Unit III

CHAPTER 3

Web Search Engine

Syllabus

Web Search Engine : Web search overview, web structure, the user, paid placement, search engine optimization/spam, Web size measurement, search engine optimization/spam, Web Search Architectures.

XML retrieval : Basic XML concepts, Challenges in XML retrieval, A vector space model for XML retrieval, Evaluation of XML retrieval, Text-centric versus data-centric XML retrieval.

Syllabus Topic : Web Search Overview

3.1 Web Search Overview

3.1.1 Background and History

Q. 3.1.1 Define web search engine and explain basic operation of web search engine. (Ref. Sec. 3.1.1) **(3 Marks)**

- The first basic search engine was created by three computer science students at McGill University in 1990. The tool was named Archie (a play on the word "Archive").
- Archie created a searchable database of public file names on FTP (File Transfer Protocol) sites. Since the tool was launched before the Internet took off on a large-scale basis, the database was small enough to be searched manually.

- Following Table 3.1.1 shows development of search engine :

Table 3.1.1

Search Engine Name	Development Year
Archie	1990
Lycos	1993
Yahoo	1994
WebCrawler	1994
AltaVista	1995
Looksmart	1995
WiseNut	2001
Excite	1995
Hotbot	1996
Dogpile	1996
Google	1996
MSN Search	1998
ASK	1996
Teoma	2000
Infoseek	1994
Overture	1998
Alltheweb	1999
AOL Search	1999

- The basic operation of web search engine is as follows :
  - A client (such as a browser) sends an http request to a web server. The browser specifies a URL such as, http://www.google.com/contact/index.html.
  - In this example of URL, the string http refers to the protocol to be used for transmitting the data. The string www.google.com is known as the domain and specifies the root of a hierarchy of web pages.
- The mass publishing of information on the Web is essentially useless unless this wealth of information can be discovered and consumed by other users. Early attempts at making web information "discoverable" fell into two broad categories :

  (1) Full-text index search engines such as Altavista, Excite and Infoseek.

  (2) Taxonomies populated with web pages in categories, such as Yahoo!

3.1.2 Introduction to Web Search Engine

Q. 3.1.2 Explain steps performed by search engine. (Ref. Sec. 3.1.2) **(3 Marks)**

- Search Engine consists of number of databases, web pages, newsgroups, programs, images etc. It helps to locate information on World Wide Web.
- User can search for any information by passing query in form of keywords or phrase. It then searches for relevant information in its database and return to the user.
- Web crawler, database and the search interface are the major component of a search engine that actually makes search engine to work. Search engines make use of Boolean expression AND, OR, NOT to restrict and widen the results of a search.
- Following are the steps that are performed by the search engine :
  1. The search engine looks for the keyword in the index for predefined database instead of going directly to the web to search for the keyword.
  2. It then uses software to search for the information in the database. This software component is known as web crawler.

3. Once web crawler finds the pages, the search engine then shows the relevant web pages as a result. These retrieved web pages generally include title of page, size of text portion, first several sentences etc.
4. These search criteria may vary from one search engine to the other. The retrieved information is ranked according to various factors such as frequency of keywords, relevancy of information, links etc.
5. User can click on any of the search results to open it.

3.1.3 Characteristics of Web Search Engine

Q. 3.1.3 Explain characteristics of web search engine. (Ref. Sec. 3.1.3) **(5 Marks)**

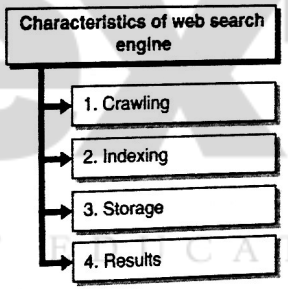

Fig. 3.1.1 : Characteristics of web search engine

➜ **1. Crawling**

- It finds website addresses and the contents of a website for storage in the search engine database and collects large amounts of information simultaneously.
- It scans brand new information on the Internet or it can locate older data.

➜ **2. Indexing**

Once the search engine has crawled the contents of the Internet, it indexes that content based on the occurrence of keyword phrases in each individual website.

➜ **3. Storage**

Storing web content within the database of the search engine is essential for fast and easy searching.

➜ **4. Results**

- Results are the hyperlinks to websites that show up in the search engine page when a certain keyword or phrase is queried.
- When you type in a search term, the crawler runs through the index and matches what you typed with other keywords.
- Algorithms created by the search engine designers are used to provide the most relevant data first. Each search engine has its own set of algorithms and therefore returns different results.

3.1.4 Challenges issue in Web Search Engine

Q. 3.1.4 What are various challenges in web search engine ? (Ref. Sec. 3.1.4) **(5 Marks)**

- Users are unfamiliar with logical view of data while searching.
- Users are unfamiliar with search engine interfaces.
- Many different kinds of users.
- Distributed and hidden data.
- Exponential growth of Web pages.
- Unstructured and redundant data.
- Unedited and volatile data.

Syllabus Topic : Web Structure

3.2 The Structure of the Web

Q. 3.2.1 Describe structure of web. (Ref. Sec. 3.2) **(5 Marks)**

The structure of web consists of following entities :

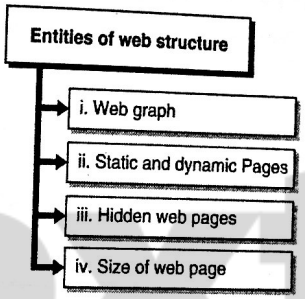

Fig. 3.2.1 : Entities of web structure

➜ **(i) Web graph**

- The relationship between sites and pages indicated by hyperlinks gives rise to what is called a ***Web graph***.
- When it is viewed as a purely mathematical object, each page forms a node in this graph and each hyperlink forms a directed edge from one node to another.
- Pages and links are continuously added and removed at sites around the world; it is never possible to capture more than a rough approximation of the entire Web graph.

➜ **(ii) Static and dynamic pages**

- Web pages are described as "static" or "dynamic". Static page can be generated in advance of any request, placed on disk, and transferred to the browser or Web crawler on demand.

- For example, home page. A dynamic Web page is assumed to be generated at the time the request is made, with the contents of the page partially determined by the details of the request. Many of dynamic Web pages are not suitable for crawling and indexing.

➜ **(iii) Hidden web pages**

Many pages are part of the so-called "hidden" or "invisible" or "deep" Web. This hidden Web includes pages that have no links referencing them, those that are protected by passwords, and those that are available only by querying a digital library or database.

➜ **(iv) Size of web page**

Web page size is an important factor to measure as the bigger a page is, the longer it takes to download the required resources to display it.

Syllabus Topic : The User

3.3 Users

Q. 3.3.1 Define user and explain different types of queries used by user. (Ref. Sec. 3.3) **(5 Marks)**

- Search engine is a provision that allows Internet users to search the data via the World Wide Web (WWW). A user enters keywords or key phrases into a web search engine and receives a list of Web content results in the form of websites, images, videos or other online data.
- A user issuing an following queries like :

(i) Informational query

(ii) Transactional query

(iii) Navigational query

➜ **(i) Informational query**

Users employ informational searches whenever they need guidance, background information, or specific information about a topic or product without having any concrete intention to purchase or any wish to seek out a certain landing page.

➜ **(ii) Transactional query**

- Transactional search queries have their own specific patterns and formats. These include not only verbs but also specific names.
- Furthermore, though it may not be made explicit, there are some searches that imply a transactional intent.

➜ **(iii) Navigational query**

- Navigation-oriented searches are targeting a certain kind of website.
- The user is not seeking a particular product or company and has no clear intent to purchase.

Syllabus Topic : Paid Placement

3.4 Paid Placement

Q. 3.4.1 Write short note on paid placement. (Ref. Sec. 3.4) **(5 Marks)**

- Pay for placement is an Internet advertising model in which advertisements appear along with relevant search results from a Web search engine.
- Under this model, advertisers bid for the right to present an advertisement with specific search terms (i.e., keywords) in an open auction.
- When one of these keywords is entered into the search engine, the results of the auction on that keyword are presented, with higher-ranking bids appearing more prominently on the page.

Syllabus Topic : Search Engine Optimization / SPAM

3.5 Search Engine Optimization / SPAM

Q. 3.5.1 What is search engine optimization? (Ref. Sec. 3.5) **(5 Marks)**

- Search Engine Optimization (SEO) is the activity of optimizing web pages or whole sites in order to make them search engine friendly, thus getting higher positions in search results.
- SEO is all about optimizing a website for search engines.
- SEO is a technique for :
  - Designing and developing a website to rank well in search engine results.
  - Improving the volume and quality of traffic to a website from search engines.
  - Marketing by understanding how search algorithms work, and what human visitors might search.
- There are two types of Search engine optimization techniques :

1. White Hat SEO	2. Black Hat SEO

➜ **1. White Hat SEO**

It Improves search engine ranking of a websites.

➜ **2. Black Hat SEO**

- It exploits weaknesses in the search engine algorithms to obtain high rankings for a website.
- There are two ways of optimization :

1. On-Page SEO	2. Off-Page SEO

➜ **1. On-Page SEO**

It includes providing good content, good keywords selection, putting keywords on correct places, giving appropriate title to every page, etc.

➜ **2. Off-Page SEO**

It includes link building, increasing link popularity by submitting open directories, search engines, link exchange, etc.

3.5.1 SPAM

Q. 3.5.2 Explain SPAM in detail. (Ref. Sec. 3.5.1) **(5 Marks)**

- It is an Intentional attempts to manipulate search engine rankings for specific keywords or keyword phrase queries.
- Pages that use web spam to improve search engine results page (SERP) rankings.

There are essentially two types of spamming : boosting and hiding.

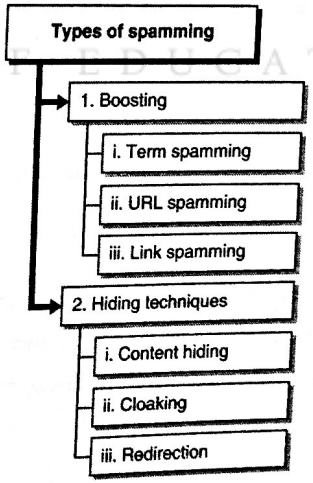

Fig. 3.5.1 : Types of spamming

➜ **1. Boosting**

This is when one takes an action intended to increase or boost the value of a page.

➜ **(i) Term Spamming**

Manipulate data through elements such as the page TITLE (title spam), Meta Description or Meta Keywords.

➜ **(ii) URL Spamming**

Some search engine give weightage to the URL. It can be considered to be manipulation.

➜ **(iii) Link Spamming**

It is another well-known one that also includes anchor text spamming. When spammers seek to drop links on pages to increase a target pages value and obviously the more infamous hack and drop techniques.

➜ **2. Hiding techniques**

This technique issued when one is using not generally noticeable methods of getting a page to rank higher or hiding of boosting techniques.

➜ **(i) Content hiding**

These are techniques where terms and links are hidden when the browser renders a page.

➜ **(ii) Cloaking**

This is when one identifies a search engine crawler and seeks to show a different version of the page to the spider than it would for the average user. This, one assumes, cuts down on the changes of being reported by users or competitors that might otherwise see the spammy page.

➜ **(iii) Redirection**

The page is automatically redirected by the browser in the same manner so that the page gets indexed by the engine, but the user will never actually see it.

Syllabus Topic : Web Size Measurement

3.6 Web Size Measurement

Q. 3.6.1 Write note on web size measurement. (Ref. Sec. 3.6) **(5 Marks)**

- Some hosts contain millions of pages with valuable content and others contain millions of pages with little or no useful content.
- Indexable pages (those pages that should be considered for inclusion in a general-purpose Web search engine) would involve all those pages that could have a substantial impact on search results.
- If we may assume that any page included in the index of a major search engine forms part of the indexable Web, a lower bound for the size of the indexable Web may be determined from the combined coverage of the major search engines.
- if the sets A1,A2, . . .represent the sets of pages indexed by each of these search engines, a lower bound on the size of the indexable Web is the size of the union of these sets I UAil.
- Unfortunately, it is difficult to explicitly compute this union. Major search engines do not publish lists of the pages they index, or even provide a count of the number of pages, although it is usually possible to check if a given URL is included in the index.
- Even if we know the number of pages each engine contains, the size of the union will be smaller than the sum of the sizes because there is considerable overlap between engines.

- Following techniques describe the estimating he combined coverage of major search engines :

1. First, a test set of URLs is generated. This step may be achieved by issuing a series of random queries to the engines and selecting a random URL from the results returned by each. We assume that this test set represents a uniform sample of the pages indexed by the engines.

2. Each URL from the test set is checked against each engine and the engines that contain it are recorded.

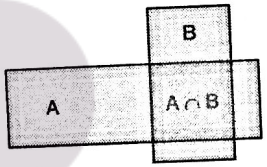

Fig. 3.6.1 : The collection overlap between search engines A and B can be used to estimate the size of the indexable Web

- Given two engines, A and B, the relationship between their collections is illustrated by Fig. 3.6.1.

- Sampling with URLs allows to estimate Pr[A ∩ B|A], the probability that a URL is contained in the intersection if it is contained in A :

$$\Pr[A \cap B \mid A] = \frac{\text{\# of test URLs contained in both A and B}}{\text{\# of test URLs contained in A}}$$

- We may estimate Pr[A ∩ B|B] in a similar fashion. If we know the size of A, we may then estimate the size of the intersection as $|A \cap B| = |A| \cdot \Pr[A \cap B|A]$ and the size of B as,

$$|B| = \frac{|A \cap B|}{\Pr[A \cap B \mid B]}$$

Thus, the size of the union may be estimated as,

$$|A \cup B| = |A| + |B| - |A \cap B|$$

- If sizes for both A and B are available, the size of the intersection may be estimated from both sets, and the average of these estimates may be used to estimate the size of the union.

$$|A \cup B| = |A| + |B| - \frac{1}{2}(|A| \cdot \Pr[A \cap B \mid A] + |B| \cdot \Pr[A \cap B \mid B])$$

Syllabus Topic : Web Search Architecture

3.7 Web Search Architecture

Q. 3.7.1	What are the various components of web search architecture? (Ref. Sec. 3.7)	**(5 Marks)**
Q. 3.7.2	Explain two major functions of search engine. (Ref. Sec. 3.7)	**(5 Marks)**

- Architecture is designed to ensure that a system will satisfy the application requirements or goals. The two primary goals of a search engine are :

1. **Effectiveness (quality) :** We want to be able to retrieve the most relevant set of documents possible for a query.

2. **Efficiency (speed) :** We want to process queries from users as quickly as possible.

- The web search architecture consist of following components :

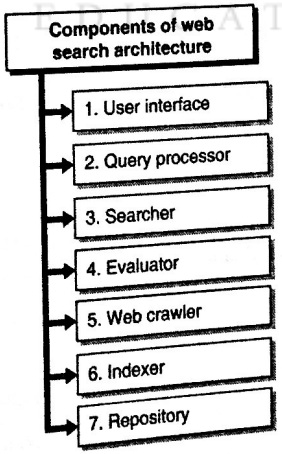

Fig. 3.7.1 : Components of web search architecture

1. User interface

User can pass the query through this interface. Queries can be in the form of informational, navigational and transactional etc.

2. Query processor

User query passed to the query processor and compare this to indexed document and return the most relevant document. It performs following tasks :

- Generate token and parsing.
- Remove less important word.
- Query expansion.

3. Searcher

Searches given query in the databases.

4. Evaluator

Evaluate the performance of accessing the web page.

5. Web Crawler

It is program that browses the web in a systematic and automated manner order to provide correct data in the form of web document.

6. Indexer

Data about web pages are stored in index database.

7. Repository

The web pages downloaded from web are compressed and stored in local repository of search engine for storage.

Search engine components support two major functions

1. The index process
2. The query process

1. The index process

Building data structures that enable searching.

2. The query process

Using these data structures to produce a ranked list of documents for a user's query

3.7.1 Index Process

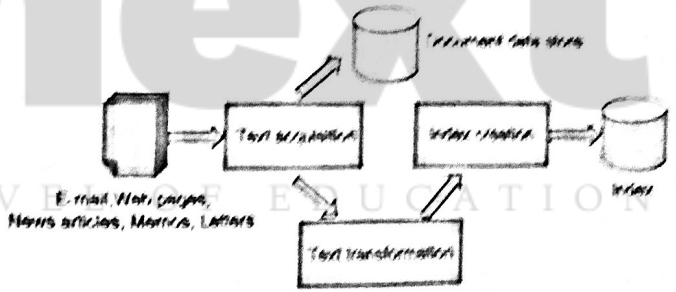

Fig. 3.7.2

Indexing process comprises of the following three tasks :

1. Text acquisition
2. Text transformation
3. Index creation

➜ **1. Text acquisition**

It identifies and stores documents for indexing.

➜ **2. Text transformation**

It transforms document into index terms or features.

➜ **3. Index creation**

It takes index terms created by text transformations and create data structures to support fast searching.

3.7.2 Query Process

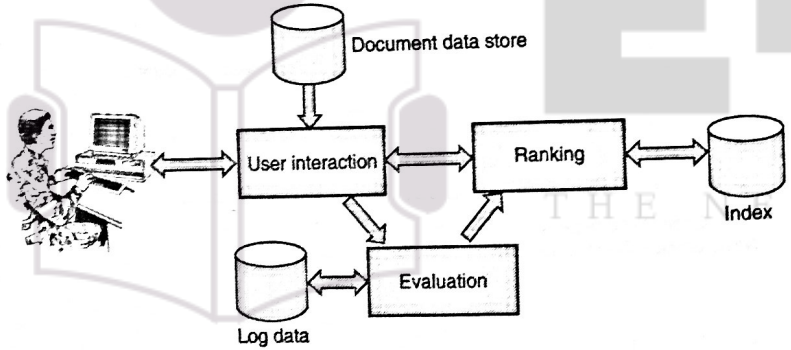

Fig. 3.7.3

Query process comprises of the following three tasks :

1. User interaction
2. Ranking
3. Evaluation

➜ **1. User interaction**

It support creation and refinement of user query and displays the results.

➜ **2. Ranking**

It uses query and indexes to create ranked list of documents.

➜ **3. Evaluation**

It monitors and measures the effectiveness and efficiency. It is done offline.

Syllabus Topic : Basic XML Concepts

3.8 Basic Concepts of XML

Q. 3.8.1 Write short note on basic concepts of XML. (Ref. Sec. 3.8) **(5 Marks)**

- XML (Extensible Markup Language) is a framework for defining markup languages.
- There is no fixed collection of markup tags.
- XML designed to be self-descriptive to store and transport data.
- XML languages simplifies data share and data transport features.
- Enables building of generic tools.

Example

```
<product id="p01">
   <prodname>AC</ prodname >
   <para>This is Air Conditioner product
   <tm>AC</tm>Cooling
   </para>
</product>
```

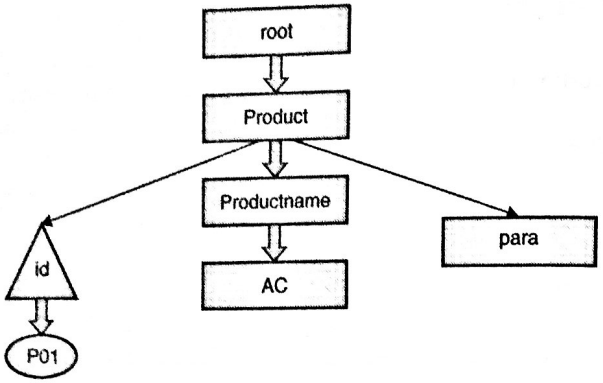

Fig. 3.8.1

3.8.1 Basic Structure

- An XML document is a basic unit of XML information composed of elements and other markup in an orderly package.
- An XML document is an ordered and labeled tree.
- An XML document can contains wide variety of data.
- Character data leaf nodes contain the actual data (text strings).
- Element nodes are labelled with a name (often called the element type), and a set of attributes, each consisting of a name and a value. It can have child nodes.
- Document Elements are the building blocks of XML. The document is divided into a hierarchy of sections, each serving a specific purpose.
  - Elements are denoted by markup tags.
  - Syntax : <element name attribute1 attribute2>

    ----content

    </element>

where, element-name is the name of the element. The name its case in the start and end tags must match.

- attribute1, attribute2 are attributes of the element separated by white spaces. An attribute defines a property of the element. It associates a name with a value, which is a string of characters.

3.8.2 Why Use XML?

- Represent semi-structured data.
- Data that are structured, but don't fit relational model.
- XML is more flexible than DBs.
- XML is more structured than simple IR.
- You get a massive infrastructure for free.

3.8.3 Schemas

- XML Schema is commonly known as XML Schema Definition (XSD). It is used to describe and validate the structure and the content of XML data.
- XML schema defines the elements, attributes and data types. Schema element supports Namespaces. It is similar to a database schema that describes the data in a database.

Syllabus Topic : Challenges in XML Retrieval

3.9 Challenges in XML Retrieval

Q. 3.9.1 Define XML and challenges in XML retrieval. (Ref. Sec. 3.9) **(5 Marks)**

First challenge document parts to retrieve

Users want us to return parts of documents (i.e., XML elements), not entire documents as IR systems usually do in unstructured retrieval. for this problem use structured document retrieval principle : ***A system should always retrieve the most specific part of a document answering the query.***

Second challenge : document parts to index : There are four approaches

1. Group nodes into non-overlapping pseudo documents.
2. **Top down processing :** Start with one of the latest elements as the indexing unit then, post process search results to find for each sub element that is the best hit. This two-stage retrieval process often fails to return the best sub element because the relevance of a whole book is often not a good predictor of the relevance of small sub elements within it.
3. **Bottom up Processing :** Instead of retrieving large units and identifying sub elements (top down), we can search all leaves, select the most relevant ones and then extend them to larger units in post processing. Similar problem as top down. the relevance of a leaf element is often not a good predictor of the relevance of elements it is contained in.
4. Indexing all elements means that search results will be highly redundant.

Third challenge : nested elements

Remove nested elements in a post processing step to reduce redundancy. Collapse several nested elements in the results list and use highlighting of query terms to draw the user's attention to the relevant passages.

Sylabus Topic : A Vector Space Model for XML Retrieval

3.10 A Vector Space Model for XML Retrieval

3.10.1 Explain vector space model for XML retrieval. (Ref. Sec. 3.10)

- Vector space model is an algebraic model for representing text documents as vectors of identifiers.
- It is used in information filtering, information retrieval, indexing and relevancy rankings.
- Vector space model is used to encode a word together with its position within the XML tree.

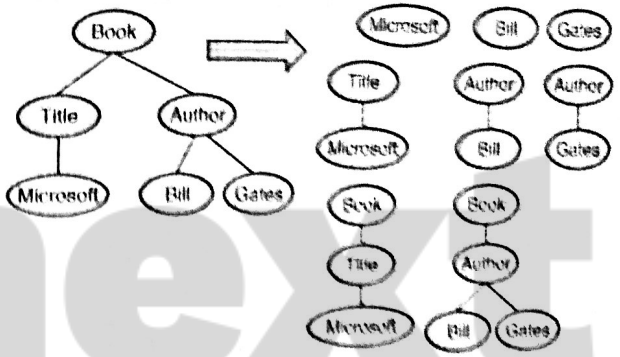

Fig. 3.10.1

- In above example first take each text node and break it into multiple nodes, one for each word. So the leaf node Bill Gates is split into two leaves Bill and Gates.
- Next we define the dimensions of the vector space to be lexicalized subtrees of documents subtrees that contain at least one vocabulary term.
- Now represent queries and documents as vectors in this space of lexicalized subtrees and compute match between them.
- This means that we the vector space formalism for XML retrieval.
- The main difference is that the dimensions of vector space in unstructured retrieval are vocabulary terms whereas they are lexicalized subtrees in XML retrieval.
- No matter that match the query structure related by inserting from additional nodes. We must that retrieval results respect this preference by computing a weight for each match.

- A simple measure of the similarity of a path c_q in a query and a path c_d in a document is the following context resemblance function CR :

$$C_R(c_q, c_d) = \begin{cases} \frac{1 + |c_q|}{1 + |c_d|}, & \text{if } c_q \text{ matches } c_d \\ 0, & \text{if } c_q \text{ does not match } c_d \end{cases}$$

- Where, $|c_q|$ and $|c_d|$ are the number of nodes in the query path and document path, respectively, and c_q matches c_d iff we can transform c_q into c_d by inserting additional nodes.

- Context resemblance Example :

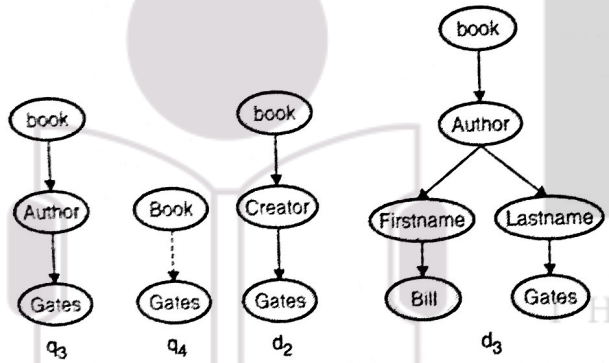

Fig. 3.10.2

$$C_R(c_q, c_d) = \begin{cases} \frac{1 + |c_q|}{1 + |c_d|}, & \text{if } c_q \text{ matches } c_d \\ 0, & \text{if } c_q \text{ does not match } c_d \end{cases}$$

$C_R(c_q, c_d) = 3/4 = 0.75$. The value of $C_R(c_q, c_d)$ is 1.0 if q and d are identical

- The final score for a document is computed as a variant of the cosine measure which we call SIMNOMERGE. SIMNOMERGE is defined as follows :

$$\text{SIMNOMERGE}(q, d) = \frac{\sum_{c_k \in B} \sum_{c_l \in B} C_R(c_k, c_l) \sum_{t \in V} \text{weight}(q, t, c_k) \text{ weight}(d, t, c_l)}{\sqrt{\sum_{c \in B, t \in V} \text{weight}^2(d, t, c)}}$$

- Where, V is the vocabulary of non-structural terms; B is the set of all XML contexts; and weight (q, t, c) and weight (d, t, c) are the weights of term t in XML context c in query q and document d respectively.

- SCORE DOCUMENTS WITH SIMNOMERGE (q, B, V, N, normalizer)

The algorithm for scoring documents with is as follows :

1. for n ← 1 to N
2. do score[n] ← 0
3. for each $(c_q, t) \in q$
4. do $w_q \leftarrow$ WEIGHT (q, t, c_q)
5. for each $c \in B$
6. do if $C_R(c_q, c) > 0$
7. then postings ← GETPOSTINGS (< c, t >)
8. for each posting ∈ postings
9. do x ← $C_R(c_q, c) * w_q *$ weight (posting)
10. score[docID(posting)] + = x
11. for n ← 1 to N
12. do score[n] ← score[n] / normalizer [n]
13. return score

Syllabus Topic : Evaluation of XML Retrieval

3.11 Evaluation of XML Retrieval

Q. 3.11.1 Write difference between Text-centric and Data-centric XML Retrieval. (5 Marks)

(Ref. Sec. 3.11)

- Documents contain a mixture of textual, multimedia, and metadata information. One way to format this mixed content is according to the Extensible Mark-up Language (XML).
- The INitiative for the Evaluation of XML retrieval (INEX) was set up in 2002 to establish an infrastructure and provide means, in the form of large test collections and appropriate scoring methods, for evaluating the effectiveness of content-oriented XML retrieval systems.
- The relevance of documents is judged by human assessors.
- Two types of information needs or in INEX Two types :
  - **Content-only or CO topics :** regular keyword queries as in unstructured information retrieval.
  - **Content-and-structure or CAS topics :** have structural constraints in addition to keywords.
- Since, content-and-structure queries have both structural and content criteria, relevance assessments are more complicated than in unstructured retrieval.

INEX relevance assessments

- The component coverage dimension evaluates whether the element retrieved is "structurally" correct, i.e., neither too low nor too high in the tree.

- We distinguish four cases :

1. Exact coverage (E)
2. Too small (S)
3. Too large (L)
4. No coverage (N)

➜ **1. Exact coverage (E)**

The information sought is the main topic of the component and the component is a meaningful unit of information.

➜ **2. Too small (S)**

The information sought is the main topic of the component, but the component is not a meaningful (self-contained) unit of information.

➜ **3. Too large (L)**

The information sought is present in the component, but is not the main topic.

➜ **4. No coverage (N)**

The information sought is not a topic of the component.

- The **topical relevance** dimension also has four levels : highly relevant (3), fairly relevant (2), marginally relevant (1) and nonrelevant (0).

$$Q(\text{rel, cov}) = \begin{cases} 1.00 \text{ if (rel, cov)} = 3E \\ 0.75 \text{ if (rel, cov)} \in \{2E, 3L\} \\ 0.50 \text{ if (rel, cov)} \in \{1E, 2L, 2S\} \\ 0.25 \text{ if (rel, cov)} \in \{1S, 1L\} \\ 0.00 \text{ if (rel, cov)} = 0N \end{cases}$$

- This evaluation scheme takes account of the fact that binary relevance judgments, which are standard in unstructured IR, are not appropriate for XML retrieval.
- The quantization function Q does not impose a binary choice relevant/nonrelevant and instead allows us to grade the component as partially relevant. The number of relevant components in a retrieved set A of components can then be computed as :

$$\#\text{ relevant items retrieved} = \sum_{c \in A} Q(\text{rel}(c), \text{cov}(c))$$

INEX evaluation measures

As an approximation, the standard definitions of precision and recall can be applied to this modified definition of relevant items retrieved, with some subtleties because we sum graded as opposed to binary relevance assessments.

Syllabus Topic : Text-centric Versus Data-centric XML Retrieval

3.12 Text-centric Versus Data-centric XML Retrieval

Q. 3.12.1 Write short note on text-centric versus data-centric XML retrieval. (Ref. Sec. 3.12) **(5 Marks)**

Data-centric XML

- Used for messaging between enterprise applications :
  - Mainly a recasting of relational data.
  - Numerical and non-text data dominate.
  - Mostly stored in the databases.

Text-centric XML

- Used for annotating content.
  - Rich in text.
  - Demands good integration of text retrieval functionality.
- Queries are user information needs. E.g., give me the Section (element) of the document that tells me how to change a brake light.

3.13 Exam Pack (Review Questions)

Syllabus Topic : Web Search Overview

Q. 1 Define web search engine and explain basic operation of web search engine. (Refer Section 3.1.1) **(3 Marks)**

Q. 2 Explain steps performed by search engine. (Refer Section 3.1.2) **(3 Marks)**

Q. 3 Explain characteristics of web search engine. (Refer Section 3.1.3) **(5 Marks)**

Q. 4 What are various challenges in web search engine? (Refer Section 3.1.4) **(5 Marks)**

Syllabus Topic : Web Structure

Q. 5 Describe structure of web. (Refer Section 3.2) **(5 Marks)**

Syllabus Topic : The User

Q. 6 Define user and explain different types of queries used by user. (Refer Section 3.3) **(5 Marks)**

Syllabus Topic : Paid Placement

Q. 7 Write short note on paid placement. (Refer Section 3.4) **(5 Marks)**

Syllabus Topic : Search Engine Optimization / SPAM

Q. 8 What is search engine optimization? (Refer Section 3.5) **(5 Marks)**

Q. 9 Explain SPAM in detail. (Refer Section 3.5.1) **(5 Marks)**

Syllabus Topic : Web Size Measurement

Q. 10 Write note on web size measurement. (Refer Section 3.6) **(5 Marks)**

Syllabus Topic : Web Search Architecture

Q. 11 What are the various components of web search architecture? (Refer Section 3.7) **(5 Marks)**

Q. 12 Explain two major functions of search engine. (Refer Section 3.7) **(5 Marks)**

Syllabus Topic : Basic XML Concepts

Q. 13 Write short note on basic concepts of XML. (Refer Section 3.8) **(5 Marks)**

Syllabus Topic : Challenges in XML Retrieval

Q. 14 Define XML and challenges in XML retrieval. (Refer Section 3.9) **(5 Marks)**

Syllabus Topic : A Vector Space Model for XML Retrieval

Q. 15 Explain vector space model for XML retrieval. (Refer Section 3.10) **(5 Marks)**

Syllabus Topic : Evaluation of XML Retrieval

Q. 16 Write difference between Text-centric and Data-centric XML Retrieval. (Refer Section 3.11) **(5 Marks)**

Syllabus Topic : Text-centric Versus Data-centric XML Retrieval

Q. 17 Write short note on text-centric versus data-centric XML retrieval. (Refer Section 3.12) **(5 Marks)**

Chapter Ends...

□□□