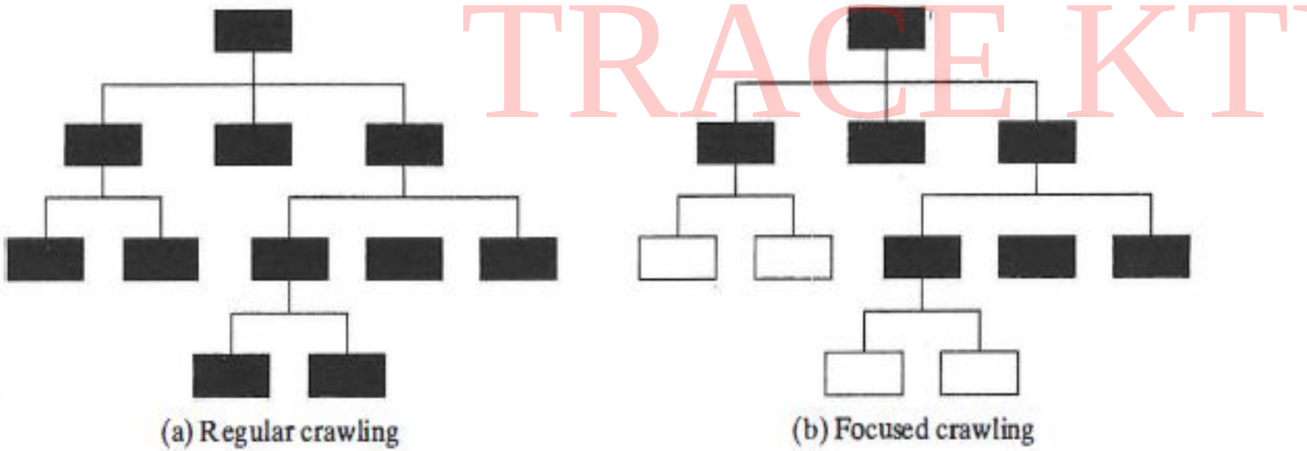
- By **starting at one page**, all **links from it are recorded and saved in a queue**.
- These **new pages are in turn searched** and their **links are saved.**
- As crawlers search the Web,
  - they may collect information about each page, such as extract keywords and store in indices for users of the associated search engine.
- A **crawler may visit a certain number of pages and then stop, build an index, and replace the existing index.**
- This type of crawler is referred to as a **periodic crawler** because it is **activated periodically.**

Incremental crawlers:

- An incremental crawler **selectively searches the Web** and only **updates the index incrementally** as opposed to replacing it.

Focused crawlers:***

- A focused crawler **visits pages related to topics of interest**.
- This concept is illustrated in the figure below.



- With focused crawling, **if it is determined that a page is not relevant or its links should not be followed, then the entire set of possible pages underneath it are pruned and not visited.**

NB: The shaded boxes represent pages that are visited.

- With focused crawling, **if it is determined that a page is not relevant or its links should not be followed, then the entire set of possible pages underneath it are pruned and not visited.**
- With thousands of focused crawlers, more of the Web can be covered than with traditional crawlers.
    - This facilitates **better scalability** as the Web grows.
- A performance objective for the focused crawler is a **high precision rate or harvest rate.**
- The focused crawler architecture consists of **three primary components;**
    - Hypertext classifier.
    - Distiller
    - Crawler

Hypertext classifier:

- Major component of the focused crawler architecture .
- Hypertext classifier **associates a relevance score for each document with respect to the crawl topic**.
- In addition, the classifier determines a **resource rating** that estimates how beneficial it would be for the crawler to follow the links out of that page.

Distiller:

- A distiller **determines which pages contain links to many relevant pages**.
  - These are called **hub-pages.**
  - These are thus **highly important pages to be visited**.
- These hub-pages may not contain relevant information, but they would be quite important to facilitate continuing the search.

Crawler:

- The crawler **performs the actual crawling on the Web**.
- The pages it visits are determined via a priority-based structure governed by the priority associated with pages by the classifier and the distiller.

2.Harvest System:

- The Harvest system is based on the use of **caching, indexing, and crawling.**
- **Harvest** is actually a set of tools that facilitate gathering of information from diverse sources.
- Indices in Harvest are topic-specific.
  - This is used to avoid the scalability problems found without this approach
- The Harvest design is centered around the use of **gatherers** and **brokers**.

Gatherer:

- A gatherer obtains information for indexing from an Internet service provider.
- Harvest gatherers use the **Essence system** to assist in collecting data.
  - Essence is a valid technique for retrieving Web documents.
  - Essence classifies documents by creating a semantic index.
  - Semantic indexing generates different types of information for different types of files and then creates indices on this information.
  - This process may first classify files based on type and then summarize the files typically based on keywords.
  - Essence uses the file extensions to help classify file types.

Broker:

- A broker provides the index and query interface.
- The relationship between brokers and gatherers can vary.
  - Brokers may interface directly with gatherers or may go through other brokers to get to the gatherers.

3.Personalization:

- Another example of Web content mining is in the area of **personalization.**
- With personalization, web access or the contents of a web page are modified to better fit the desires of the user.
- This may involve actually creating web pages that are unique per user or using the desires of a user to determine what web documents to retrieve.
- With personalization, advertisements to be sent to a potential customer are chosen based on specific knowledge concerning that customer.
- Unlike targeting, personalization may be performed on the target Web page.
- The goal here is to entice a current customer to purchase something he or she may not have thought about purchasing.
  - Eg: A common example of personalization is;
    - The use of a visitor' s name when he or she visits a page.
- Personalization is almost the opposite of targeting.
  - With targeting, businesses display advertisements at other sites visited by their users.
  - With personalization, when a particular person visits a web site, the advertising can be designed specifically for that person.

Example for Personalization;

Personalization with **My Yahoo !**

- With My Yahoo ! , a user himself personalizes what the screen looks like.
- He can provide preferences in such areas as weather, news, stock quotes, movies, and sports.
- Once the preferences are set up, each time the user logs in, his page is displayed.
- The personalization is accomplished by the user explicitly indicating what he wishes to see.
- Some observations about the use of personalization with My Yahoo! are;
  - A few users will create very sophisticated pages by utilizing the customization provided.
  - Most users do not seem to understand what personalization means and use only the default page.
- Any personalization system should be able to support both types of users.
- This personalization is not automatic, but more sophisticated approaches to personalization actually use data mining techniques to determine the user preferences.
- An automated personalization technique predicts future needs based on past needs or the needs of similar users.

- Personalization can be viewed as a type of clustering, classification, or even prediction.
    - Through classification, the desires of a user are determined based on those for the class.
    - With clustering, the desires are determined based on those users to which he or she is determined to be similar.
    - Prediction is used to predict what the user really wants to see.
- There are **three basic types of Web page personalization;**

1. Manual techniques perform personalization through user registration preferences or via the use of rules that are used to classify individuals based on profiles or demographics.

2. Collaborative filtering accomplishes personalization by recommending information (pages) that have previously been given high ratings from similar users.

3. Content-based filtering retrieves pages based on similarity between them and user profiles.

WEB STRUCTURE MINING:***

- Web structure mining can be viewed as creating a model of the Web organization or a portion thereof.
- This can be used to classify Web pages or to create similarity measures between documents.

1. PageRank***

- PageRank is used **to measure the importance of a page and to prioritize pages returned** from a traditional search engine using keyword searching.
- **Aim:**
  - To **increase the effectiveness of search engines**.
  - To **improve the efficiency.**

How is PageRank value for a page calculated ??

- It is **based on the number of pages that point to it**.
- This is actually a measure based on the number of **backlinks** to a page.

- **Backlink**:
  - Backlink is a link pointing to a page.
  - It is not simply a count of the number of backlinks.
    - A weighting is also used to provide more importance to backlinks coming from important pages.
- Given a page *p*,

The **PageRank of a page *p*** is defined as;

$$PR(p) = c \sum_{q \in B_p} \frac{PR(q)}{N_q}$$

- B_p : set of pages that point to p
- F_p :the set of links out of p.
- Here $N_q = |F_q|$
- The constant c is a value between 0 and 1 and is used for normalization.

- **Rank sink problem:**
  - The problem that exists with this PageRank calculation is that, when a cyclic reference occurs (page A points to page B and page B points to page A), the PR value for these pages increases.
  - This problem is solved by adding an additional term to the formula:

$$\mathrm{PR}'(p) = c \sum_{q \in B_p} \frac{\mathrm{PR}(q)}{N_q} + cE(v)$$

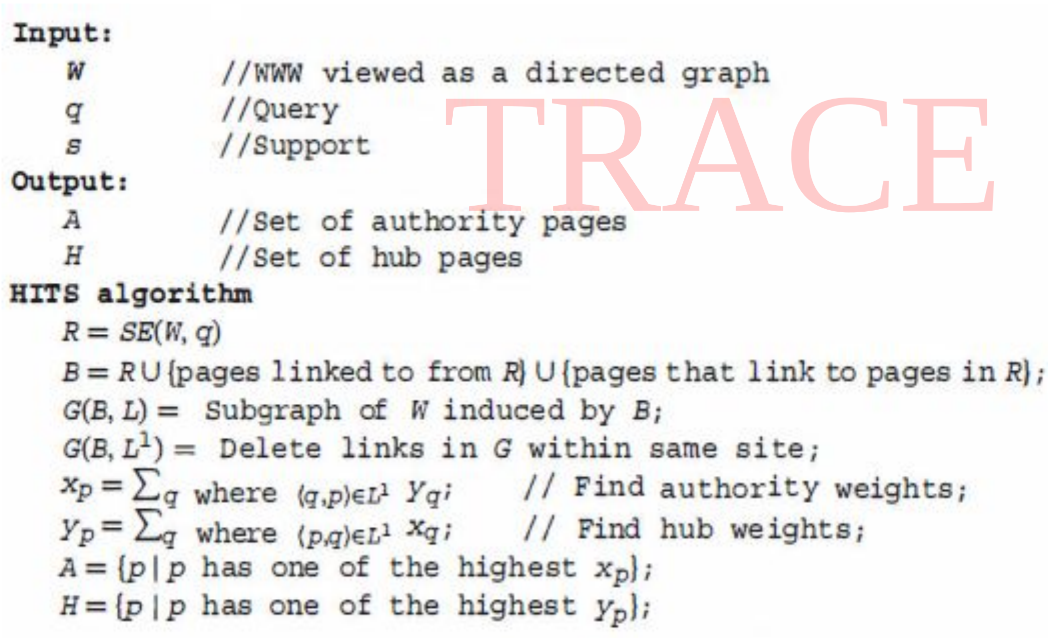
    - **c** is maximized.
    - **E(v)** is a vector that adds an artificial link.
  - This simulates a random surfer who periodically decides to stop following links and jumps to a new page.
  - E(v) adds links of small probabilities between every pair of nodes.
  - The PageRank technique is different from other approaches that look at links.
    - It does not count all links the same.
    - The values are normalized by the number of links in the page.

2.Clever ***

- Developed at IBM.
- **Aim:**
  - To **find authoritative pages and hubs.**
  - **Authoritative page:** Page which is the "best source" for the requested information.
  - **Hub page:** A page that contains links to authoritative pages.
- The Clever system **identifies authoritative pages and hub-pages by creating weights.**
- A search can be viewed as having a goal of finding the best hubs and authorities.
- Generally, sites are developed in a distributed and unsupervised manner.
- A user has no way of knowing whether the information contained within a Web page is accurate or not.
  - Anyone can produce a page that contain errors.
  - In addition, some pages might be of a higher quality than others.
  - These pages are often referred to as being the most authoritative.
  - It is important to note that authoritative pages are different from relevant pages.
  - A page may be extremely relevant, but if it contains factual errors, users certainly do not want to retrieve it.
  - The issue of authority usually does not surface in traditional IR.

Hyperlink-induced topic search: (HITS)***

- **Aim:** To **find authoritative pages and hubs.**
- The HITS technique contains two components:
    - Based on a given set of keywords (found in a query), a set of relevant pages are found.
    - Hub and authority measures are associated with these pages.
        - Pages with the highest values are returned.



```
Input:
    W          //WWW viewed as a directed graph
    q          //Query
    s          //Support
Output:
    A          //Set of authority pages
    H          //Set of hub pages
HITS algorithm
    $R = SE(W, q)$
    $B = R \cup$ {pages linked to from $R$} $\cup$ {pages that link to pages in $R$};
    $G(B, L) =$ Subgraph of $W$ induced by $B$;
    $G(B, L^1) =$ Delete links in $G$ within same site;
    $x_p = \sum_q$ where $_{(q,p) \in L^1}$ $y_q$;     // Find authority weights;
    $y_p = \sum_q$ where $_{(p,q) \in L^1}$ $x_q$;     // Find hub weights;
    $A = \{p \mid p$ has one of the highest $x_p\}$;
    $H = \{p \mid p$ has one of the highest $y_p\}$;
```

- A search engine, SE, is used to find a small set, root set(R), of pages, P, which satisfy the given query, q.
- This set is then expanded into a larger set, base set (B), by adding pages linked either to or from R.
- This is used to induce a subgraph of the Web.
    - This graph is the one that is actually examined to find the hubs and authorities.
- In the algorithm, we use the notation G(B, L) to indicate that the graph (subgraph) G is composed of vertices (pages in this case) B and directed edges or arcs (links) L.
- The weight used to find authorities, x_p and the weight used to find hubs, y_p are then calculated on G.
- Because pages at the same site often point to each other, we should not really use the structure of the links between these pages to help find hubs and authorities.
- The algorithm therefore removes these links from the graph.
- Hubs should point to many good authorities, and authorities should be pointed to by many hubs.
- This observation is the basis for the weight calculations shown in the algorithm.

WEB USAGE MINING:

- Web usage mining performs mining on Web usage data, or **Web logs.**
- A Web log is a listing of page reference data.
  - It is also known as **clickstream data** because each entry corresponds to a mouse click.
- These logs can be examined from either a **client perspective** or a **server perspective.**
  - When evaluated from a server perspective, mining uncovers information about the sites where the service resides.
    - It can be used to improve the design of the sites.
  - By evaluating a client's sequence of clicks, information about a user (or group of users) is detected.
    - This could be used to perform prefetching and caching of pages.

Applications of Web usage mining:

- **Personalization** for a user can be achieved by keeping track of previously accessed pages.
    - These pages can be used to identify the typical browsing behavior of a user and subsequently to predict desired pages.
- By determining frequent access behavior for users, **needed links can be identified** to improve the overall performance of future accesses.
- Information concerning frequently accessed pages can be used for **caching.**
- Web usage patterns can be used to **gather business intelligence to improve sales and advertisement.**
- Identifying common access behaviors can be used to **improve the actual design of Web pages and to make other modifications to the site.**

Eg: Suppose that visitors to an e-commerce site can be identified as customers or non-customers.

- The behavior of customers can be compared with that for those who do not purchase anything.
- This can be used to identify changes to the overall design. It may be determined that many visitors never get past a particular page.
- That target page can be improved in an attempt to turn these visitors into customers.

Web usage mining actually consists of **three types of activities:**

- **Preprocessing:**
  - Involves reformatting the Web log data before processing.
- **Pattern discovery:**
  - Looking to find hidden patterns within the log data.
- **Pattern analysis**:
  - Process of looking at and interpreting the results of the discovery activities.

1. Preprocessing:

- The Web usage log probably is not in a format that is usable by mining applications.
- As with any data to be used in a mining application, the data may need to be reformatted and cleansed.
- Steps of preprocessing:
  - Cleansing
  - User identification
  - Session identification
  - Path completion
  - Formatting

Log:

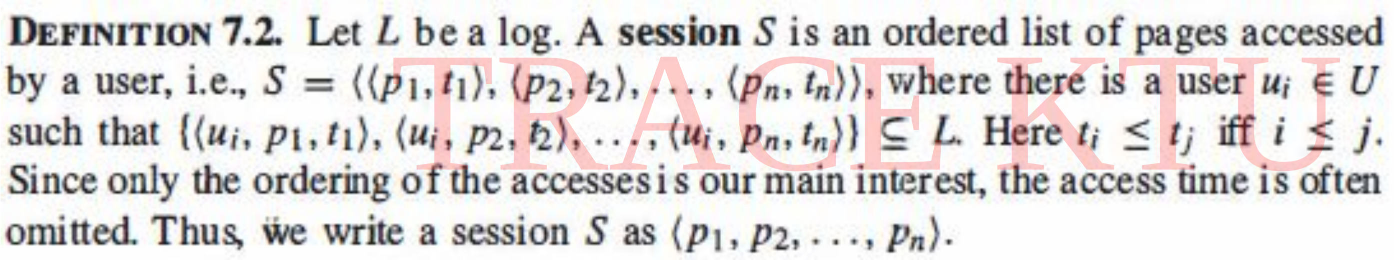
- Let P be a set of literals, called pages or clicks, and U be a set of users.
- A log is a set of triples $\{\langle u_1,p_1,t_1\rangle\},\ldots\ldots,\{\langle u_n,p_n,t_n\rangle\}$ where u_i ε U, p_i ε P and t_i is a timestamp.

- Standard log data consist of the source site, destination site, and timestamp.
    - The source and destination sites could be listed as a URL or an IP address.
    - The above definition assumes that the source site is identified by a user ID and the destination site is identified by a page ID.
    - Additional data such as Web browser information also may be included.
- Before processing the log, the data may be changed in several ways.
    - For security or privacy reasons, the page addresses may be changed into unique (but non-identifying) page identifications (such as alphabetic characters).
        - This conversion also will save storage space.
    - The data may be cleansed by removing any irrelevant information.
        - Eg: The log entries with figures (gif, jpg, etc.) can be removed.

- For a server site, a common technique is to divide the log records into sessions.

Session:

- A session is a set of page references from one source site during one logical period.
- Historically, a session would be identified by a user logging into a computer, performing work, and then logging off.
- The login and logoff represent the logical start and end of the session.



DEFINITION 7.2. Let L be a log. A **session** S is an ordered list of pages accessed by a user, i.e., $S = \langle\langle p_1, t_1\rangle, \langle p_2, t_2\rangle, \ldots, \langle p_n, t_n\rangle\rangle$, where there is a user $u_i \in U$ such that $\{\langle u_i, p_1, t_1\rangle, \langle u_i, p_2, t_2\rangle, \ldots, \langle u_i, p_n, t_n\rangle\} \subseteq L$. Here $t_i \leq t_j$ iff $i \leq j$. Since only the ordering of the accesses is our main interest, the access time is often omitted. Thus, we write a session S as $\langle p_1, p_2, \ldots, p_n\rangle$.

- Associated with each session *i* s a unique identifier, which is called a session ID.
- The length of a session S is the number of pages in it, which is denoted as len(S).

- A major problem associated with the preprocessing activity is the **correct identification of the actual user.**
- User identification is complicated by the use of proxy servers, client side caching, and corporate firewalls.
- **Tracking who is actually visiting a site (and where they come from) is difficult.**
- Even though a visit to a Web page will include a source URL or IP address that indicates the source of the request, this may not always be accurate in determining the source location of the visitor.
- Users who access the Internet through an Internet service provider (ISP) will all have the source location of that provider.
- It is not unique to the individual.
- In addition, the same user may use different ISPs.
- Also, there will be many users accessing the Web at the same time from one machine.
- **Cookies** can be used to assist in identifying a single user regardless of machine used to access the Web.
    - A cookie is a file that is used to maintain client-server information between accesses that the client makes to the server.
    - The cookie file is stored at the client side and sent to the server with each access.

Path completion:

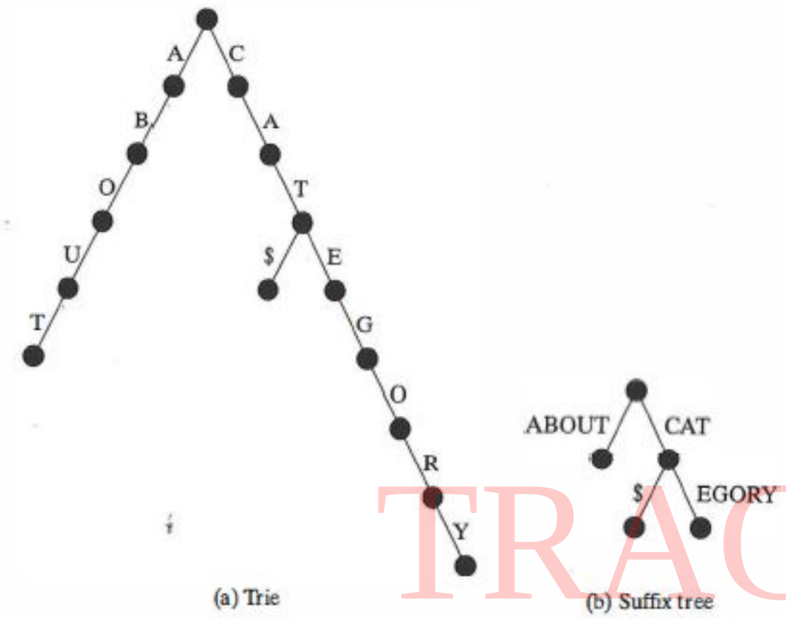
- It is an attempt to add page accesses that do not exist in the log but that actually occurred.
- Some missing pages can be easily added.
- Eg: If a user visits page A and then page C, but there is no link from A to C, then at least one page in this path is missing.
- Algorithms are used both to infer missing pages and to generate an approximate timestamp.

2.Data Structures:

- To keep track of patterns identified during the Web usage mining, several unique data structures have been proposed.
- A common data structure that is used for this is a **trie**.

Trie:

- A trie is a rooted tree, where each path from the root to a leaf represents a sequence.
- Tries are used to store strings for pattern-matching applications.
- Each character in the string is stored on the edge to the node.
- Common prefixes of strings are shared.
- A problem in using tries for many long strings is the space required.



(a) Trie

(b) Suffix tree

<u>NB:</u>

- *In both trees there is an extra edge labeled "$."*
- *This symbol (or any symbol that is not in the alphabet and is used to construct the strings) is added to ensure that a string that is actually a prefix of another (CAT is a prefix of CATEGORY) terminates in a leaf node.*

- Figure(a) shows a standard trie for the three strings;

  {ABOUT, CAT, CATEGORY}.

- There are many nodes with a degree of one.
- This is a waste of space.
- This is solved by compressing nodes together when they have degrees of one.
- Figure (b) shows a compressed version of this trie.
- Here a path consisting of nodes with single children is compressed to one edge.
- The compressed trie is called a **suffix tree.**

Suffix tree - Characteristics:

- Each internal node except the root has at least two children.
- Each edge represents a nonempty subsequence.
- The subsequences represented by sibling edges begin with different symbols.

Suffix tree - Uses:

- Suffix tree helps to find any subsequence in a sequence.
- It also helps to find the common subsequences among multiple sequences.

3. Pattern Analysis:

- Once patterns have been identified, they must be analyzed to determine how that information can be used.
- Some of the generated patterns may be deleted and determined not to be of interest.
- Nowadays, web logs are not only used to identify frequent types of traversal patterns, but also to identify patterns that are of interest.

g-sequence:

- **g-sequence is a vector that consists not only of the pages visited (events) but also of wildcards.**

  Eg: The g-sequence b * c stands for;

  A sequence consisting of b, any number of pages, then c.

  - With the use of wildcards, it is indicated that the events need not be contiguous.
- An important example to application of pattern analysis is , *comparing the differences between traversal patterns of the customers of an e-business site and those that are not customers.*
  - Visitors to a site have been classified as;
    - short-time visitors, active investigators, and customers.
- Firstly, pre-processing filters out the visitors who are short-time.
- Using concept hierarchies, the contents of the Web pages are then abstracted to more general concepts.
- The log is then divided into those for customers and those for non-customers.
- Each log is then examined to find patterns based on any desired requirements (Eg:frequency).
- The patterns found across the two logs are then compared for similarity.
- Similarity is determined using the following rule:

Two patterns are comparable if their g-sequences have at least the first n pages the same.Here, n is supplied by the user.