UNIT-II

Cataloging and Indexing: Objectives, Indexing Process, Automatic Indexing, formation Extraction.
Data Structures: Introduction, Stemming Algorithms, Inverted file structures, N-gram data structure, PAT data structure, Signature file structure, Hypertext data structure.

CATALOGING AND INDEXING

INDEXING:

The transformation from received item to searchable data structure is called indexing.

• Process can be manual or automatic.

• Creating a direct search in document data base or indirect search through index files.

• Concept based representation: instead of transforming the input into a searchable format some systems transform the input into different representation that is conceptbased .Search ? Search and return item as per the incoming items.

• History of indexing: shows the dependency of information processing capabilities on manual and then automatic processing systems .

• Indexing originally called cataloguing : oldest technique to identity the contents of items to assist in retrieval.

• Items overlap between full item indexing , public and private indexing of files

Objectives :

The public file indexer needs to consider the information needs of all users of library system . Items overlap between full item indexing , public and private indexing of files.

•Users may use public index files as part of search criteria to increase recall.

•They can constrain there search by private index files

•The primary objective of representing the concepts within an item to facilitate users finding relevant information .

•Users may use public index files as part of search criteria to increase recall.

•They can constrain there search by private index files

•The primary objective of representing the concepts within an item to facilitate users finding relevant information

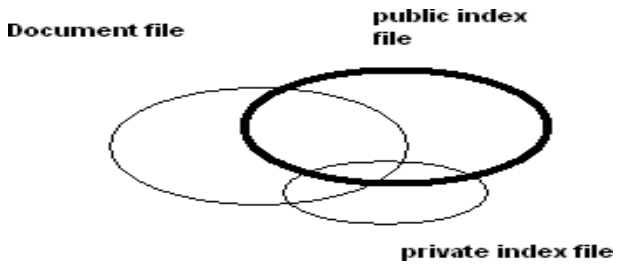


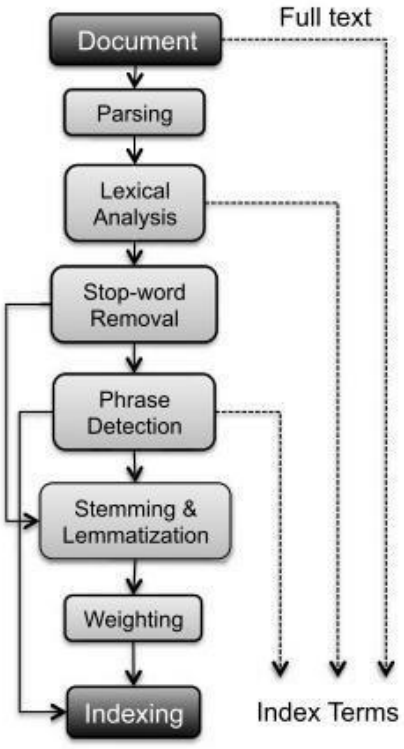
Fig : Indexing process

1. Decide the scope of indexing and the level of detail to be provided. Based on usage scenario of users.

2. Second decision is to link index terms together in a single index for a particular concept.

TEXT PROCESSING

Fig. 2.3 Text processing phases in an IR system



1. Document Parsing. Documents come in all sorts of languages, character sets, and formats; often, the same document may contain multiple languages

or formats, e.g., a French email with Portuguese PDF attachments. Document parsing deals with the recognition and "breaking down" of the document structure into individual components. In this pre processing phase, unit documents arecreated; e.g., emails with attachments are split into one documentrepresenting the email and as many documents as there are attachments.

2. Lexical Analysis. After parsing, lexical analysis tokenizes a document, seen as an input stream, into words. Issues related to lexical analysis include the correct identification of accents, abbreviations, dates, and cases. The difficulty of this operation depends much on the language at hand: for example, the English language has neither diacritics nor cases, French has diacritics but no cases, German has both diacritics and cases. The recognition of abbreviations and, in particular, of time expressions would deserve a separate chapter due to its complexity and the extensive literature in the field For current approaches

3. Stop-Word Removal. A subsequent step optionally applied to the results of lexical analysis is stop-word removal, i.e., the removal of high-frequency words. For example, given the sentence "search engines are the most visible information retrieval applications" and a classic stop words set such as the one adopted by the Snowball stemmer,1 the effect of stop-word removal would be: "search engine most visible information retrieval applications".

4. Phrase Detection. This step captures text meaning beyond what is possible with pure bag- of-word approaches, thanks to the identification of noun groups and other phrases. Phrase detection maybe approached in several ways, including rules (e.g., retaining terms that are not separated by punctuation marks), morphological analysis
, syntactic analysis, and combinations thereof. For example, scanning our example sentence "search engines are the most visible information retrieval applications" for noun phrases would probably result in identifying "search engines" and "information retrieval".

5. Stemming and Lemmatization. Following phrase extraction, stemmingand lemmatization aim at stripping down word suffixes in order to normalize the word. In particular, stemming is a heuristic process that"chops off" the ends of words in the hope of achieving the goal correctly most of the time; a classic rule based algorithm for this was devised by Porter [280]. According to the Porter stemmer, our example sentence "Search engines are the most visible information

retrieval applications" would result in: "Search engine are the most visible inform retriev applic".

1. Lemmatization is a process that typically uses dictionaries and morphological analysis of words in order to return the base or dictionary form of a word, thereby collapsing its inflectional forms (see, e.g., [278]). For example, our sentence would result in "Search engine are the most visible information retrieval application" when lemmatized according to a WordNet-based lemmatizer
2. Weighting. The final phase of text pre-processing deals with termweighting. As previously mentioned, words in a text have different descriptive power; hence, index terms can be weighted differently to account for their significance within a document and/or a document collection. Such a weighting can be binary, e.g., assigning 0 for term absence and 1 for presence.

Indexing Process

Indexing Process When an organization with multiple indexers decides to create a public or private index some procedural decisions on how to create the index terms assist the indexers and end users in knowing what to expect in the index file.

The first decision is the scope of the indexing to define what level of detail the subject index will contain. This is based upon usage scenarios of the end users. The other decision is the need to link index terms together in a single index for a particular concept

SCOPE OF NDEXING

- When perform the indexing manually, problems arise from two sources the author and the indexer the author and the indexer .
- Vocabulary domain may be different the author and the indexer.
- This results in different quality levels of indexing.
- The indexer must determine when to stop the indexing process.
- Two factors to decide on level to index the concept in a item.
- The exhaustively and how specific indexing is desired.
- Exhaustively of index is the extent to which the different concepts in the itemare indexed.
- For example, if two sentences of a 10-page item on microprocessors

discusses on-board caches, should this conceptbe indexed

- Specific relates to preciseness of index terms used in indexing.
- For example, whether the term “processor” or the term “microcomputer” or the term “Pentium” should be used in theindex of an item is based upon the specificity decision. Indexing an item only on the most important concept in it and using general index terms yields low exhaustively and specificity.
- Another decision on indexing is what portion of an item to be indexed Simplest case is to limit the indexing to title and abstract (conceptual) zone .
- General indexing leads to loss of precision and recall.

PREORDINATION AND LINKAGES

- Another decision on linkages process whether linkages are available between index terms for an item .
- Used to correlate attributes associated with concepts discussed inan item .’this process is called preordination .
- When index terms are not coordinated at index time the coordination occurs at search time. This is called post coordination , implementing by “AND” ing index terms .
- Factors that must be determined in linkage process are the number of terms that can be related.
- Ex., an item discusses ‘the drilling of oil wells in Mexico by CITGO and the introduction of oil refineries in Peru by the U.S.’

Figure 3.2 shows the different types of linkages. It assumes that an item discusses the drilling of oil wells in Mexico by CITGO and the introduction of oil refineries in Peru by the U.S. When the linked capability is added, the system does not erroneously relate Peru and Mexico since they are not in the same set of linked items. It still does not have the ability to discriminate between which country is introducing oil refineries into the other country. Introducing roles in the last two examples of Figure 3.2 removes this ambiguity. Positional roles treat the data as a vector allowing only one value per

position. Thus if the example is expanded so that the U.S. was introducing oil refineries in Peru, Bolivia and Argentina, then the positional role technique would require three entries, where the only difference would be in the value in the "affected country" position. When modifiers are used, only one entry would be required and all three countries would be listed with three "MODIFIER"s.

INDEX TERMS	Methodology
oil, wells, Mexico, CITGO, refineries, Peru, BP, drilling	No linking of terms
(oil wells, Mexico, drilling, CITGO)	linked (Precoordination)
(U.S.,oil refineries, Peru, introduction)	
(CITGO, drill, oil wells, Mexico) (U.S., introduction, oil refineries, Peru)	linked (Precoordination) with position indicating role
(SUBJECT: CITGO; ACTION: drilling; OBJECT: oil,wells MODIFIER: in Mexico) (SUBJECT:U.S.; ACTION:introduces; OBJECT: oil refineries; MODIFIER: in Peru)	linked (Pre-coordination) with modifier indicating role

Figure 3.2 Linkage of Index Terms

3.3 AUTOMATIC INDEXING

Automatic indexing is the capability for the system to automatically determine the index terms to be assigned to an item. The simplest case is when all words in the document are used as possible index terms (total document indexing).

More complex processing is required when the objective is to emulate a human indexer and determine a limited number of index terms for the major concepts in the item. As discussed, the advantages of human indexing are the ability to determine concept abstraction and judge the value of a concept. The disadvantages of human indexing over automatic indexing are cost, processing time and consistency. Once the initial hardware cost is amortized, the costs of automatic indexing are absorbed as part of the normal operations and maintenance costs of the computer system. There are no additional indexing costs versus the salaries and benefits regularly paid to human indexers. Processing time of an item by a human indexer varies significantly based upon the indexer's knowledge of the concepts being indexed, the exhaustivity and specificity guidelines and the amount and accuracy of preprocessing via Automatic File Build. Even for relatively short

items (e.g., 300 - 500 words) it normally takes at least five minutes per item. A significant portion of this time is caused by the human interaction with the computer (e.g., typing speeds, cursor positioning, correcting spelling errors, taking breaks between activities).

Automatic indexing requires only a few seconds or less of computer time based upon the size of the processor and the complexity of the algorithms to generate the index. Another advantage to automatic indexing is the predictably of algorithms. If the indexing is being performed automatically, by an algorithm, there is consistency in the index term selection process. Human indexers typically generate different indexing for the same document.

For example, the user may determine that searching for economic issues is for less precise than political issues in a particular newspaper based information system. The user may also determine that it is easier to find economic data in a information database containing Business Weekly than the newspaper source.

Indexes resulting from automated indexing fall **into two classes: weighted and unweighted**. In an unweighted indexing system, the existence of an index term in a document and sometimes its word location(s) are kept as part of the searchable data structure. No attempt is made to discriminate between the value of the index terms in representing concepts in the item. Looking at the index, it is not possible to tell the difference between the main topics in the item and a casual reference to a concept. This architecture is typical of the commercial systems through the 1980s. Queries against unweighted systems are based upon Boolean logic and the items in the resultant Hit file are considered equal in value. The last item presented in the file is as likely as the first item to be relevant to the user's information need.

In a weighted indexing system, an attempt is made to place a value on the index term's representation of its associated concept in the document. An index term's weight is based upon a function associated with the frequency of occurrence of the term in the item. Luhn, one of the pioneers in automatic indexing, introduced the concept of the "resolving power" of a term. Luhn postulated that the significance of a concept in an item is directly proportional to the frequency of use of the word associated with the concept in the document (Luhn-58, Salton-75). This is reinforced by the studies of Brookstein, Klein and Raita that show "content bearing" words are not randomly distributed (i.e., Poisson distributed), but that their occurrence "clump" within items (Brookstein-95).

Typically, values for the index terms are normalized between zero and one. The higher the weight, the more the term represents a concept discussed in the item. The weight can be adjusted to account for other information such as the number of items in the database that contain the same

concept

3.3.1 Indexing by Term

When the terms of the original item are used as a basis of the index process, **there are two major techniques for creation of the index: statistical and natural language. Statistical techniques can be based upon vector models and probabilistic models with a special case being Bayesian models.** They are classified as statistical because their calculation of weights use statistical information such as the frequency of occurrence of words and their distributions in the searchable database. Natural language techniques also use some statistical information, but perform more complex parsing to define the final set of index concepts.

Often weighted systems are discussed as vectorized information systems. This association comes from the SMART system at Cornell University created by Dr. Gerald Salton. The system emphasizes weights as a foundation for information detection and stores these weights in a vector form. Each vector represents a document and each position in a vector represents a different unique word (processing token) in the database. The value assigned to each position is the weight of that term in the document. A value of zero indicates that the word was not in the document.

Queries can be translated into the vector form. Search is accomplished by calculating the distance between the query vector and the document vectors. In addition to a vector model, the other dominant approach uses a probabilistic model. The model that has been most successful in this area is the Bayesian approach. This approach is natural to information systems and is based upon the theories of evidential reasoning. The Bayesian approach could be applied as part of index term weighting, but usually is applied as part of the retrieval process by calculating the relationship between an item and a specific query. A Bayesian network is a directed acyclic graph in which each node represents a random variable and the arcs between the nodes represent a probabilistic dependence between the node and its parents

Figure 3.3 shows the basic weighting approach for index terms or associations between query terms and index terms.

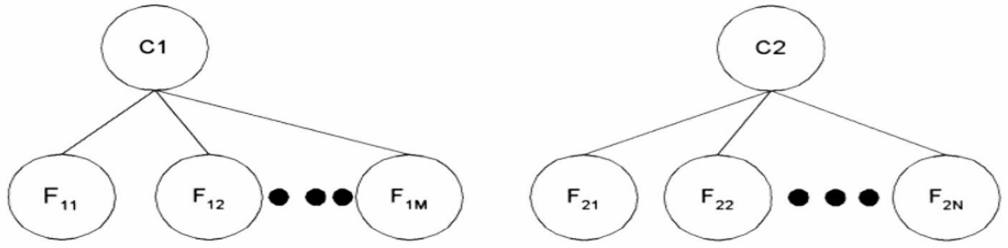


Figure 3.3 Two-level Bayesian network

The nodes C_1 and C_2 represent "the item contains concept C_i" and the F nodes represent "the item has feature (e.g., words) F_{ij}." The network could also be interpreted as C representing concepts in a query and F representing concepts in an item. The goal is to calculate the probability of C_i given F_{ij}. To perform that calculation two sets of probabilities are needed:

1. The prior probability $P(C_i)$ that an item is relevant to concept C
2. The conditional probability $P(F_{ij}/C_i)$ that the features F_{ij} where $j = 1$, m are present in an item given that the item contains topic C_i.

The automatic indexing task is to calculate the posterior probability $P(C_i/F_{i1}, \ldots, F_{im})$, the probability that the item contains concept C_i given the presence of features F_{ij}. The Bayes inference formula that is used is:

$$P(C_i/F_{i1}, \ldots, F_{im}) = P(C_i)\, P(F_{i1}, \ldots, F_{im}/C_i) \backslash P(F_{i1}, \ldots, F_{im}).$$

If the goal is to provide ranking as the result of a search by the posteriors, the Bayes rule can be simplified to a linear decision rule:

$$g(C_i/F_{i1}, \ldots, F_{im}) = \Sigma_k\, I(F_{ik})\, w(F_{ik}, C_i)$$

Where $I(F_{ik})$ is an indicator variable that equals 1 only if F_{ik} is present in the item (equals zero otherwise) and w is a coefficient corresponding to a specific feature/concept pair. A careful choice of w produces a ranking in decreasing order that is equivalent to the order produced by the posterior probabilities. Interpreting the coefficients, w, as weights corresponding to each feature (e.g., index term) and the function g as the sum of the weights of the features, the result of applying the formula is a set of term weights.

Another approach to defining indexes to items is via use of natural language processing. The DR-LINK (Document Retrieval through Linguistic Knowledge) system processes items at the morphological, lexical, semantic, syntactic, and discourse levels. Each level uses information from the previous level to perform its additional analysis. The discourse level is abstracting information beyond the sentence level and can determine abstract concepts using pre-defined models of event relationships. This allows the indexing to include specific term as well as abstract concepts such as time (e.g., differentiates between a company was sold versus a company will be sold). Normal automatic indexing does a poor job at identifying and extracting "verbs" and relationships between objects based upon the verbs.

3.3.2 Indexing by Concept

The basis for concept indexing is that there are many ways to express the same idea and increased retrieval performance comes from using a single representation. Indexing by term treats each of these occurrences as a different index and then uses thesauri or other query expansion techniques to expand a query to find the different ways the same thing has been represented. Concept indexing determines a canonical set of concepts based upon a test set of terms and uses them as a basis for indexing all items This is also called Latent Semantic Indexing because it is indexing the latent semantic information in items. The determined set of concepts does not have a label associated with each concept (i.e., a word or set of words that can be used to describe it), but is a mathematical representation (e.g., a vector). An example of a system that uses concept indexing is the MatchPlus system developed by HNC Inc. The MatchPlus system uses neural networks to facilitate machine learning of concept/word relationships and sensitivity to similarity of use (Caid-93). The systems goal is to be able to determine from the corpus of items, word relationships (e.g., synonyms) and the strength of these relationships and use that information in generating context vectors. Two neural networks are used. One neural network learning algorithm generates stem context vectors that are sensitive to similarity of use and another one performs query modification based upon user feedback.

Word stems, items and queries are represented by high dimensional (at least 300 dimensions) vectors called context vectors. Each dimension in a vector could be viewed as an abstract concept class. The approach is based upon cognitive science work by Waltz and Pollack. To define context vectors, a set of n features are selected on an ad hoc basis (e.g., high frequency terms after removal of stop words). The selection of the initial features is not critical since they evolve and expand to the abstract concept classes used in the indexing process. For any word stem k, its context vector V^k is an n-dimensional vector with each component j interpreted as follows:

$\mathbf{V}^k$ positive if k is strongly associated with feature j
$\mathbf{V}^k \approx \mathbf{0}$ if word k is not associated with feature j
$\mathbf{V}^k$ negative if word k contradicts feature j

The interpretation of components for concept vectors is exactly the same as weights in neural networks. Each of the n features is viewed as an abstract concept class. Then each word stem is mapped to how strongly it reflects each concept in the items in the corpus. There is overlap between the concept classes (features) providing a distributed representation and insulating against a small number of entries for context vectors that could have no representation for

particular stems, Once the context vectors for stems are determined, they are used to create the index for an item. A weighted sum of the context vectors for all the stems in the item is calculated and normalized to provide a vector representation of the item in terms of the n concept classes (features).

3.3.3 Multimedia Indexing

Indexing associated with multimedia differs from the previous discussions of indexing. The automated indexing takes place in multiple passes of the information versus just a direct conversion to the indexing structure. The first pass in most cases is a conversion from the analog input mode into a digital structure. Then algorithms are applied to the digital structure to extract the unit of processing of the different modalities that will be used to represent the item. In an abstract sense this could be considered the location of a processing token in the modality. This unit will then undergo the final processing that will extract the searchable features that represent the unit. Indexing video or images can be accomplished at the raw data level (e.g., the aggregation of raw pixels), the feature level distinguishing primitive attributes such as color and luminance, and at the semantic level where meaningful objects are recognized (e.g., an airplane in the image/video frame).

An example is processing of video. The system will periodically collect a frame of video input for processing. It might compare that frame to the last frame captured to determine the differences between the frames. If the difference is below a threshold it will discard the frame. For a frame requiring processing, it will define a vector that represents the different features associated with that frame. Each dimension of the vector represents a different feature level aspect of the frame. The vector then becomes the unit of processing in the search system. This is similar to processing an image. Semantic level indexing requires pattern recognition of objects within the images.

A single phoneme can be divided into four states for the Markov model. It is the textual words assocaited with the audio that becomes the searchable structure. In addition to storing the extracted index searchable data, a multimedia item needs to also store some mechanism to correlate the different modalities during search. There are two main mechanisms that are used, positional and temporal. Positional is used when the modalities are interspersed in a linear sequential composition. For example a document that has images or audio inserted, can be considered a linear structure and the only relationship between the modalities will be the juxtaposition of each modality. This would allow for a query that would specify location of an image of a boat within one paragraph of "Cuba and refugees".

The second mechanism is based upon time because the modalities are executing concurrently.

The typical video source off television is inherently a multimedia source. It contains video, audio, and potentially closed captioning. Also the creation of multimedia presentations are becoming more common using the Synchronized Multimedia Integration Language (SMIL). It is a mark-up language designed to support multimedia presentations that integrate text (e.g., from slides or free running text) with audio, images and video. In both of these examples, time is the mechanism that is used to synchronize the different modalities. Thus the indexing must include a time-offset parameter versus a physical displacement. It also suggests that the proximity to increase precision will be based upon time concurrency (or ranges) versus physical proximity.

3.4 Information Extraction

There are two processes associated with information extraction: determination of facts to go into structured fields in a database and extraction of text that can be used to summarize an item. In the first case only a subset of the important facts in an item may be identified and extracted. In summarization all of the major concepts in the item should be represented in the summary. The process of extracting facts to go into indexes is called Automatic File Build. Its goal is to process incoming items and extract index terms that will go into a structured database. This differs from indexing in that its objective is to extract specific types of information versus understanding all of the text of the document.

An Information Retrieval System's goal is to provide an in depth representation of the total contents of an item. Information Extraction system only analyzes those portions of a document that potentially contain information relevant to the extraction criteria.

The objective of the data extraction is in most cases to update a structured database with additional facts. The updates may be from a controlled vocabulary or substrings from the item as defined by the extraction rules. The term "slot" is used to define a particular category of information to be extracted. Slots are organized into templates or semantic frames. Information extraction requires multiple levels of analysis of the text of an item. It must understand the words and their context (discourse analysis). The processing is very similar to the natural language processing described under indexing. In establishing metrics to compare information extraction, the previously defined measures of precision and recall are applied with slight modifications to their meaning.

Recall refers to how much information was extracted from an item versus how much should have been extracted from the item. It shows the amount of correct and relevant data extracted versus the correct and relevant data in the item. Precision refers to how much information was extracted accurately versus the total information extracted. Additional metrics used are overgeneration and fallout. Overgeneration measures the amount of irrelevant information that is extracted. This

could be caused by templates filled on topics that are not intended to be extracted or slots that get filled with non-relevant data. Fallout measures how much a system assigns incorrect slot fillers as the number of potential incorrect slot fillers increases. These measures are applicable to both human and automated extraction processes. Human beings fall short of perfection in data extraction as well as automated systems.

Another related information technology is document summarization. Rather than trying to determine specific facts, the goal of document summarization is to extract a summary of an item maintaining the most important ideas while significantly reducing the size.

Examples of summaries that are often part of any item are titles, table of contents, and abstracts with the abstract being the closest. The abstract can be used to represent the item for search purposes or as a way for a user to determine the utility of an item without having to read the complete item.

summaries does not intrinsically imply major deficiencies between the summaries. Most automated algorithms approach summarization by calculating a score for each sentence and then extracting the sentences with the highest scores. Some examples of the scoring techniques are use of rhetorical relations, contextual inference and syntactic coherence using cue words and statistical weighting properties. There is no overall theoretic basis for the approaches leading to many heuristic algorithms.

Statistical classification approach based upon a training set reducing the heuristics by focusing on a weighted combination of criteria to produce “optimal” scoring scheme. They selected the following five feature sets as a basis for their algorithm:

1.Sentence Length Feature that requires sentence to be over five words in length

2.Fixed Phrase Feature that looks for the existence of phrase “cues” (e.g. “in conclusion)

3.Paragraph Feature that places emphasis on the first ten and last five paragraphs in an item and also the location of the sentences within the paragraph

4.Thematic Word Feature that uses word frequency

5.Uppercase Word Feature that places emphasis on proper names andacronyms.

DATA STRUCTURES

- Introduction to Data Structures
- Stemming Algorithms
- Inverted File Structure

- N-Gram Data Structure
- PAT Data Structure
- Signature File Structure
- Hypertext and XML Data Structures

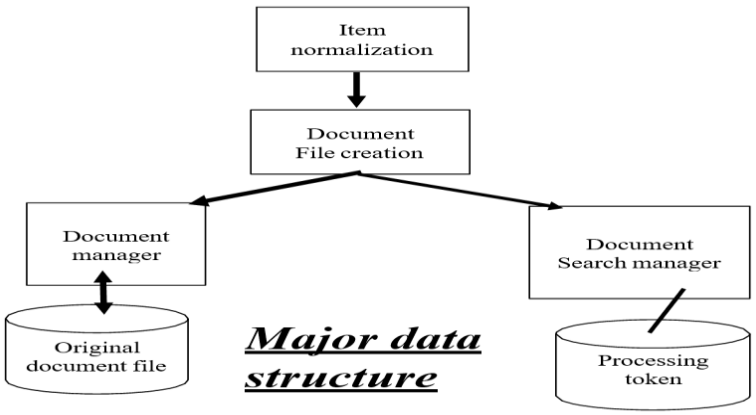
4.Data structure :

The knowledge of data structure gives an insight into the capabilitiesavailable to the system .

- Each data structure has a set of associated capabilities .
- Ability to represent the concept and their r/s.
- Supports location of those concepts IntroductionTwo major data structures in any IRS:

1. One structure stores and manages received items in theirnormalized form is called document manger

2. The other data structure contains processing tokens andassociated data to support search.



Result of a search are references to the items that satisfy the search statement which are passed to the document manager for retrieval.

Focus : on data structure that support search function

Stemming : is the transformation often applied to data before placing it in the searchable data structure

Stemming represents concept(word) to a canonical (authorized; recognized; accepted)morphological (the patterns of word formation in a particular language) representation .Risk with stemming : concept discrimination information may be lost in the

process. Causing decrease in performance.

Advantage : has a potential to increase recall.

STEMMING ALGORITHMS

- Stemming algorithm is used to improve the efficiency of IRS and improverecall.
- Conflation(the process or result of fusing items into one entity; fusion; amalgamation)is a term that is used to refer mapping multiple morphological variants to single representation(stem).
- Stem carries the meaning of the concept associated with the word and the affixes(ending) introduce subtle(slight) modification of the concept.
- Terms with a common stem will usually have similar meanings,for

  example:
- Ex : Terms with a common stem will usually have similarmeanings, for example:
- CONNECT
- CONNECTED
- CONNECTING
- CONNECTION
- CONNECTIONS
- Frequently, the performance of an IR system will be improved if term groups such as this are conflated into a single term. This maybe done by removal of the various suffixes -ED, -ING, -ION, IONS to leave the single term CONNECT
- In addition, the suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is always advantageous.

❖ Major usage of stemming is to improve recall.

➢ Important for a system to categories a word prior to making the decision tostem.

➢ Proper names and acronyms (A word formed from the initial letters of a namesay IARE …) should not have stemming applied.

➢ Stemming can also cause problems for natural language processing NPLsystems by causing loss of information .

PORTER STEMMING ALGORITHM

- Based on a set condition of the stem
- A consonant in a word is a letter other than A, E, I, O or U, some important stem conditions are

1. The measure m of a stem is a function of sequence of vowels (V) followed by a sequence of consonant (C) .
2. C (VC)mV. m is number VC repeats The case m = 0 covers the null word.
3. *<X> - stem ends with a letter X 3.*v* - stem contains a vowel
4. *d- stem ends in double consonant (e.g. -TT, -SS).

5. *o- stem ends in consonant vowel sequence where the final consonant is not w,x,y(e.g. -WIL, -HOP).

Suffix cond.s takes the form current _suffix = = pattern Actions are in the form old_suffix ->. New_suffix

Rules are divided into steps to define the order for applying the rule.

Examples of the rules

Step	Condition	Suffix	Replacement	Example
1a	Null	Sses	Ss	Stresses -> stress
1b	*v*	Ing	Null	Making -> mak
1b1	Null	At	Ate	Inflated-> inflate
1c	*v*	Y	I	Happy->happi
2	m>0	aliti	al	Formaliti-> formal
3	m>0	Icate	Ic	Duplicate->duplie
4	m>1	Able	Null	Adjustable -> adjust
5a	m>1	e	Null	Inflate-> inflat
5b	m>1 and *d	Null	Single letter	Control -> control

2. Dictionary look up stemmers

- ❖ Use of dictionary look up.
- ❖ The original term or stemmed version of the term is looked upin a dictionary and replaced by the stem that best represents it.
- ❖ This technique has been implemented in INQUERY and Retrieval ware systems-

INQUERY system uses the technique called Kstem.

- ❖ Kstem is a morphological analyzer that conflates words variants to a root form.
- ❖ It requires a word to be in the dictionary
- ❖ Kstem uses 6 major data files to control and limit the stemming process.

1. Dictionary of words (lexicon)
2. Supplemental list of words for dictionary

3. Exceptional list of words that should retain a ‘e’ at the end (e.g., “suites” to “suite” but “suited” to “suit”).
4. Direct _conflation - word pairs that override stemming algorithm.
5. County_nationality _conflation (British maps to Britain)
6. Proper nouns -- that should not be stemmed

- New words that are not special forms (e.g., dates, phone numbers) are located in the dictionary to determine simpler forms by stripping off suffixes and respelling plurals as defined in the dictionary.

3. Successor stemmers:

- Based on length of prefixes .
- The smallest unit of speech that distinguishes on word from another
- The process uses successor varieties for a word .

Uses information to divide a word into segments and selects on of the segments to stem.

The successor variety for any prefix of a word is the number of children that are associated with the node in the symbol tree representing that prefix.

Figure 4.2 shows the symbol tree for the terms bag, barn, bring, both, box, and bottle. The successor variety for any prefix of a word is the number of children that are associated with the node in the symbol tree representing that prefix. For example, the successor variety for the first letter “b” is three. The successor variety for the prefix “ba” is two.

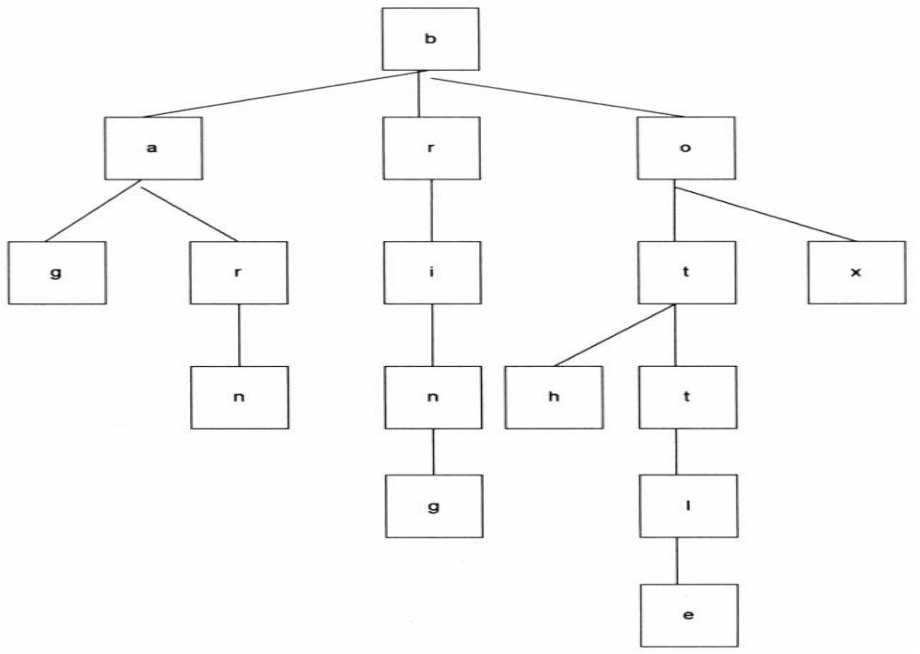


Figure 4.2 Symbol Tree for terms bag, barn, bring, box, bottle , both

1. Cutoff method: a cutoff value is selected to define stem length. The value varies for each possible set of words.
2. Peak and Plateau: a segment break is made after a character whose successor variety exceeds that of the character immediately preceding it and the character immediately following it.
3. Complete word method: break on boundaries of complete words.

4. Entropy method: uses the distribution of successor variety letters. Let $|D_{ak}|$ be the number of words beginning with the k length sequence of letters a. Let $|D_{akj}|$ be the number of words in Dak with successor j. The probability that a member of Dak has the successor j is given by $|D_{akj}|/|D_{ak}|$. The entropy (Average Information as defined by Shannon-51) of $|D_{ak}|$ is:

$$H_{ak} = \sum_{p=1}^{26} -(|D_{akj}|/|D_{ak}|) \ (\log_2(|D_{akj}|/|D_{ak}|))$$

Using this formula a set of entropy measures can be calculated for a word and its predecessors. A cutoff value is selected and a boundary is identified whenever the cutoff value is reached. Hafer and Weiss experimented with the techniques, discovering that combinations of the techniques performed best, which they used in defining their stemming process. Using the words in Figure 4.2 plus the additional word "boxer," the successor variety stemming is shown in Figure 4.3.

PREFIX	Successor Variety	Branch Letters
b	3	a,r,o
bo	2	t,x
box	1	e
boxe	1	r
boxer	1	blank

Figure 4.3 Successor Variety Stemming

If the cutoff method with value four was selected then the stem would be "boxe." The peak and plateau method can not apply because the successor variety monotonically decreases. Applying the complete word method, the stem is "box." The example given does not have enough values to apply the entropy method.

The advantage of the peak and plateau and the complete word methods is that a cutoff value does not have to be selected. After a word has been segmented, the segment to be used as the stem must be selected. Hafer and Weiss used the following rule:

```
if (first segment occurs in <= 12 words in database)
        first segment is stem
else (second segment is stem)
```

The idea is that if a segment is found in more than 12 words in the text being analyzed, it is probably a prefix. Hafer and Weiss noted that multiple prefixes in the English language do not occur often and thus selecting the first or second segment in general determines the appropriate stem.

4.3 INVERTED FILE STRUCTURE

Inverted file structure

Most common data structure

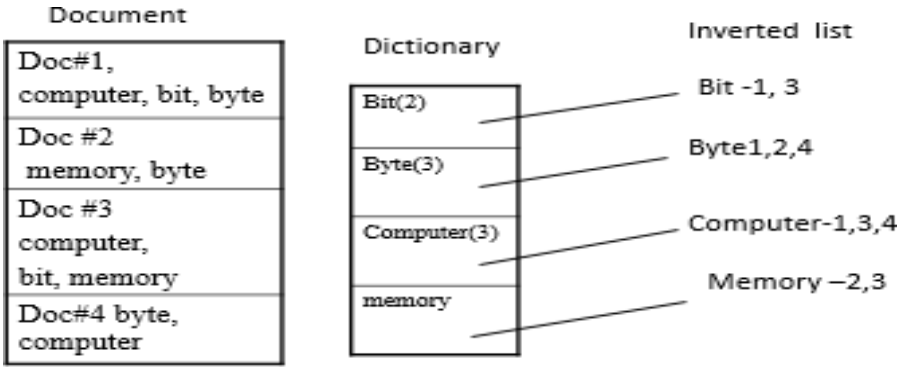
Inverted file structures are composed of three files

The document file

1. The inversion list (Posting List)
2. Dictionary
3. The inverted file : based on the methodology of storing an inversion of documents.
4. For each word a listof documents in which the word is found is stored(inversion of document
5. Each document is given a unique the numerical identifier that is stored in inversion list . Dictionary is used to located the inversion list for a particular word.

Which is a sorted list(processing tokens) in the system and a pointer to the location of its inversion list.

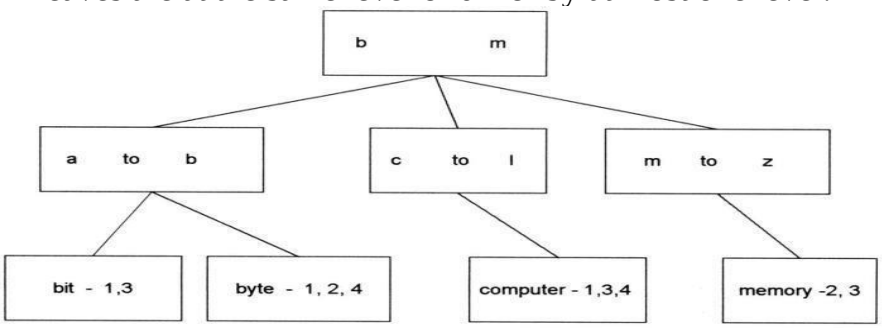
Dictionary can also store other information used in query optimization such as length of inversion lists to increase the precision.



- Use zoning to improve
- precision and Restrict entries.
- Inversion list consists of document identifier for each documentin which the word is found.

Ex: bit 1(10),1(12) 1(18) is in 10,12, 18 position of the word bit in the document #1.

- When a search is performed, the inversion lists for the terms in the query are locate and appropriate logic is applied between inversion lists.
- Weights can also be stored in the inversion list.
- Inversion list are used to store concept and their relationship.
- Words with special characteristics can be stored in their own dictionary. Ex: Date… which require date ranging and numbers.
- Systems that support ranking are re-organized in ranked order.
- B trees can also be used for inversion instead of dictionary.
- The inversion lists may be at the leaf level or referenced in higher level pointers.
- A B-tree of order m is defined as:
- A root node with between 2 and 2m keys
- All other internal nodes have between m and 2m keys
- All keys are kept in order from smaller to larger.
- All leaves are at the same level or differ by at most one level.



4.4 N-GRAM DATA STRUCTURE

- N-Grams can be viewed as a special technique for conflation (stemming) and asa unique data structure in information systems.
- N-Grams are a fixed length consecutive series of "n" characters.

- Unlike stemming that generally tries to determine the stem of a word thatrepresents the semantic meaning of the word, n-grams do not care about semantics.
- The searchable data structure is transformed into overlapping n-grams, whichare then used to create the searchable database.
- It is possible that the same n-gram can be created multiple times from a singleword.

Figure 4.7 for the word phrase "sea colony." For n-grams, with n greater than two, some systems allow interword symbols to be part of the n-gram set usually excluding the single character with interword symbol option. The symbol # is used to represent the interword symbol which is anyone of a set of symbols (e.g., blank, period, semicolon, colon, etc.). Each of the n-grams created becomes a separate processing tokens and are searchable. It is possible that the same n-gram can be created multiple times from a single word.

se ea co ol lo on ny	Bigrams (no interword symbols)
sea col olo lon ony	Trigrams (no interword symbols)
#se sea ea# #co col olo lon ony ny#	Trigrams (with interword symbol #)
#sea# #colo colon olony lony#	Pentagrams (with interword symbol #)

Figure 4.7 Bigrams, Trigrams and Pentagrams for "sea colony"

Uses :

- Widely used as cryptography in world war II Spelling errors detection and correction Use bigrams for conflating terms.
- N-grams as a potential erroneous words.
- Damerau specified 4 categories of errors:

Error Category	Example
single char insertion	compuuter
single char deletion	compter
single char substitution	compiter
Transposition of 2 adjacent chars	comp*tu*er

- Zamora showed trigram analysis provided a viable data structurefor identifying misspellings and transposed characters.
- This impacts information systems as a possible basis for identifying potential input errors for correction as a procedure within the normalization process.
- Frequency of occurrence of n-gram patterns can also be used for identifying the language of an item.
- Trigrams have been used for text compression and to manipulate the length of index terms.
- To encode profiles for the Selective Dissemination of Information.
- To store the searchable document file for retrospective search databases.

Advantage:

They place a finite limit on the number of searchable token

MaxSeg n=(□)n maximum number of unique n grams that can be generated. " n" is the length of n-grams

□ number of process able symbols

Disadvantage: longer the n gram the size of inversion list increase. Performance has 85 % precision .

4.5 PAT data structure (practical algorithm to retrieve information coded in alphanumeric)

- PAT structure or PAT tree or PAT array : continuous text input data structures(string like N- Gram data structure).
- The input stream is transformed into a searchable data structure consisting of substrings, all substrings are unique.
- Each position in a input string is a anchor point for a sub string.
- In creation of PAT trees each position in the input string is the anchor point for a sub-string that starts at that point and includesall new text up to the end of the input.
- Binary tree, most common class for prefix search,But Pat trees are sorted logically which facilitate range search, and more accurate then inversion file .
- PAT trees provide alternate structure if supporting strings search.

A substring can start at any point in the text and can be uniquely indexed by its starting location and length. If all strings are to the end of the input, only the starting location is needed since the length is the difference from the location and the total length of the item. It is possible to have a substring go beyond the length of the input stream by adding additional null characters. These substrings are called sistring (semi-infinite string). Figure 4.9 shows some possible sistrings for an input text.

Text	Economics for Warsaw is complex.
sistring 1	Economics for Warsaw is complex.
sistring 2	conomics for Warsaw is complex.
sistring 5	omics for Warsaw is complex.
sistring 10	for Warsaw is complex.
sistring 20	w is complex.
sistring 30	ex.

Figure 4.9 Examples of sistrings

A PAT tree is an unbalanced, binary digital tree defined by the sistrings. The individual bits of the sistrings decide the branching patterns with zeros branching left and ones branching right. PAT trees also allow each node in the tree to specify which bit is used to determine the branching via bit position or the number of bits to skip from the parent node.

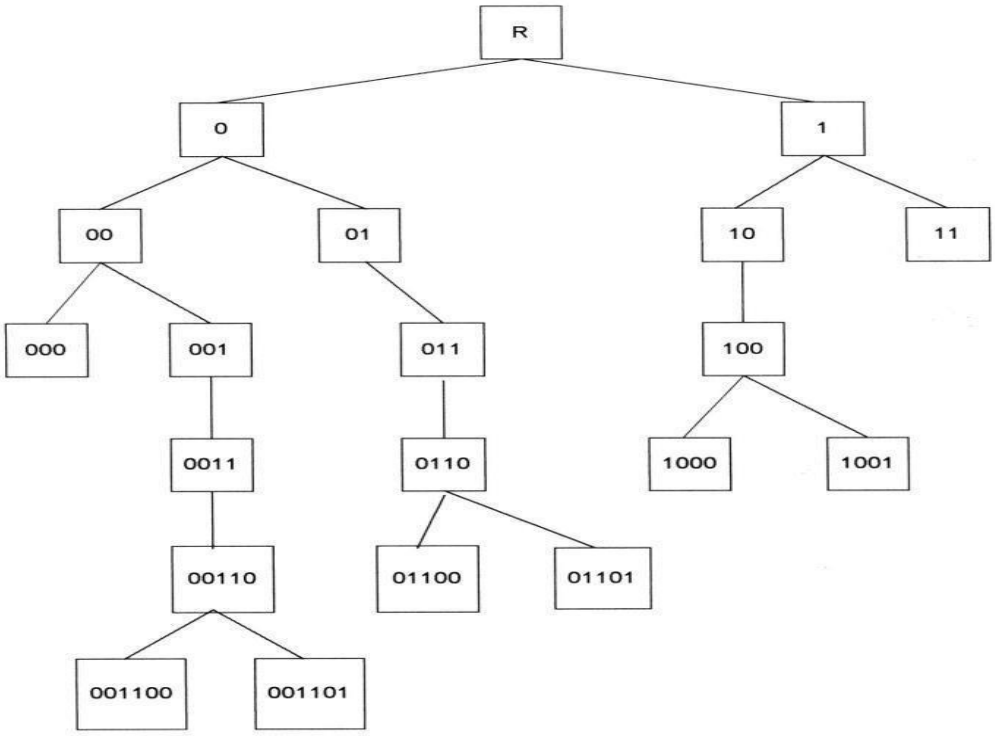
This is useful in skipping over levels that do not require branching. The key values are stored at the leaf nodes (bottom nodes) in the PAT Tree. For a text input of size "n" there are "n" leaf nodes and "n-1" at most higher level nodes. It is possible to place additional constraints on sistrings for the leaf nodes. We may be interested in

limiting our searches to word boundaries.

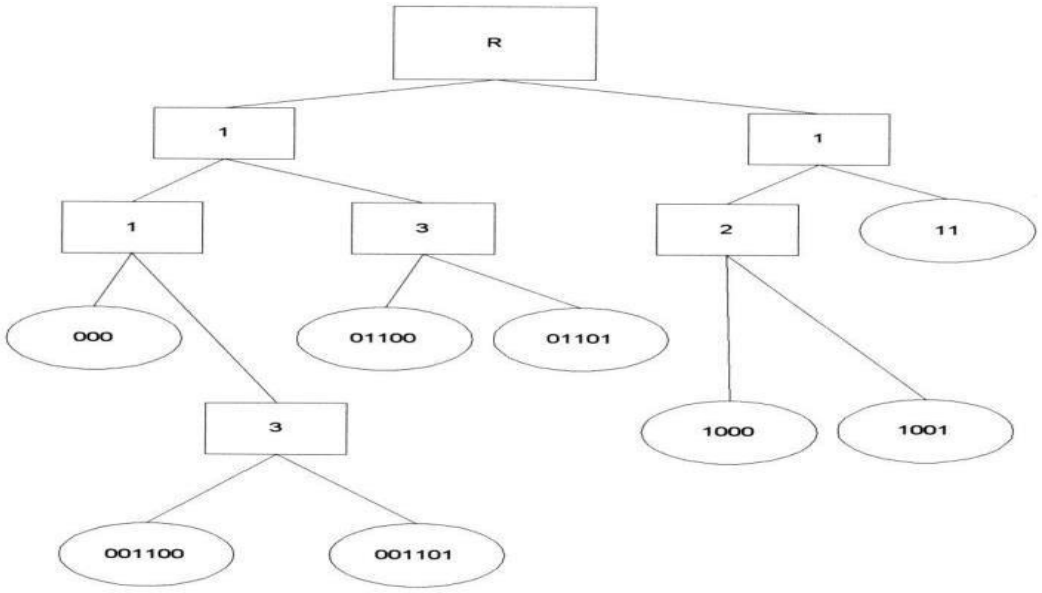
- The key values are stored at the leaf nodes (bottom nodes) in the PAT Tree.
- For a text input of size "n" there are "n" leaf nodes and "n-1" at most higher level nodes.
- It is possible to place additional constraints on sistrings for the leaf nodes.
- If the binary representations of "h" is (100), "o" is (110), "m" is (001) and "e" is (101) then the word "home" produces the input100110001101. Using the sistrings.

INPUT	100110001101
sistring 1	1001....
sistring 2	001100...
sistring 3	01100....
sistring 4	11.....
sistring 5	1000...
sistring 6	000.....
sistring 7	001101
sistring 8	01101

The full PAT binary tree is



The value in the intermediate nodes (indicated by rectangles) is the number of bits to skip until the next bit to compare that causes differences between similar terms.



Skipped final version of PAT tree

4.6 Signature file structure

- The coding is based upon words in the code.
- The words are mapped into word signatures .
- A word signature is fixed length code with a fixed number of bits set to 1.
- The bit positions that are set to one are determined via a hash function of the word.
- The word signatures are Ored together to create signature of an item..
- Partitioning of words is done in block size ,Which is nothing butset of words, Code length is 16 bits .
- Search is accomplished by template matching on the bit position .
- provide a practical solution applied in parallel processing , distributed environment etc.

- To avoid signatures being too dense with "1"s, a maximum number of words is specified and an item is partitioned into blocks of that size.
- The block size is set at five words, the code length is 16 bits and the number of bits that are allowed to be "1" for each word is five.
- TEXT: Computer Science graduate students study (assume block size is five words)

WORD	Signature
computer	0001 0110 0000 0110
Science	1001 0000 1110 0000
graduate	1000 0101 0100 0010
students	0000 0111 1000 0100
study	0000 0110 0110 0100
Block Signature	1001 0111 1110 0110

Superimposed Coding

Application(s)/Advantage(s)

- Signature files provide a practical solution for storing and locating information in a number of different situations.
- Signature files have been applied as medium size databases, databases with low frequency of terms, WORM devices, parallel processing machines, and distributed environments

4.7 HYPERTEXT AND XML DATA STRUCTURES

- The advent of the Internet and its exponential growth and wide acceptance as a new global information network has introduced new mechanisms for representing information.
- This structure is called hypertext and differs from traditional information storage data structures in format and use.
- The hypertext is Hypertext is stored in HTML format and XML .
- Bot of these languages provide detailed descriptions for subsets of text similar to the zoning.
- Hypertext allows one item to reference another item via a embedded pointer .
- HTML defines internal structure for information exchange over WWW on the internet.
- XML: defined by DTD, DOM, XSL, etc.

Document and term clustering

Two types of clustering:

1) clustering index terms to create a statistical thesaurus and

2) clustering items to create document clusters. In the first case clustering is used to increase recall by expanding searches with related terms. In document clustering the search can retrieve items similar to an item of interest, even if the query would not have retrieved the item. The clustering process is not precise and care must be taken on use of clustering techniques to minimize the negative impact misuse can have.

4.7.1 Definition of Hypertext Structure

The Hypertext data structure is used extensively in the Internet environment and requires an electronic

media storage for the item. Hypertext allows one item to reference another item via an imbedded pointer. Each separate item is called a node and the reference pointer is called a link. The referenced item can be of the same or a different data type than the original (e.g., a textual item references a photograph). Each node is displayed by a viewer that is defined for the file type associated with the node.

For example, Hypertext Markup Language (HTML) defines the internal structure for information exchange across the World Wide Web on the Internet. A document is composed of the text of the item along with HTML tags that describe how to display the document. Tags are formatting or structural keywords contained between less-than, greater than symbols (e.g., <title>, meaning display prominently).

The HTML tag associated with hypertext linkages is <a

href= …#NAME /a> where "a" and "/a" are an anchor start tag and anchor end tag denoting the text that the user can activate, "href" is the hypertext reference containing either a file name if the referenced item is on this node or an address (Uniform Resource Locator - URL) and a file name if it is on another node.

"#NAME" defines a destination point other than the top of the item to go to. The URL has three components: the access method the client used to retrieve the item, the Internet address of the server where the item is stored, and the address of the item at the server (i.e., the file including the directory it is in). For example, the URL for the HTML specification appears:

http://info.cern.ch/hypertext/WWW/MarkUp/HTML.html "HTTP" stands for the Hypertext Transfer Protocol which is the access protocol used to retrieve the item in HTML. Other Internet protocols are used for other activities such as file transfer (ftp://), a specific text search system (gopher://), remote logon (tenet://) and collaborative newsgroups (news://). The destination point is found in "info.cern.ch" which is the name of the "info" machine at CERN with "ch" being Switzerland, and "/hypertext/WWW/MarkUP/HTML.html" defines where to find the file HTML.html. Figure 4.14 shows an example of a segment of a HTML document. Most of the formatting tags indicated by < > are not described, being out of the scope of this text, but detailed descriptions can be found in the hundreds of books available on HTML. The <a href= ... > are the previously described hypertext linkages.

An item can have many hypertext linkages. Thus, from any item there are multiple paths that can be followed in addition to skipping over the linkages to continue sequential reading of the item. This is similar to the decision a reader makes upon reaching a footnote, whether to continue reading or skip to the footnote. Hypertext is sometimes called a "generalized footnote."

```
<CENTER>
<IMG SC="/images/home_iglo.jpg" WIDTH=468 HEIGHT=107
BORDER=0 ALT="WELCOME TO NETSCAPE><BR>
<P>
<DL>
<A HREF="/comprod/mirror/index.html">
<DD>
The beta testing is over: please read our report <A
HREF="http://www.charm.net/doc/charm/report/theme.html"> and your
can find more references at
HREF="http://www.charm.net/doc/charm/results/tests.html">
```

Figure 4.14 Example of Segment of HTML

In a conventional item the physical and logical structure are closely related. The item is sequential with imbedded citations to other distinct items or locations in the item. From the author's perspective, the substantive semantics lie in the sequential presentation of the information. Hypertext is a non-sequential directed graph structure, where each node contains its own information. The author assumes the reader can follow the linked data as easily as following the sequential presentation. A node may have several outgoing links, each of which is then associated with some smaller part of the node called an anchor. When an anchor is activated, the associated link is followed to the destination node, thus navigating the hypertext network. The organizational and reference structure of a conventional item is fixed at printing time while hypertext nodes and links can be changed dynamically. New linkages can be added and the information at a node can change without modification to the item referencing it.

4.7.3 XML

The eXtensible Markup Language (XML) is starting to become a standard data structure on the WEB. Its first recommendation (1.0) was issued on February 10, 1998. It is a middle ground between the simplicities but lack of flexibility of HTML and the complexity but richness of SGML (ISO 8879). Its objective is extending HTML with semantic information. The logical data structure within XML is defined by a Data Type Description (DTD) and is not constrained to the 70 defined tags and 50 attributes in the single DTD for HTML. The user can create any tags needed to describe and manipulate their structure. The W3C (Worldwide Web Consortium) is redeveloping HTML as a suite of XML tags. The following is a simple example of XML tagging:

```
<company>Widgets Inc.</company>
<city>Boston</city>
<state>Mass</state>
<product>widgets</product>
```

The W3C is also developing a Resource Description Format (RDF) for representing properties of WEB resources such as images, documents and relationships between them. This will include the Platform for Internet Content Selection (PICS) for attaching labels to material for filtering (e.g., unsuitable for children).

UNIT-III

5.1 Classes of Automatic Indexing

Automatic indexing is the process of analyzing an item to extract the information to be permanently kept in an index. This process is associated with Data Flow in an Information Processing System is reproduced here as Figure 5.1 to show where the indexing process is in the overall processing of an item. The figure is expanded to show where the search process relates to the indexing process. The left side of the figure including Identify Processing Tokens, Apply Stop Lists, Characterize tokens, Apply Stemming and Create Searchable Data Structure is all part of the indexing process. All systems go through an initial stage of zoning and identifying the processing tokens used to create the index. Some systems automatically divide the document up into fixed length passages or localities, which become the item unit that is indexed Filters, such as stop lists and stemming algorithms, are frequently applied to reduce the number of tokens to be processed. The next step depends upon the search strategy of a particular system. Search strategies can be classified as statistical, natural language, and concept. An index is the data structure created to support the search strategy.

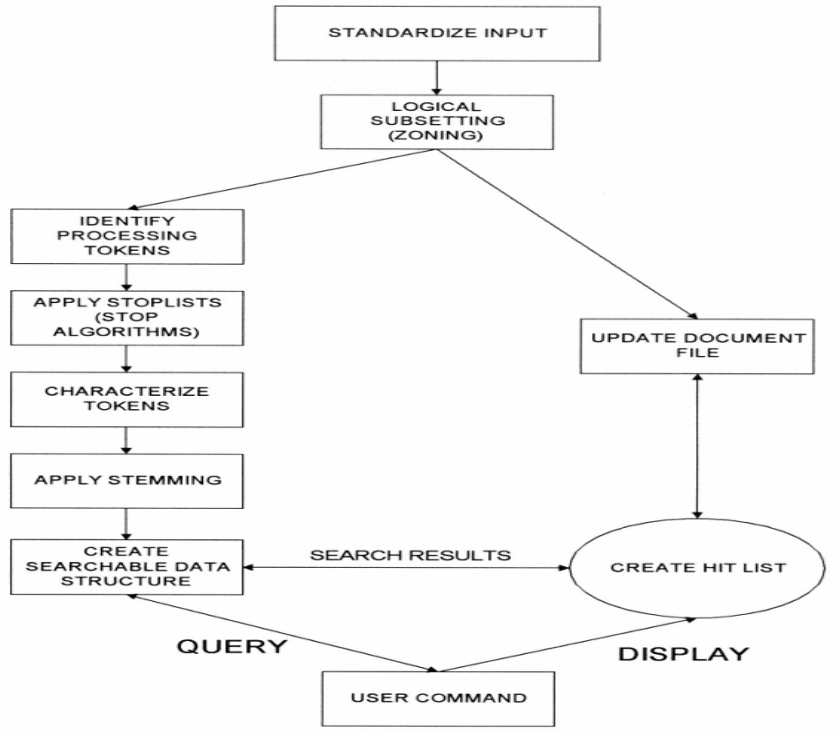


Figure 5.1 Data Flow in Information Processing System

Statistical strategies cover the broadest range of indexing techniques and are the most prevalent in commercial systems. The basis for a statistical approach is use of frequency of occurrence of events. The events usually are related to occurrences of processing tokens (words/phrases) within documents and within the database. The words/phrases are the domain of searchable values. The statistics that are applied to the event data are probabilistic, Bayesian, vector space, neural net. The static approach stores a single statistic, such as how often each word occurs in an item, that is used in generating relevance scores after a standard Boolean search. Probabilistic indexing stores the information that are used in calculating a probability that a particular item satisfies (i.e., is relevant to) a particular query. Bayesian and vector approaches store information used in generating a relative confidence level of an item's relevance to a query. It can be argued that the Bayesian approach is probabilistic, but to date the developers of this approach are more focused on a good relative relevance value than producing and absolute probability. Neural networks are dynamic learning structures that are discussed under concept indexing where they are used to determine concept classes.

Natural Language approaches perform the similar processing token identification as in statistical techniques, but then additionally perform varying levels of natural language parsing of the item. This parsing disambiguates the context of the processing tokens and generalizes to more abstract concepts within an item (e.g., present, past, future actions). This additional information is stored within the index to be used to enhance the search precision.

Concept indexing uses the words within an item to correlate to concepts discussed in the item. This is a generalization of the specific words to values used to index the item. When generating the concept classes automatically, there may not be a name applicable to the concept but just a statistical significance.

5.2 Statistical Indexing

Statistical indexing uses frequency of occurrence of events to calculate a number that is used to indicate

the potential relevance of an item. One approach used in search of older systems does not use the statistics to aid in the initial selection, but uses them to assist in calculating a relevance value of each item for ranking. The documents are found by a normal Boolean search and then statistical calculations are performed on the Hit file, ranking the output (e.g., term frequency algorithms). Since the index does not contain any special data.

Probabilistic systems attempt to calculate a probability value that should be invariant to both calculation method and text corpora. This allows easy integration of the final results when searches are performed across multiple databases and use different search algorithms. A probability of 50 per cent would mean that if enough items are reviewed, on the average one half of the reviewed items are relevant. The Bayesian and Vector approaches calculate a relative relevance value (e.g., confidence level) that a particular item is relevant. Quite often term distributions across the searchable database are used in the calculations. An issue that continues to be researched is how to merge results, even from the same search algorithm, from multiple databases. The problem is compounded when an attempt is made to merge the results from different search algorithms. This would not be a problem if true probabilities were calculated.

5.2.1 Probabilistic Weighting

The probabilistic approach is based upon direct application of the theory of probability to information retrieval systems. This has the advantage of being able to use the developed formal theory of probability to direct the algorithmic development. It also leads to an invariant result that facilitates integration of results from different databases. The use of probability theory is a natural choice because it is the basis of evidential reasoning (i.e., drawing conclusions from evidence). This is summarized by the Probability Ranking Principle (PRP) and its Plausible Corollary.

The source of the problems that arise in application of probability theory come from a lack of accurate data and simplifying assumptions that are applied to the mathematical model. If nothing else, these simplifying assumptions cause the results of probabilistic approaches in ranking items to be less accurate than other approaches. The advantage of the probabilistic approach is that it can accurately identify its weak assumptions and work to strengthen them. In many other approaches, the underlying weaknesses in assumptions are less obvious and harder to identify and correct.

There are many different areas in which the probabilistic approach may be applied. The method of logistic regression is described as an example of how a probabilistic approach is applied to information retrieval (Gey-94). The approach starts by defining a "Model 0" system which exists before specific probabilistic models are applied. In a retrieval system there exist query terms q_i and document terms d_i, which have a set of attributes $(v_1, \ldots, v_n)$ from the query (e.g., counts of term frequency in the query), from the document (e.g., counts of term frequency in the document) and from the database (e.g., total number of documents in the database divided by the number of documents indexed by the term).

The logistic reference model uses a random sample of query-document-term triples for which binary relevance judgments have been made from a training sample. Log O is the logarithm of the odds (logodds) of relevance for term t_k which is present in document D_j and query Q_i:

$$\log(O(R \mid Q_i, D_j, t_k)) = c_0 + c_1 v_1 + \ldots + c_n v_n$$

The logarithm that the i^{th} Query is relevant to the j^{th} Document is the sum of the logodds for all terms:

$$\log(O(R \mid Q_i, D_j)) = \sum_{k=1}^{q} [\log(O(R \mid Q_i, D_j, t_k)) - \log(O(R))]$$

where O(R) is the odds that a document chosen at random from the database is relevant to query Q_i. The coefficients c are derived using logistic regression which fits an equation to predict a dichotomous independent variable as a function of independent variables that show statistical variation (Hosmer-89). The inverse logistic transformation is applied to obtain the probability of relevance of a document to a query:

$$P(R \mid Q_i, D_j) = 1 \setminus (1 + e^{-\log(O(R \mid Q_i, D_j))})$$

The coefficients of the equation for logodds is derived for a particular database using a random sample of query-document-term-relevance quadruples and used to predict odds of relevance for other query-document pairs.

Gey applied this methodology to the Cranfield Collection (Gey-94). The collection has 1400 items and 225 queries with known results. Additional attributes of relative frequency in the query (QRF), relative frequency in the document (DRF) and relative frequency of the term in all the documents (RFAD) were included, producing the following logodds formula:

$$Z_j = \log(O(R \mid t_j)) = c_0 + c_1\log(QAF) + c_2\log(QRF) + c_3\log(DAF) + c_4\log(DRF)$$

$$+ c_5\log(IDF) + c_6\log(RFAD)$$

where QAF, DAF, and IDF were previously defined, QRF = QAF\ (total number of terms in the query), DRF = DAF\(total number of words in the document) and RFAD = (total number of term occurrences in the database)\ (total number of all words in the database). Logs are used to reduce the impact of frequency information; then smooth out skewed distributions. A higher maximum likelihood is attained for logged attributes.

The coefficients and log (O(R)) were calculated creating the final formula for ranking for query vector $\vec{Q}$, which contains q terms:

$$\log(O(R \mid \vec{Q})) = -5.138 + \sum_{k=1}^{\dot{}} (Z_j + 5.138)$$

The logistic inference method was applied to the test database along with the Cornell SMART vector system which uses traditional term frequency, inverse document frequency and cosine relevance weighting formulas. The logistic inference method outperformed the vector method.