UNIT - II

Cataloging and Indexing:

(i) History and objectives of Indexing
(ii) Indexing process
(iii) Automatic Indexing
(iv) Information Extraction.
(v) Data structure:
(vi) Introduction to Data Structure
(vii) Stemming Algorithms
(viii) Inverted file Structure
(ix) N-Gram Data Structures
(x) PAT Data Structure
(xi) Signature file Structure
(xii) Hypertex & XML Data Structures
(xiii) Hidden Markov models

I History and objectives of Indexing:

To understand the system design associated with creation & manipulation of the searchable data structures, it is necessary to understand the objectives of the indexing process.

Reviewing the history of indexing shows the dependency of information processing capabilities on manual & then automatic processing systems. most of the 1980's the goals of commercial IRS were constrained to facilitating the manual indexing paradigm. In the 1990's, exponential growth in computer processing capabilities with a continuing decrease in cost of computer sys has allowed inform IRS to implement the previously theoretical fun's introducing a new inform retrieval paradigm.

i) History of indexing:

Indexing (originally called cataloging) is the oldest technique for identifying the contents of items to assist in their retrieval. The objective of cataloging is to give access points to a collection that are expected & most useful to the users of the information.

The basic information required on an item, what is the item and what it is about, has not changed over the centuries.

As early as the third-millennium, in Babylon, libraries of cuneiform tablets were arranged by subject. upto 19th century there was little advancement in cataloging, only changes in the methods used to present the basic inform?

» In the late 1800s subject indexing became hierarchical (eg: Dewey Decimal system).

» In 1963, the Library of Congress initiated a study on the computerization of bibliographic surrogates.

» From 1966-68 the library of congress ran it MARC (machine Readable cataloging) standardizes the structure, contents and coding of bibliographic records.

Indexing (cataloging), was accomplished by creating a bibliographic citation in a structured file that references the original text. These files contain citation information about the item, keywording the subjects of the item, & in some systems a constrained length free text field used for an abstract/summary.

The indexing process is typically performed by professional indexers associated with library organizations. Throughout the history of libraries, this has been the most imp and difficult processing step. Most items are retrieved based upon what the item is used about. The user's ability to find items on a particular subject is limited by the indexer creating index terms for that subject.

The initial introduction of computers to assist the cataloguing fun did not change its basic oper of a human indexer determining those terms to assign a particular item. The standardization of data structures (eg: MARC format) did allow sharing of the indexes blw libraries. It reduced the manual overhead associated with maintaining a card catalog.

In the 1990s, the significant reduction in cost of processing power & memory in modern computers, along with access to the full text of an item from the publishing stages in electronic form, it allows use of the full text of an item as an alternative to the indexer-generated subject index. The indexer is no longer required to enter index terms that are redundant with words in the text of item.

(ii) Objectives of Indexing:

The objectives of indexing have changed with the evolution of IRS. Availability of the full text of the item in searchable form alters the objectives historically used in determining guidelines for manual indexing. The full text searchable data structure for items in the document file provides a new class of indexing called total document indexing.

previously indexing defined the source & major concepts of an item & provided a mechanism for standardization of index terms. i.e use of a controlled vocabulary.

A controlled vocabulary is a finite set of index terms from which all index terms must be selected (the domain of the index). In a manual indexing environment, the use of controlled vocabulary makes the indexing process slower, but potentially simplifies the search process. The extra processing time comes from the indexer trying to determine the appropriate index terms for concepts that are not specifically in the controlled vocabulary set.

uncontrolled vocabularies have the opposite effect, making indexing faster but the search process much more difficult.

The availability of items in electronic form changes the objectives of manual indexing. The source information can automatically be extracted. Most of the concepts discussed in the document is locatable via search of the total document index.

The words used in an item do not always reflect the value of the concepts being presented. It is the combination of the word & their semantic implications that contain the value of the concepts being discussed. The utility of the a concept is also determined by the user's need.

Public Indexing of the concept adds adds additional index terms over the words in the item to achieve abstraction. Private Index files are limiting the num of items indexed to user & Concepts bounded by the specific user's interest domain. There is overlap b/w the Private & Public Index files, but the Private Index file is indexing fewer concepts in an item than the Public Index file and the file owner uses his specific vocabulary of index terms.

The primary objective of representing the concepts within an item to facilitate the user's finding relevant information, electronic indexes to items provide a basis for other applications to assist the user. The format of the index, in most cases, supports the ranking of the o/p to present the items most likely to be relevant to the user's information needs first.

II Indexing process:

When an oranization with multiple indexers decides to create a public or private index some procedural decisions on how to create the index terms assist the indexers & end users in knowining 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 use scenarios of the end users. The other

The other decision is the need to link index terms together in a single index in a particular index concept. Linking index terms is needed when there are multiple independent concept found within an item.

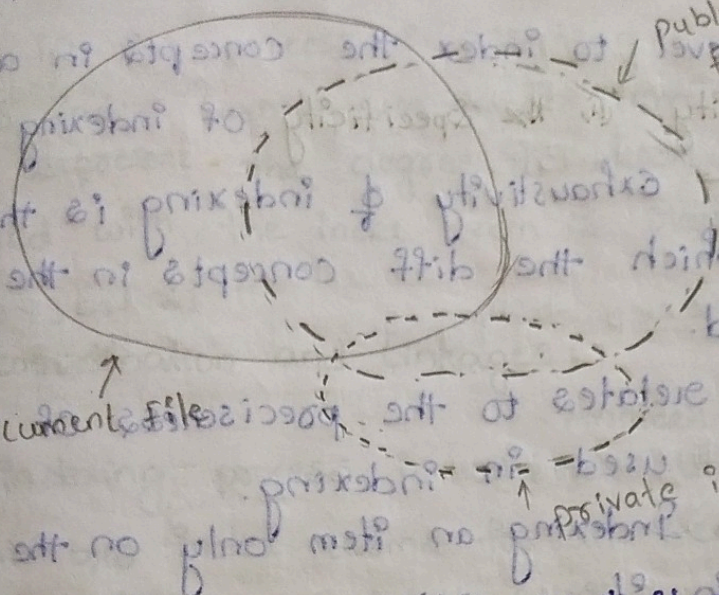


Fig: Items overlap b/w full item indexing, public file indexing & private file indexing.

(i) Scope of Indexing:

When performed manually, the process of reliably & consistently determining the bibliographic terms that represent the concepts in an item is extremely difficult. Problems arise from interaction of two sources: (i) Author (ii) Indexer.

Author → The vocabulary domain of the author may be different than that of the indexer, causing the indexer to misinterpret the emphasis & possibly even the concepts being presented.

Indexer → The indexer is not an expert on all areas & has different levels of knowledge in the diff. areas being presented in the item.

There are 2 factors involved in deciding on what level to index the concepts in an item: (i) Exhaustivity (ii) Specificity of indexing desired.

Exhaustivity → Exhaustivity of indexing is the extent to which the diff concepts in the item, are indexed.

Specificity relates to the preciseness of the index terms used in indexing.

Indexing an item only on the most imp concept in it & using general index terms yields low exhaustivity & specificity. This approach requires a minimal num of index terms per item & reduces the cost of generating the index. High exhaustivity and specificity indexes almost every concept in the item using as many detailed terms as needed.

Another decision on indexing is what portions of an item should be indexed. The simplest case is to limit the indexing to the Title & Abstract zones. This indexes the material that the author considers most important & reduces the cost associated with indexing an item.

Weighting of index terms is not common in manual indexing systems. Weighting

of index terms is not common in par

is the process of assigning an importance to an index term's use in an item. The weight should represent the degree to which the concept associated with the index term is represented in the item.

(ii) Precoordination and Linkages:

Another decision on the indexing process is whether linkages are available b/w index terms for an item. Linkages are used to correlate related attributes associated with concept discussed in an item. This process of creating term linkages at index creation time is called precoordination. when index terms are not coordinated at index time, the coordination occurs at search time. This is called postcoordination.

factors that must be determined in the linkage process are the num of terms that can be related, any ordering constraints in the linked terms, & any additional descriptors are associated with the index terms.

The below fig 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.

Index Terms

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)

Linked (Precoordination) with position indicating role

(US, introduction, oil refineries, Peru)

(SUBJECT : CITGO;
ACTION : drilling;
OBJECT : Oil, wells
MODIFIER : in Mexico)

Linked (Pre-coordination) with modifier indicating role

(SUBJECT : U.S.,
ACTION : introduces;
OBJECT : Oil refineries;
MODIFIER : in Peru)

fig : Linkage of Index Terms

The positional roles treat the data as a vector allowing only one value per position. When the modifiers are used, only one entry would be req.

methodology

III Automatic

capability fo
the index t
Simplest ca
used as pos
more compl
to emulate
num of in
the item.
Advantage:
determi
of a co
Disadvanto
are c

thesaurus
indexer's
exhaustiv
accuracy

(Ex: 300
nts per
caused
Ex:
Spelli

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 req when the objective is to emulate a human indexer & determine a limited num of index terms for the major concepts in the item.

Advantage: Human indexing are the ability to determine concept abstraction & judge the value of a concept.

Disadvantage: human indexing over automatic indexing are cost, processing time & consistency.

Processing time of an item by a human indexer varies significantly based upon the indexer's knowledge of concepts being indexed, the exhaustivity & specificity guidelines & the amount & accuracy of preprocessing via Automatic file Build. Even for relatively short items (Ex: 300-500 words) it normally takes at least 5 mints per item. A significant portion of this time is caused by the human interaction with the computer. Ex: typing speeds, cursor positioning, correcting spelling errors, talking breaks blw activities.

Automatic Indexing req only a few seconds to less of computer time based upon the size of the processor & the complexity of the algorithms to generate the index.

Another advantage of human indexers typically generate different indexing for the same document.

In Automatic Indexing, a sophisticated researcher understands the automatic process & be able to predict its utility & deficiencies, allowing for compensation for system characteristics in a search strategy. Even the end user, after interacting with the system, understands for certain classes of 'inform' & certain sources, the ability to find relevant items is worse than other classes & sources.

Indexing resulting from automated indexing fall into 2 classes: (i) weighted (ii) unweighted

(i) 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 occurance of the term in in the item. The values for the index terms are normalized blw 0 and 1.

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.

(ii) Unweighted Indexing System: In this, the existence of an index term in a document & sometimes its word locations are kept as part of the searchable data structure. No attempt is made to discriminate b/w the value of the index terms in representing concepts in the item. Looking at the index, it is not possible to tell the difference b/w the main topics in the item & a casual reference to a concept.

Automatic indexing can either try to preserve the original text of an item basing the final set of searchable index values on the original text or map the item into a completely different representation, called concept indexing, & the use of the concepts as a basis for the final set of index values.

(i) Indexing by Terms: 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.

i, Indexing by Term:

When the terms of the original item are used as a basis of the index process, there are 2 major techniques for creation of the index: 1. Statistical 2. natual language.

Statistical techniques can be used based upon vector models & probabilistic models with a special case being Bayesian models. They are classified as statistical because their calculation of weights use statistical inform^n such as the frequency of occurrence of words & their distributions in the searchable database.

Natural lang^g technique also use some statistical inform^n, but perform more complex parsing to define the final set of index concepts

weighted systems are discussed as vectorized information systems. The system emphasizes weights as a foundation for information detection & stores these weights in a vector form. Each vector represents a document & each position in a vector represents a different unique word 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 b/w the query vector & the document vectors.

In addition to 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 inform system. It could be applied as part of index term weighting, but usually is applied as part of the retrieval process by calculating the relationship b/w an item & specific query.

The below fig shows the basic weighting approach for index terms & association b/w query terms & index terms.

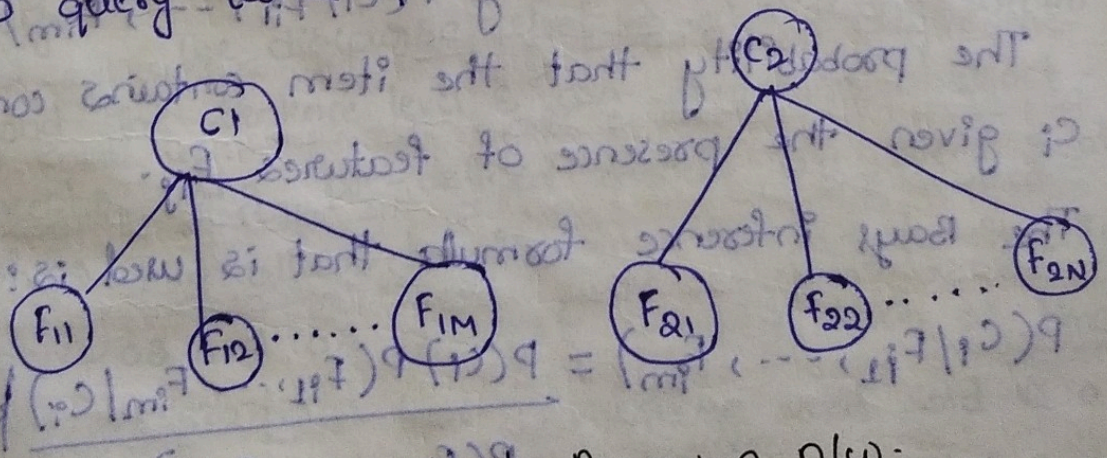


fig: Two-level Bayesian n/w.

Here the nodes C_1 & C_2 represents the

item contains concept C_i and 'F' Nodes represent the item has feature (e.g. word) F_{ij}. The n/w could also be interpreted as C representing concepts in a query & 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_i.
2. The conditional probability $P(F_{ij}/C_i)$ that the features F_{ij} where $j = 1, \ldots m$ are present in an item given that the item contains topic C_i.

The automatic index task is 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 Bays inference formula that is used is:

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

If the goal is to provide ranking as the result of a search by the posteriors, the Bayes

represent
The
representing
Concepts

the
that
e needed:
tem is
that
present
ins topic
calculate
im);
concept
is:
C_i) {
s the
Bayes

rule can be simplified to a linear decision rule:

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

where $I(F_{ik})$ is an indicator variable that equals 1, if and only if F_{ik} is present in the item (equal 0 otherwise) 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 other the order produced by the posterior probabilities.

Natural language technique:

The DR-LINK (Document Retrieval through Linguistic knowledge) system provides items at the morphological, lexical, semantic, syntactic, & discourse levels. Each level uses inform from the previous level to perform its additional analysis.

The discourse level is abstracting inform beyond the sentence level & can determine abstract concepts using pre-defined models of event relationships. This allows the indexing to include specific terms as well as abstract concepts such as time (ex: differentiates b/w a company was sold Vs a company will be sold). Normal automatic indexing does a poor job at identifying & extracting "verbs" & relationships b/w objects based upon the verbs.

(ii) Indexing by concept:

The basis for concept indexing is that there are many ways to express the same idea & increased retrieval performance comes from using a single representation. 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 inform in items.

The determined set of concepts does not have a label associated with each concept, but is a mathematical representation (ex: vector).

Ex: mathPlus.

These system uses neural n/w to facilitate machine learning of concept/word relationships & sensitivity to similarity & use. The system goal is to be able to determine from the corpus of items, word relationships (ex: synonyms) & the strength of these relationships & use that inform in generating content vectors.

Two neural n/w are used. One is learning algorithm generates stem content vectors that are sensitive to sly similarity of use &

another one is performs query modification based upon user feedback.

word stems, items & 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 class concept class. For any word stem k, its context vector V^k is an n-dimensional vector with each component j interpreted as follows:

V^k +ve if k is strongly associated with feature j

$V^k \approx 0$ if word k is not associated with feature j

V^k -ve if word k contradicts feature j.

IV. Information Extraction:

There are 2 process associated with inform^n extraction:

one is Determination of facts to go into structured fields in a database. And another is extraction of text that can be used to summarize an item.

In the 1st case only a subset of the imp facts in an item may be identified & 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 & extract index terms

that will go into a structured database.

An Information extraction system only analyzes those portions of a document that contain inform^n relevant to the extraction criteria. The objective of data extraction is in most cases to update a structured database with additional facts. The updates may be from a controlled vocabulary of substrings from the item as defined by the extraction rules. The term slot is used to define a particular category of inform^n to be extracted. Slots are organized into templates of Semantic frames. Inform^n extracts req multiple levels of analysis of the text of an item. It must understand the words & their context. The processing is very similar to the natural lang processing described under indexing. It's also used precision & recall. These are applied with slight modifications to their meaning. Recall refers to how much inform^n was extracted from an item vs how much should have been extracted from the item.

precision refers to how much inform^n was extracted accurately vs the total inform^n extracted.

Additional metrics used are overgeneration & fallout.
Overgeneration → measures the amount of irrelevant inform^n that is extracted.
Fallout → measures how much a system assigns incorrect slot fillers as the num of incorrect slot fillers increases.
These measures are applicable to both human & automated extraction process.

Another related inform^n 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 imp ideas while significantly reducing the size.

(iii) Multimedia Indexing:

The automated indexing takes place in multiple passes of the information (moves) vs just a direct conversion to the indexing structure. The 1st pass in most cases is a conversion from the analog i/p mode into a digital structure (signal).
Indexing video & images can be accomplished at the raw data level (ex: aggregation of raw pixels), the feature level distinguishing primitive attributes such as color & luminance, & at the semantic level where meaningful objects are recognized (brightness, images).

consider an analog audio i/p, the system will convert the audio to digital format and determine the phonemes associated with the utterances. The phonemes will be used as i/p to a Hidden Markov Search model, That will determine with a confidence level the words that were spoken.

In addition to storing the extracted index searchable data, a multimedia item needs to also store some mechanism to correlate the diff modalities during search. There are 2 main mechanisms that are used, (i) Positional (ii) Temporal.

Positional: is used when the modalities are interspersed in a linear sequential composition.

Ex: A document that has images & audio inserted can be considered a linear structure & the only relationship between the modalities will be the juxtaposition of each modality.

Temporal: It is based upon time because the modalities are executing concurrently. The typical video source of television is inherently a multimedia source. It contains video, audio, & potentially closed captioning. The creation of multimedia presentations are becoming more common using the Synchronized multimedia Integration Language (SMIL). It is a mark-up lang designed to support multimedia presentations that integrate text (eg: from slides or free running text) with audio, videos, images.

data struct

I Introduction

data struc

one structu

in their

this struct

other stru

associated

data stru

Data structure

Introduction to data structures:

There are a major data structures in any information system. one structure stores & manages the received items in their normalized form. The process supporting this structure is called the "document manager." other structure contains the processing tokens & associated data to support search.

In below fig shows the expanded major data structure.

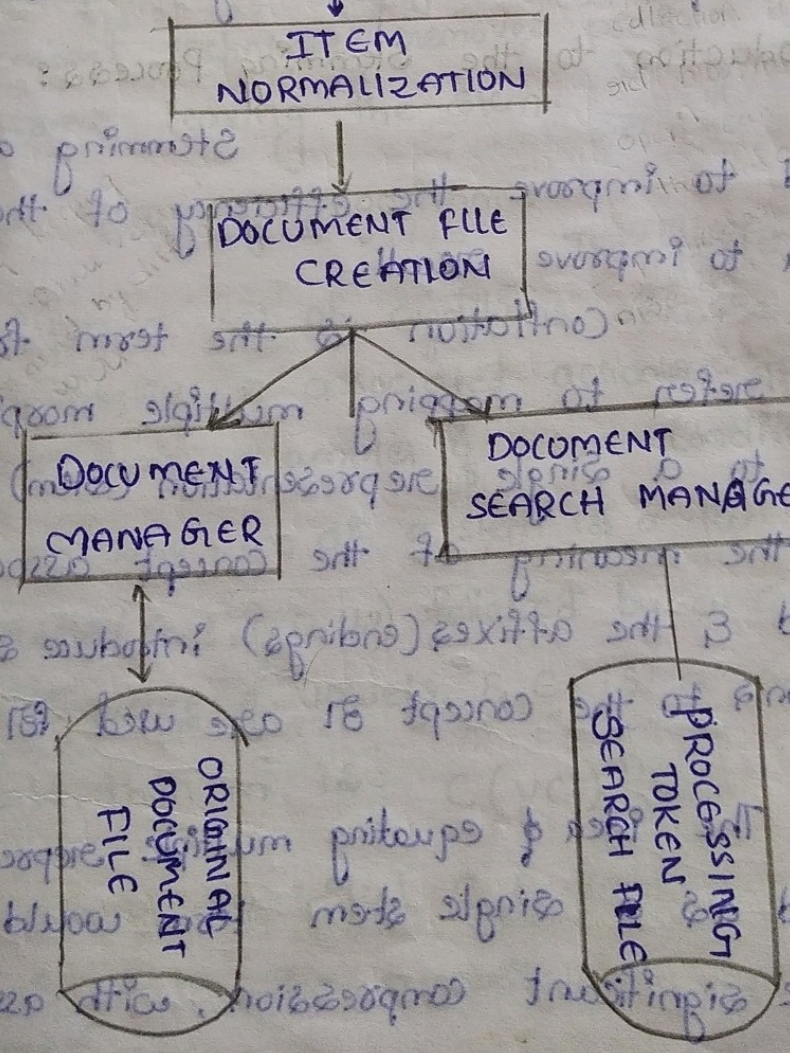


fig: major Data Structure.

Data structures used to support the search function. It does not address the document management function nor the data structures & other related theory associated with parsing of queries.

VI Stemming Algorithms:

The concept of stemming has been applied to information systems from their initial automation in the 1960's. The original goal of stemming was to improve performance & require less system resources by reducing the num of unique words that a system has to contain.

(i) Introduction to the Stemming Process:

Stemming algorithms are used to improve the efficiency of the inform? system & to improve recall.

Conflation is the term frequently used to refer to mapping multiple morphological variants to a single representation (stem). The stem carries the meaning of the concept associated with word & the affixes (endings) introduce subtle modifications to the concept or are used for syntactical purpose.

The idea of equating multiple representations of a word as a single stem term would appear to provide significant compression, with associated savings in storage & processing.

Ex: The stem "comput" could associate "computable

computability, computation, computer, computational, computed, computerize", to one compressed word. The stem can be identified by set of words.

The most common stemming algorithm removes suffixes & prefixes, some times recursively to derive the final stem. Other techniques such as table lookup & successor stemming provide alternatives that req additional overheads.

successor stemmers determine prefix overlap as the length of a stem is increased.

Table lookup req a large data structure. The affix removal technique removes prefixes & suffixes from terms leaving the stem.

(ii) Porter stemming Algorithm:

The porter Alg is based upon a set of conditions of the stem, suffix & prefix & associated actions given the condition. Some examples of stem conditions are:

1. The measure, m, of a stem is a fun? of sequences of vowels (a, e, i, o, u, y) followed by a consonant. If V is a sequence of vowels & C is a sequence of consonants, then m is: $C(VC)^m V$

where the initial C & final V are optional & m is the number VC repeats.

measure of word ↙

measure	Example
m = 0	free, why
m = 1	frees, whose
m = 2	prologue, compute

2. *<X> – stem ends with letter X
3. *v* – stem contains a vowel
4. *d – stem ends in double consonant
5. *o – stem ends with consonant - vowel - consonant sequence where the final consonant is not w, x, or y.

Suffix conditions take the form current_suffix = pattern.

Actions are in the form old_suffix → new_suffix

Rules are divided into steps to define the order of Applying the rules. The following are some 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	inflat(ed) → inflate
1c	*v*	y	i	happy → happi
2	m>0	aliti	al	formaliti → formal
3	m>0	icate	ic	duplicate → duplic
4	m>1	able	NULL	adjustable → adjust
5a	m>1	e	NULL	inflate → inflat
5b	m>1 and *d and *<L>	NULL	single letter	controll → control

Given the below word "duplicatable", the following are the steps in the stemming process:

duplicat rule 4
duplicate rule 1b1
duplic rule 3

iii, Dictionar

still may
those that
ing plural
consistent
addressed.
of the t
replaced
this tec
INQUERY

stemming
morpholo
to a root
words
root.
to "me
not
K stem
Natur
return
forms
in th
form

iii, Dictionary look up stemmers

In this Approach, simple stemming rules still may be applied. The rules are taken from those that have the fewest exceptions (e.g., removing pluralization from nouns). But even the most consistent rules have exceptions that need to be addressed. The original term or stemmed version of the term is looked up in a dictionary and replaced by the stem that best represents it. This technique has been implemented in the INQUERY and Retrievalware systems.

The INQERY system uses a stemming technique called Kstem. Kstem is a morphological analyzer that conflates word variants to a root form. It tries to avoid collapsing words with different meanings into the same root. For example, "memorial" and "memorize" reduce to "memory". But "memorial" and "memorize" are not synonyms and have very different meanings. Kstem, like other stemmers associated with Natural Language processors and dictionaries, returns words instead of truncated word forms. Generally, Kstem requires a word to be in the dictionary before it reduces one word form to another. Some endings are always removed,

even if the root form is not found in the dictionary (e.g., 'ness' 'ly'). If the word being processed is in the dictionary, it is assumed to be unrelated to the root after stemming and conflation is not performed (e.g., 'factorial' needs to be in the dictionary or it is stemmed to 'factory'). For irregular morphologies, it is necessary to explicitly map the word variant to the root desired (for example, "matrices" to "matrix").

The Kstem system uses the following six major data files to control and limit the stemming process:

1. Dictionary of words (lexicon)
2. Supplemental list of words for the dictionary
3. Exceptions list for those words that should retain an "e" at the end (e.g., "suites" to "suite" but "suited" to "suit")
4. Direct_Conflation - allows definition of direct conflation via word pairs that override the stemming algorithm
5. Country_Nationality - Conflations between nationalities and countries (British maps to Britain)
6. Proper Nouns - a list of proper nouns that should not be stemmed.

The strength of the Retrieval ware System lies in its Thesaurus / semantic Network support data structure that contains over 400,000 words. The dictionaries that are used contain the morphological variants of words. 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.

(iv) Successor stemmers

Successor stemmers are based upon the length of prefixes that optimally stem expansions of additional suffixes. The algorithm is based upon analogy in structural linguistics that investigated word and morpheme boundaries based upon the distribution of phonemes, the smallest unit of speech that distinguish one word from another. The process determines the successor varieties for a word, uses this information to divide a word into segments and selects one of the segments as the stem. The Successor variety of a segment of a word in a set of words in the no. of distinct letters that occupy the segment length plus

one character, for example, the successor variety for the first three letters (i.e. word segment) of a five-letter word is the no. of words that have the same first three letters but a different fourth letter, plus one for the current word. A Graphical representation of successor variety is shown in a symbol tree.

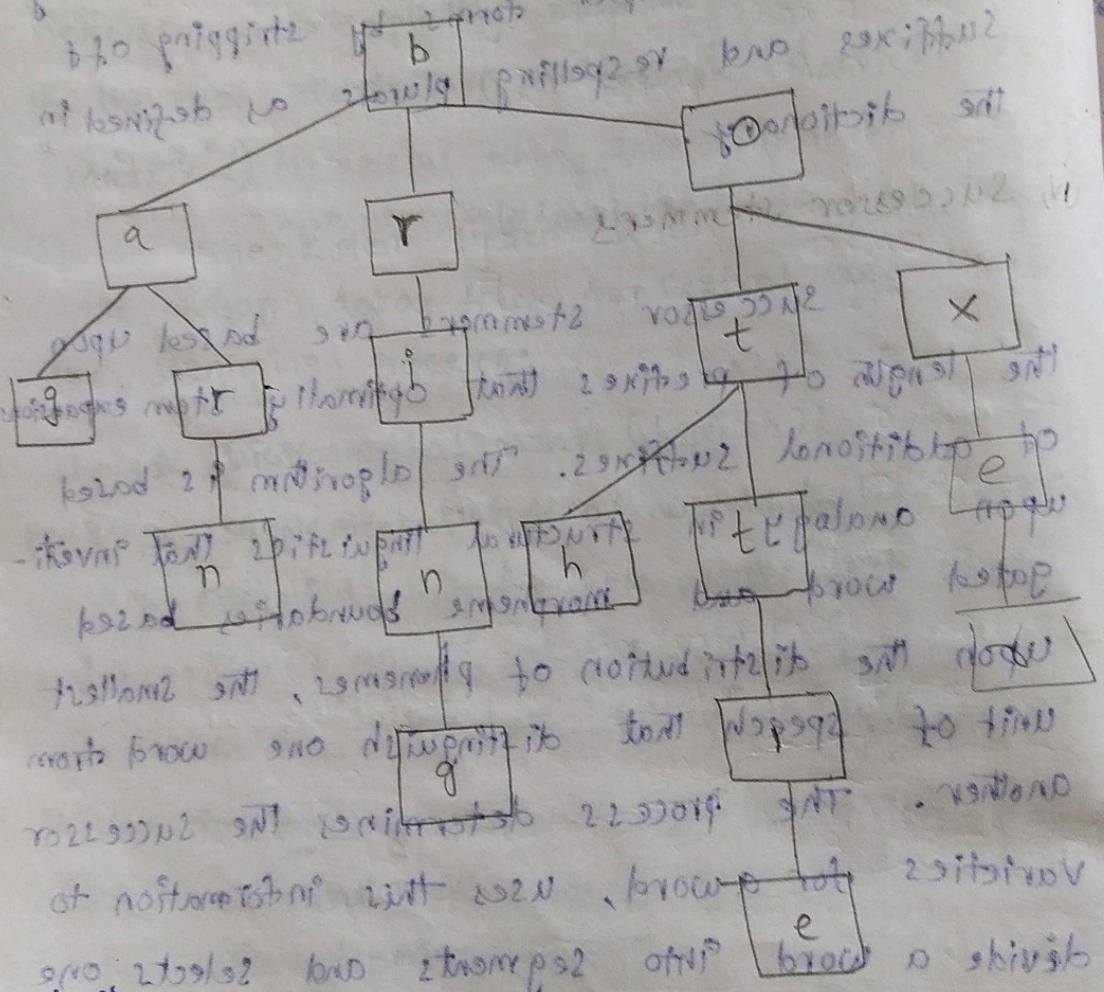


fig shows

The symbol of tree for the terms bag, barn, bring, both, box and bottle. The successor variety for any prefix of a word is the num of children

that are associated with the node in the symbol tree representing that prefix.

Example: The successor variety for the first letter b is three. The successor variety for the prefix "ba" is 2.

The successor varieties of a word are used to segment a word by applying one of the following four methods:

1. Cutoff method: a cutoff value is selected to define stem length. The value varies for each possible set of words.
2. Peak & plateau: a segment break is made after a character whose successor variety exceeds that of the character immediatly preceding it & 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 num of words begining with the k length sequence of letters a. Let $|D_{akj}|$ be the number of words in D_{ak} with successor j.

The probability that a member of D_{ak} has the successor j is given by $\frac{|D_{akj}|}{|D_{ak}|}$. The entropy 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 & its predecessors. A cutoff value is selected & a boundary is identified whenever the cutoff value is reached.

Hafer & Weiss experimented with the techniques, discovering that combinations of the techniques performed best, which they used in defining their stemming process.

The additional word "boxer", the successor variety stemming in shown below.

PREFIX	Successor Variety	Branch letters
b	3	a,e,o
bo	2	t,x
box	1	e
boxe	1	r
boxer	0	blank

fig: Successor variety stemming

If the cutoff method, with value 4 was selected then the stem would be "boxe".

peak & plateau method can't 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.

After a word has been segmented, the

segment to be used as the stem must be selected. Hafer & 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 & Weiss noted that multiple prefixes in the English language do not occur often & thus selecting the 1st or 2nd segment in general determines the appropriate stem.

» Conclusions:

Frakes summarized studies of various stemming studies. He cautions that some of the authors failed to report test statistics, especially sizes, making interpretation difficult. Also some of the test sample size were so small as to make their results questionable.

Frakes came to the following conclusions:

» Stemming can affect retrieval & where effels were identified they were +ve. There is a little diff b/w retrieval effectiveness of diff full stemmers with the exception of the Hafer & Weiss stemmer.

» Stemming is as effective as manual conflation.

» Stemming is depending upon the nature of the vocabulary

Paice has defined a stemming performance measure called Error Rate Relative to Truncation (ERRT) that can be used to compare stemming algorithms.

The approach depends upon the ability to partition terms semantically & morphologically related to each other into "concept groups". After applying a stemmer that is not perfect, concept groups may still contain multiple stems rather than one. This introduces an error reflected in the Understemming Index (UI). Also it is possible that the same stem is found in multiple groups. This error state is reflected in the overstemming Index (OI).

UI and OI values can be calculated based upon truncated word lengths. The perfect case is where UI & OI equal zero. ERRT calculated as the distance from the origin to the (UI, OI) coordinate of the stemmer being evaluated (OP) versus the distance from the origin to the worst case intersection of the line generated by pure truncation (OT).

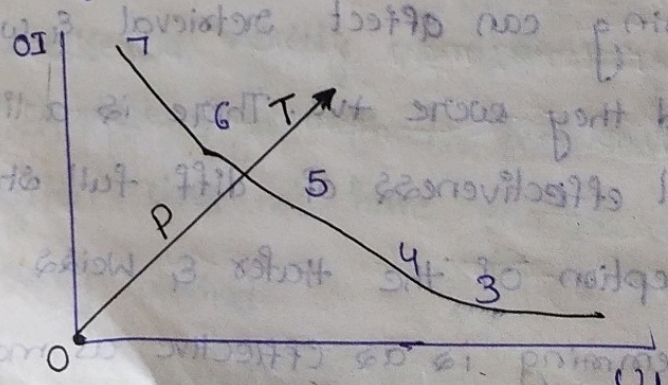


fig: Computation of ERRT value.

VII. Inverted file structure:

The most common data structure used in both database management system & IRS is the inverted file structure.

These are composed of 3 basic files:

(i) The document file,
(ii) The inversion lists (posting files)
(iii) The dictionary.

The name inverted file comes from its methodology of storing an inversion of the documents. Each document in the system is given a unique numerical identifier. It is that identifier that is stored in the inversion list. The way to locate the inversion list for a particular word is via the dictionary.

The Dictionary is typically a stored sorted list of all unique words in the system & a pointer to the location of its inversion list. Dictionaries can also store other information used in query optimization such as the length of inversion lists.

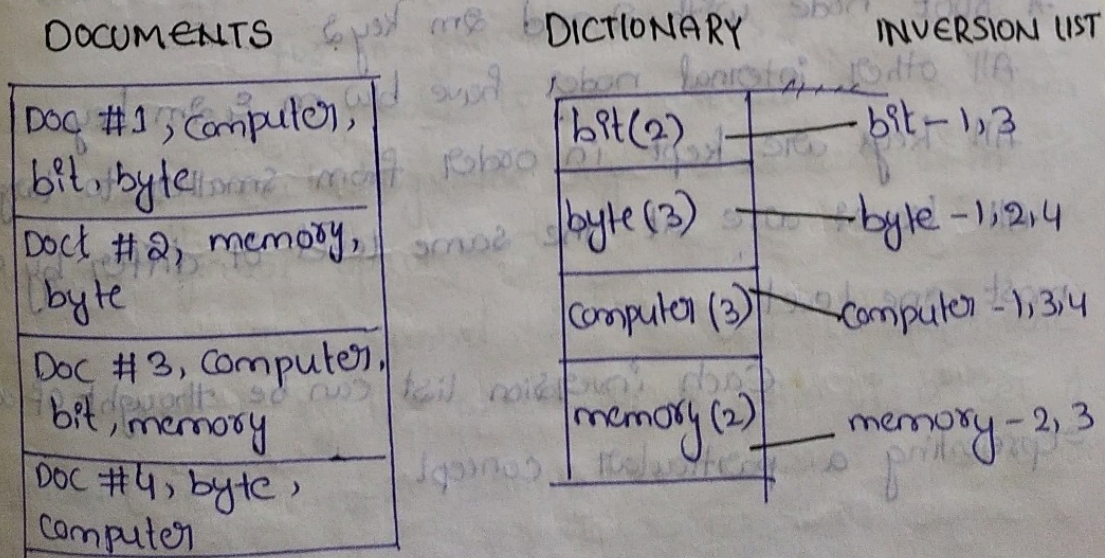


fig: Inverted File Structure

Thus if the word "bit" was the tenth, 12th & 18th word in document #1, then the inversion list would appear:

bit - 1(10), 1(12), 1(18)

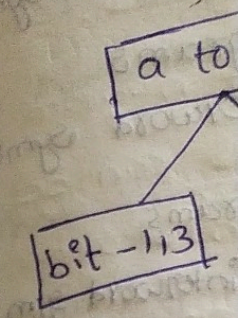
weights can also be stored in inversion lists. when a search is performed, the inversion lists for the terms in the query are located & the appropriate logic is applied b/w inversion lists. The result is a final hit list of items that satisfy the query. For systems that support ranking, the list is reorganized into ranked order.

Inverted file structure fig; the query (bit AND computer) would use the Dictionary to find the inversion lists for "bit" & "computer". These 2 lists would be logically ANDed: (1,3) AND (1,3,4) resulting in the final hit list containing (1,3).

Rather than using a dictionary to point to the inversion list, B-trees can be used. 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 2 and 2m keys
All other internal nodes have b/w m & 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.

Each inversion list can be thought of as representing a particular concept.



VIII

N - Gr

viewed a

& as a

N-Gram

character

of char

transfor

then use

example

in belo

allow

set u

interwo

the

symbo

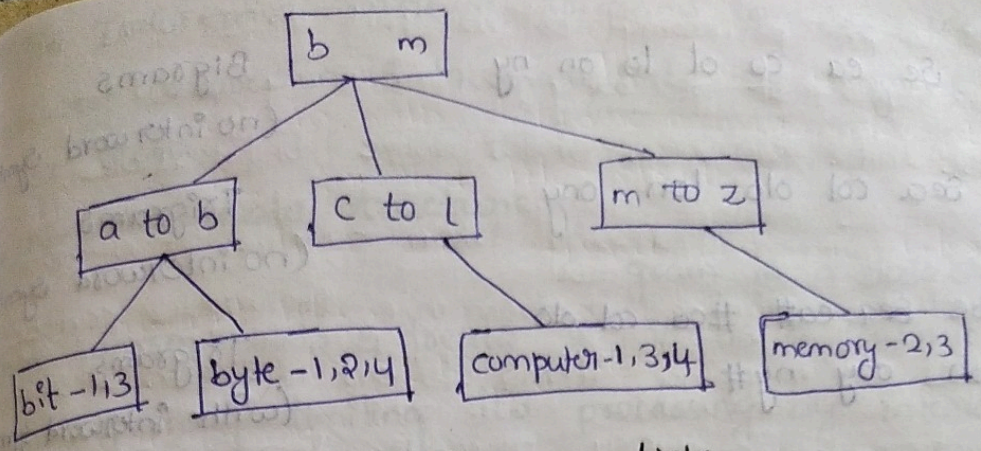


fig: B-Tree Inversion Lists.

viii. N-Gram Data Structures:

N-Grams can be viewed as a special technique for conflation (stemming) & as a unique data structure in information systems. N-Grams are fixed length consecutive series of "n" characters. Instead they are based upon a fixed num of characters. The searchable data structure is transformed into overlapping n-grams, which are then used to create the searchable database.

Example: bigrams, trigrams & pentagrams are given in below fig for the word phrase "sea colony".

For n-grams, with $n > 2$, 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 symbol (ex: semicolon, colon, etc).

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 sym #)
#sea# #colo colon olony lony#	pentagrams (with interword sym #)

fig: Bigrams, Trigrams & Pentagrams for "sea colony"

ii) History:

The first use of n-grams dates to World War II when it was used by cryptographers. Fletcher Pratt states that "with the backing of bigram & trigram tables any cgr cryptographer can dismember an simple substitution cipher."

It doesn't follow the normal definition of stemming because what is produced by creating n-grams are word fragments Vs semantically meaningful word stems.

Another major use of n-grams is in spelling error detection & correction. Damerau specified 4 categories of spelling errors shown in fig.

Error category	Example
Single character Insertion	Compuuter
single character Deletion	compter
Single character Substitution	Compiter
Transposition of 2 adjacent characters	Comptuer

fig: categories of spelling errors.

In Information retrieval, trigrams have been used for text compression & to manipulate the length of index terms.

N-Gram Data structure:

n-gram is a data structure that ignores words & treats the i/p as a continuous data, optionally limiting its processing by inter word symbols. The data structure consists of fixed length overlapping symbol segments that define the searchable processing tokens. These token have logical linkages to all the items in which the tokens are found.

Inversion lists, document vectors & other proprietary data structures are used to store & the linkage data structure & str are used in the search process. In some cases just the least frequently occuring n-gram is kept as part of a 1st pass search process.

The advantage of n-grams is that they place a finite limit on the num of searchable tokens.

$$maxSeg_n = (\lambda)^n$$

The max number of unique n-grams that can be generated, maxSeg, can be calculated as a fun of n which is the length of the n-grams, & λ which is the number of processable symbols from the alphabet.

Although there is a savings in the num

of unique processing tokens & implementation techniques allow for fast processing on minimally sized machines, false hits can occur under some architectures.

Processing token bounds of n-gram data structures, optimized performance techniques can be applied in mapping items to an n-gram searchable structure & in query processing. There is no semantic meaning in a particular n-gram since it is a fragment of processing token & may not represent a concept. Thus n-grams are a poor representation of concept & their relationships.

But the juxtaposition of n-grams can be used to equate to standard word indexing, achieving the same levels of recall and within 85% precision levels with a significant improvment in performance. Vector representations of the n-grams from an item can be wed to calculate the similarity b/w items.

[IX PAT Data Structure]

IX Signature file structure:

The goal of signature file structure is to provide a fast test to eliminate the majority of items that are not related to query. The items that satisfy the test can either be evaluated by another search algorithm to

eliminate addi
wer to review
in a highly
fast test.

Bec

and unordered
than an inve
concatenated
ficant inversi

information c
items in pla
search is a
producing
size.

created v
based upo
ped into a

requires
signatures
signatures

advantage
the column
columns

eliminate additional false hits & delivered to the user to review. The text of the items is represented in a highly compressed form that facilitates the fast test.

Because file structure is highly compressed and unordered, it requires significantly less space than an inverted file structure & new items can be concatenated to the end of the structure Vs the significant inversion list update.

Since items are seldom deleted from information data bases, it is typical to leave deleted items in place & mark them as deleted. Signature file search is a linear scan of the compressed Vs of items producing a response time linear with respect to file size.

The surrogate signature search file is created via superimposed coding. The coding is based upon words in the item. The words are mapped into a "word signature".

Search of the signature matrix requires O(N) search time. Other techniques are a level signatures & use of B-tree structures with similar signatures clustered at leaf nodes.

Another implementation approach takes advantage of the fact that searches are performed on the columns of the signature matrix, ignoring those columns that are not indicated by hashing of any of

the search terms. Thus the signature matrix may be stored in column order Vs row order, called vertical partitioning. This is in effect storing the signature matrix using an inverted file structure. Signature files provide a practical soln for storing & locating information in a number of different situations. These files have been applied as medium size databases, databases with low frequency of terms, parallel processing machines, & distributed environments.

PAT Data structure:

Using n-grams with interaction symbols included blw valid processing tokens equates to a continuous text i/p data structure that is being indexed in contiguous "n" character tokens. A different view of addressing a continuous text i/p data structure comes from PAT trees & PAT array. The i/p stream is transformed into a searchable data structure consisting of substrings.

The original concepts of PAT tree data structures were described as patricia trees & have gained new momentum as possible structure for searching text & images & applications in genetic databases.

The name PAT is short for PAtricia Trees. PATRICIA stands for Practical Algorithm TO Retrieve Information coded In Alphanumerics.

In creation of PAT trees each position

in the i/p string is the anchor point for a sub-string that starts at the point & includes all new text up to the end of the i/p. All substrings are unique. This view of text lends itself to many diff search processing structures.

A substring can start at any point in the text & can be uniquely indexed by its starting location & length. If all strings are to the end of the i/p, only the starting location is needed since the length is the difference from the location & the total length of the & ilem. It is possible to have a substring go beyond the length of the i/p stream by adding additional null characters.

These substrings are called sistring. (Semi-infinite string). The below fig shows some possible sistrings for an i/p 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	Wsa is complex.
Sistring 30	ex.

fig: 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 (0) zeros branching left & (1) 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.

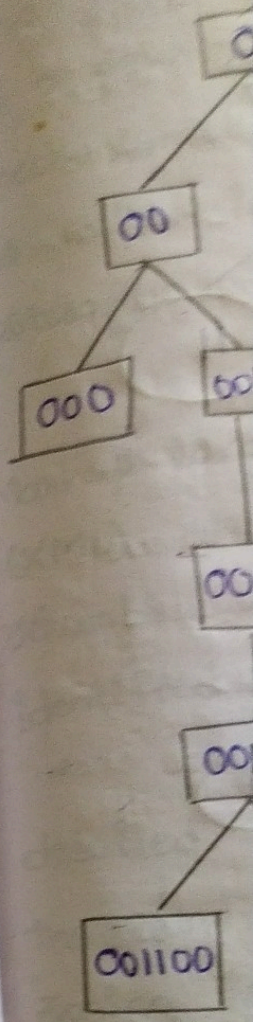
The key values are stored at the leaf nodes (bottom nodes) in the PAT tree. For a text i/p of size "n" there are "n" leaf nodes and "n-1" at most higher level nodes.

The below fig shows an example of the sistrings used in generating a PAT tree.

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

fig: Sistrings for input "100110001101".

If the binary representations of h is (100), o is (110), m is (001) and e is (101). then the word home produces the i/p 100110001101... using this sistrings, the full PAT binary tree is shown in below fig:



fig

are in

fig In
nodes (
to skip
differen

Comparis
ensure t

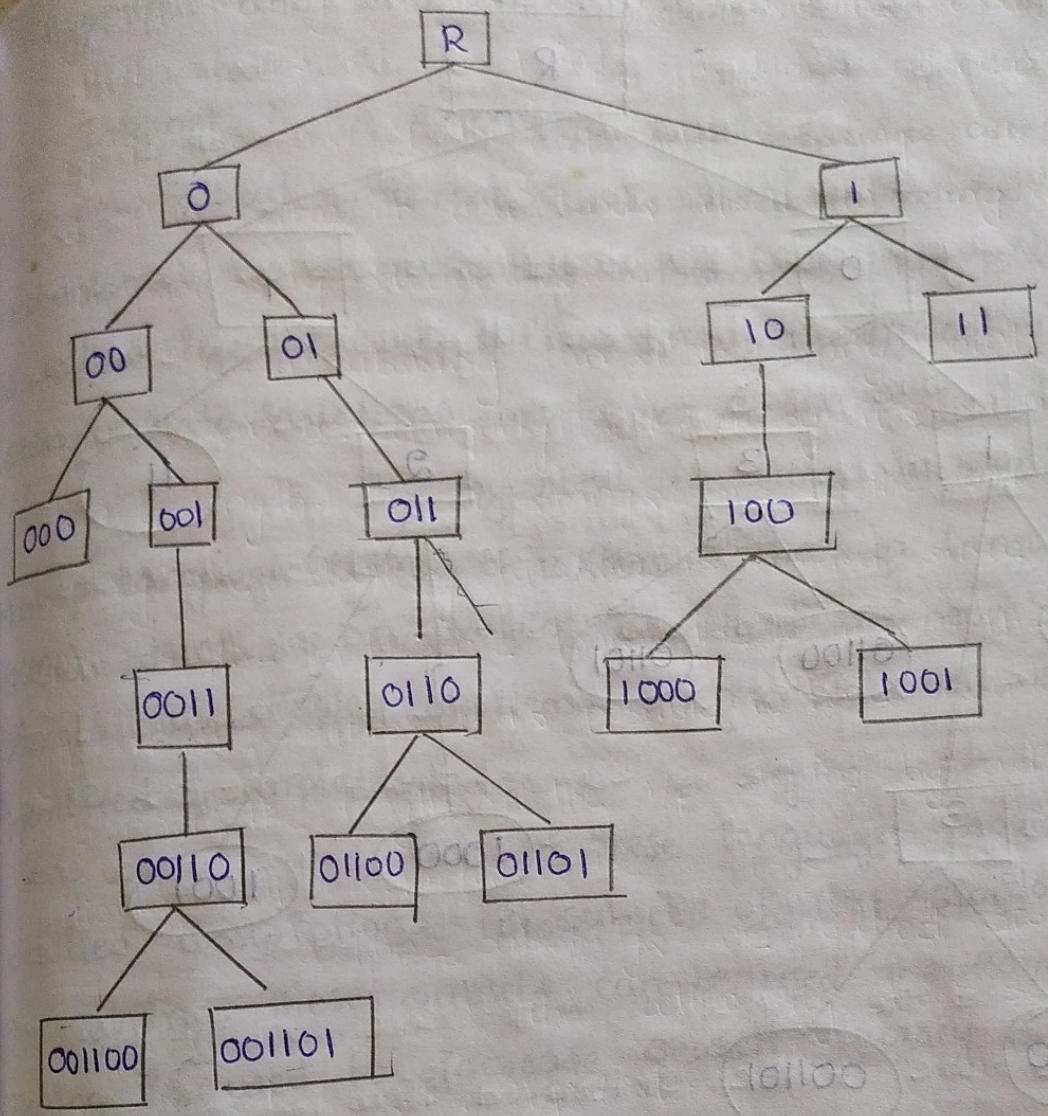


fig: PAT Binary tree for input "10011000l101".

A more compact tree where skip values are in the intermediate nodes is shown in below fig. In this version the value in the intermediate nodes (indicated by rectangles) is the number of bits to skip until the next bit to compare that causes difference b/w similar terms.

This final version saves space but req comparing a search value to the leaf node contents to ensure the skipped bits match the search term (ie skipped bits are not compared).

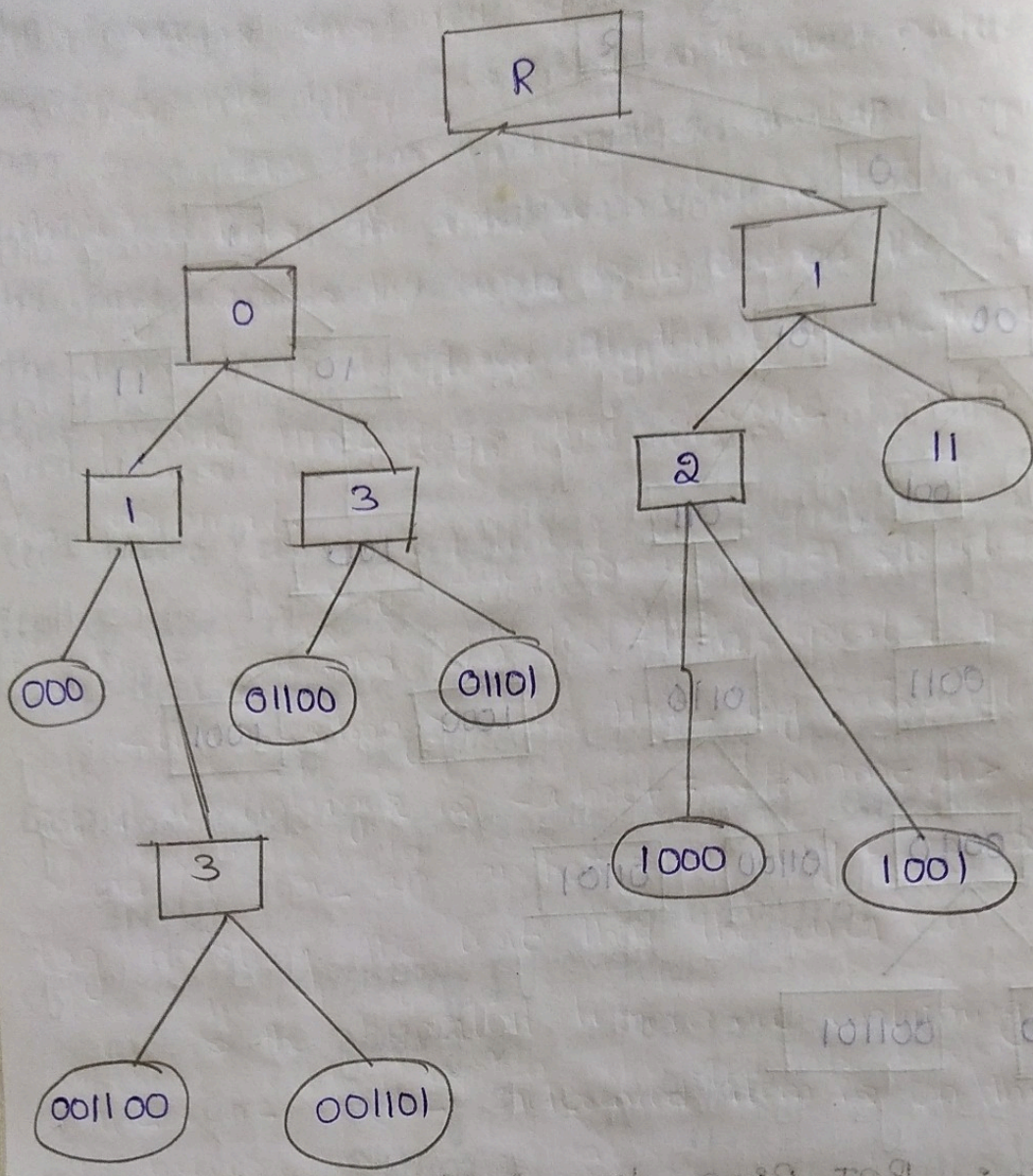


fig : PAT TREE skipping bits for "10011000110l"

PAT trees & arrays provide an alternative structure if string searching is the main goal. They store the text in an alternative structure supporting string manipulation. This structure does not have facilities to store more abstract concepts & their relationships associated with an item.

Hypertext and XML Data Structures:

The advent of the Internet & its exponential growth & wide acceptance as a new global information n/w has introduced new mechanisms for representing information. This structure is called hypertext & differs from traditional inform? storage data structures in format & use.

The hypertext is stored in Hypertext Markup Language (HTML) & eXtensible Markup Language (XML). HTML is an evolving standard as new requirements for display of items on the Internet are identified & implemented.

Both of these languages provide detailed descriptions for subsets of text similar to the zoning. These subsets can be used the same way zoning is used to increase search accuracy & improve display of hit results.

i) Definition of Hypertext structure:

The Hypertext data structure is used extensively in the Internet environment & 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 & the reference pointer is called a link. The referenced item can be of the same or a different data type than the original.

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 b/w less-than, greater than symbol (ex: <title>,). The HTML tag associated with hypertext linkages is <a href=... #NAME /a> where "a" and "/a" are an anchor start tag & 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 (URL - Uniform Resource Locator) & 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 3 components:

(i) The access method the client used to retrieve the item.

(ii) The Internet address of the server where the item is stored.

(iii) The address of the item at the server.

Ex: The URL for the HTML specification appears:

http://info.cern.ch/hypertext/WWW/markup/HTML.html

"HTTP" stands
is the acces
HTML. Othe
activities suc
search syste
collaborative

which is th
"ch." & "/hyp
to find the

linkages. Th
that can be
linkages to
This is si
upon reac
or skip to
a "generali

(iii) Hyperf

been aro
written b
describi

"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 (telnet://) & collaborative newsgroups (news://).

The distination point is found "info.cern.ch" which is the name of the "info" machine at CERN with "ch" & "/hypertext/www/markup/HTML.html" defines where to find the file HTML.html.

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."

(ii) Hypertext history:

The concept of hypertext has been around for over 50 years. In 1945, an article written by Vannevar Bush in 1933 was published describing the memex (memory extender) system.

It was a microfilm based system

that would allow the user to store much of the information from the scientific explosion of the 1940s on microfilm and retrieve it at multiple readers at the user's desk via individuals links.

The term "hypertext" came from Ted Nelson in 1965. Nelson's vision of all the world's literature being interlinked via hypertext references is part of his Xanadu system.

The lack of cost effective computers with sufficient speed and memory to implement hypertext effectively was one of the main inhibitors to its development. One of the first commercial uses of a hypertext system was the mainframe system, Hypertext Editing System, developed at Brown University by Andries van Dam and

Hypertext became more available in the early 1990's via its use in CD-ROMs for a variety of educational and entertainment products. Its current high level of popularity originated with it being part of the specification of the WWW by the CERN (the European Center for Nuclear Physics Research) in Geneva, Switzerland. The Mosaic browser, freely available from CERN on the Internet, gave everyone who had access the ability to receive & display hypertext documents.

XML :

The eXtensible markup Language (XML) is starting to become a standard data structure on the WEB. The logical data structure within XML is defined by a Data Type Description (DTD) & is not constrained to the 70 defined tags & 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 developing 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 blw them. This will include the Platform for Internet Content Selection (PICS) for attaching labels to material for filtering (eg., unsuitable for children).

Hypertext links for XML are being defined in the Xlink (XML linking Language) and Xpoint (XML pointer language) specifications. This will allow for distinction for different types of links to locations within a document and external to the document. This will allow an application to know if a link is just a repositioning reference within an item or link to another document that is an extension of the existing document. This will help in determining what needs to be retrieved to define the total item to be indexed.

Finally XML will include an XML Style Sheet linking definition to define how to display items on a particular style sheet and handle cascading style sheets. this will allow designers to limit what is displayed to the user and allow expansion to the whole item if desired.

(xii) Hidden Markov Models: (HMM)

HMM have been applied for the last 20 years to solving problems in speech recognition and to a lesser extent in the areas locating named entities, optical character recogonition & topic identification.

Its have been applied more generally

to information retrieval search with good results. One of the 1st comphrehensive & practical descriptions of Hidden markov Models was written by Dr. Lawrence Rabiner.

A HMM can best be understood by 1st defining a discrete markov process. A more formal definition of a discrete Hidden markov model is summarized by mitlendrof and schauble. as consisting of the following:

1. $S = \{S_0, \cdots S_{n-1}\}$ as a finite set of states where S_0 always denotes the initial state. Typically the states are interconnected such that any state can be reached from any other state.
2. $V = \{v_0, \cdots, v_{m-1}\}$ is a finite set of o/p symbols. This will correspond to the physical o/p from the system being modeled.
3. $A = S \times S$ a transition probability matrix where a_{ij} represents the probability of transitioning from state i to state j such that $\sum_{j=0}^{n-1} a_{i,j} = 1$ for all $i = 0, \cdots, n-1$. Every value in the matrix is a +ve value between 0 and 1. For the case where every state can be reached from every other state every value in the matrix will be non-zero.

4. $B = S \times V$ is an o/p probability matrix where element $b_{j,k}$ is a fun? determining the probability and $\sum_{k=0}^{m-1} b_{j,k} = 1$ for all $j = 0, \ldots n-1$.

5. The initial state distribution.

The HMM will generate an o/p symbol at every state transition. The transition probability is the probability of the next state given the current state. The o/p probability is the probability that a given o/p is generated upon arriving at the next state.

It can be used as both a generator of possible sequences of o/p & their probabilities & given a particular out sequence it can model its generation by an appropriate HMM model. The complete specification of a HMM requires specification of the states, the o/p symbols & 3 probability measures for the state transitions, o/p probability fun's & the initial states. The distributions are frequently called A, B & Π, and the following notation is used to define the model:

$$\lambda = (A, B, \Pi).$$

One of the primary problems

associated with HMM is how to efficiently calculate the probability of sequence of observed o/p given the HMM model.

—— o ——