UNIT-II

CATALOGING AND INDEXING

Cataloging and Indexing: History and Objectives of Indexing, **Indexing Process, Text Processing, Automatic Indexing,** Information Extraction

Data Structure: Introduction to Data Structure, Stemming Algorithms, Inverted File Structure, N-Gram Data Structures, PAT Data Structure, Signature File Structure, Hypertext and XML Data Structures, Hidden Markov Models.

Cataloging and Indexing: History and Objectives of Indexing

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 concept-based 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 of indexing:

- *Total document indexing*
  - The full text searchable data structure for items in the Document File provides a new class of indexing called total document indexing.
  - Current systems have the ability to automatically weight the processing tokens based upon their potential importance in defining the concepts in the item.
  - Previously, indexing defined the source and major concepts of an item and provided a mechanism for standardization of index terms (i.e., use of a controlled vocabulary).
- *Controlled vocabulary*
  - In a manual indexing environment, the use of a controlled vocabulary makes the indexing process slower, but potentially simplifies the search process.

- Controlled vocabularies aid the user in knowing the domain of terms that the indexer had to select from.

- *Uncontrolled vocabularies*
  - Make indexing faster but the search process much more difficult.
  - The availability of items in electronic form changes the objectives of manual indexing.
  - Modern systems, with the automatic use of thesauri and other reference databases, can account for diversity of language/vocabulary use and thus reduce the need for controlled vocabularies.
- *Automatic text analysis algorithms*
  - The words used in an item do not always reflect the value of the concepts being presented.
  - It is the combination of the words and their semantic implications that contain the value of the concepts being discussed.
  - The utility of a concept is also determined by the user's need.
- *Public File indexer*
  - Public File indexer needs to consider the information needs of all users of the library system.
  - Individual users of the system have their own domains of interest that bound the concepts in which they are interested.
  - It takes a human being to evaluate the quality of the concepts being discussed in an item to determine if that concept should be indexed.
- *Private Index files*
  - Allows the user to logically subset the total document file into folders of interest including only those documents that, in the user's judgment, have future value.
  - Allows the user to judge the utility of the concepts based upon his need versus the system need and perform concept abstraction.
- *Selective indexing*
  - Is based upon the value of concepts increases the precision of searches.
- *Full document indexing*
  - Availability of full document indexing saves the indexer from entering index terms that are identical to words in the document.

Indexing Process

- When an organization with multiple indexers decides to create a public or private index, some procedural decisions assist the indexers and end users on how to create the index terms.
  - Scope of the indexing
  - Linking index terms
- *Scope of the indexing*
  - Defines the level of detail that the subject index will contain.
  - When performed manually, the process of determining the bibliographic terms that represent the concepts in an item is difficult.
  - Problems arise from interaction of two sources:
    - The author and
    - The indexer.

- The vocabulary domain of the author may be different than that of the indexer, causing the indexer to misinterpret the importance.
- The indexer is not an expert on all areas and has different levels of knowledge in the different areas being presented in the item.
- The indexer must determine when to stop the indexing process.

- There are two factors involved in deciding on what level to index the concepts in an item:
  - Exhaustivity
  - Specificity.
    - Exhaustivity of indexing
      - Is the extent to which the different concepts in the item are indexed.
      - For example, if two sentences of a 10-page item on microprocessors discusses on-board caches, should this concept be indexed?
    - Specificity
      - Relates to the preciseness of the index terms used in indexing.
      - For example, whether the term "processor" or "microcomputer" or "Pentium" should be used in the index of an item.
    - Using general index terms yields low exhaustivity and specificity.

- *Pre coordination and Linkages*
  - Another decision on the indexing process is whether linkages are available between index terms for an item.
  - Linkages are used to correlate related attributes associated with concepts discussed in an item.
    - Precoordination
      - Process of creating term linkages at index creation time.
      - When index terms are not coordinated at index time, the coordination occurs at search time.
    - Post coordination
      - Coordinating terms after (post) the indexing process.
      - Implemented by "and" ing index terms together, which only finds indexes that have all of the search terms.

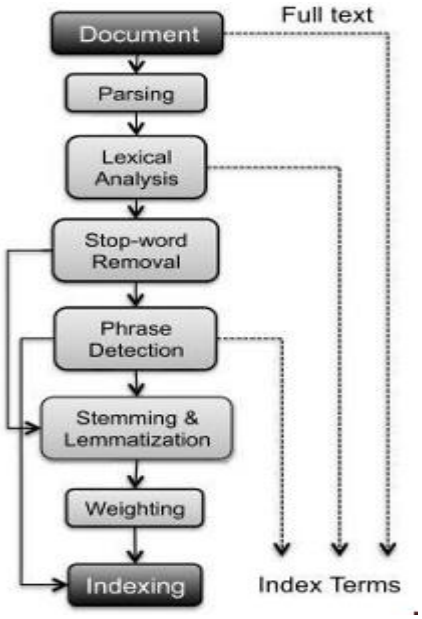
INDEX TERMS	Methodology
oil, wells, Mexico, CITGO, refineries, Peru, BP, drilling	No linking of terms
(oil wells, Mexico, drilling, CITGO)	linked (Precoordination)
(U.S.,oil refineries, Peru, introduction)	
(CITGO, drill, oil wells, Mexico) (U.S., introduction, oil refineries, Peru)	linked (Precoordination) 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 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.
- Introducing roles in the last two examples of Figure removes this ambiguity.
- 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.

Text Processing

- *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 preprocessing phase, unit documents are created; e.g., emails with attachments are split into one document representing the email and as many documents as there are attachments.

- *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.



- *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 word 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”.
- *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 may be 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”.
- *Stemming and Lemmatization*
  - Following phrase extraction, stemming and 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.
  - According to the Porter stemmer, our example sentence “Search engines are the most visible information retrieval applications” would result in: “Search engin are the most visibl 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
   - *Weighting*
     - The final phase of text preprocessing deals with term weighting. 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.

Automatic Indexing

- Capability for the system to automatically determine the index terms to be assigned to an item.
- More complex processing is required when emulate a human indexer and determine a limited number of index terms.
- Consistency in the index term selection process as indexing is performed automatically by an algorithm.
- Indexes from automated indexing fall into two classes:
  - Unweighted.
  - Weighted

Unweighted indexing system

- Index term in a document and its word location(s) are kept in the searchable data structure.
- 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 the first item to be relevant to the user's information need.

Weighted indexing system

- Weight of the index term is based upon a function associated with the frequency of occurrence of the term in the item.
- Values for the index terms are normalized between 0 and 1.
- The higher the weight, the more the term represents a concept discussed in the item.
- Query process uses the weights along with any weights assigned to terms in the query to determine a rank value.

Automatic Indexing

1. Indexing by Term
2. Indexing by Concept
3. Multimedia Indexing

Indexing by Term

- There are two major techniques for creation of the index terms:
  - Statistical techniques
  - Natural language techniques

Statistical techniques

  - Based upon **vector models and probabilistic models** with a special case being Bayesian models.
  - Calculation of weights in those models use statistical information such as the frequency of occurrence of words and their distributions in the searchable database.
- Vector model (Example. SMART system)
  - The system emphasizes weights as a foundation for information detection and stores these weights in a vector form.
  - Each vector represents a document.
  - 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.
- Probabilistic model.
  - Bayesian approach is the most successful model in this area.
  - It is based upon the theories of evidential reasoning (drawing conclusions from evidence).
  - The Bayesian approach could be applied as part of index term weighting by calculating the relationship between an item and a specific query.
  - A Bayesian network is a directed acyclic graph
  - Each node represents a random variable.
  - Arcs between the nodes represent a probabilistic dependence between the node and its parents.

- Figure shows the basic weighting approach for index terms or associations between query terms and index terms.
- The nodes C_1 and C_2 represent "the item contains concept C_i"
- F nodes represent "the item has feature F_{ij} (e.g., words)
- The network 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}.

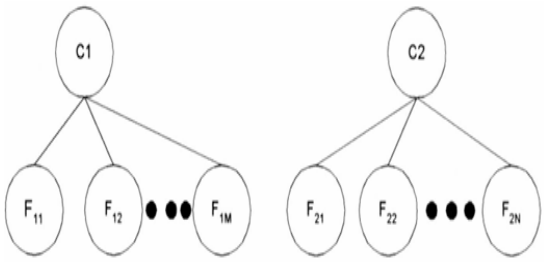


Fig Two-level Bayesian network

To perform that calculation two sets of probabilities are needed:

- The prior probability P(Ci) that an item is relevant to concept C.
- The conditional probability P(Fij/Ci) that the features Fij where j=1, m is present in an item given that the item contains topic Ci.

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

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

- 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:

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

- 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.
- Function g is the sum of the weights of the features
- w is weight corresponding to each feature (index term)
- w produces a ranking in decreasing order that is equivalent to the order produced by the posterior probabilities.

Natural language techniques

  - Perform more complex parsing to define the final set of index concepts.
  - Weighted systems are discussed as vectorized information systems.
- DR-LINK (Document Retrieval through Linguistic Knowledge) system
  - Define indexes to items via natural language processing.
  - 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.
- This allows the indexing to include specific term as well as abstract concepts such as time.

Indexing by Concept

- Indexing by term treats each occurrence as a different index.
- Then uses thesauri or other query expansion techniques to expand a query to find the different ways the same thing has been represented.
- Basis for concept indexing
  - There are many ways to express the same idea and increased retrieval performance comes from using a single representation.
  - Concept indexing determines a related set of concepts based upon a test set of terms
  - Uses those concepts for indexing all items
  - Latent Semantic Indexing
    - Indexing the latent semantic information in items.
- Example of concept indexing
  - Match Plus system
    - Neural networks facilitate machine learning of concept/word relationships.
    - Goal is to determine word relationships (e.g., synonyms) and the strength of these relationships and use that information in generating context vectors, from the corpus of items.
    - Two neural networks are used. One neural network learning algorithm generates stem context vectors that are sensitive to similarity of use. Another one performs query modification based upon user feedback.
    - Context vectors: Word stems, items and queries that are represented by high dimensional (at least 300 dimensions) vectors. Each dimension in a vector could be viewed as an abstract concept class.

Multimedia 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 (e.g., Virage) 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.

Analog audio input

- System will convert the audio to digital format and determine the phonemes associated with the utterances.
- The phonemes will be used as input to a Hidden Markov Search model that will determine with a confidence level the words that were spoken. A single phoneme can be divided into four states for the Markov model. It is the textual words associated with the audio that becomes the searchable structure. In addition to storing the extracted index searchable data, a multimedia item also needs to store some mechanism to correlate the different modalities during search.
- There are two main mechanisms that are used
    - Positional
        - Is used when the modalities are scattered 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 just a position of each modality
    - Temporal Positional
        - 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.

Synchronized Multimedia Integration Language (SMIL)

- 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.
- Time is the mechanism that is used to synchronize the different modalities.
- Indexing must include a time-offset parameter versus a physical displacement.

Advantages of human indexing

- Ability to determine concept abstraction
- Ability to judge the value of a concept.

Disadvantages of human indexing

- Cost
- Processing time
- Consistency

Information Extraction

- *There are two processes associated with information extraction*:
  - Determination of facts to go into structured fields in a database
    - Only a subset of the important facts in an item may be identified and extracted.
  - Extraction of text that can be used to summarize an item.
    - Summarization: all the major concepts in the item should be represented.
- *Extraction*
  - 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.
  - 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).
  - Metrics to compare information extraction.
    - 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. It refers to how much information was extracted accurately versus the total information extracted.
    - Over generation: It 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 Slots that filled with non-relevant data.
    - Fallout — It measures how much a system assigns incorrect slot fillers.
- *Document summarization*
  - Goal is to extract a summary of an item maintaining the most important ideas while significantly reducing the size.
  - Examples of summaries are part of any item such as titles, table of contents, and abstracts.

- The abstract can be used to represent the item for search purposes or to determine the item without having to read the complete item.
- Most automated algorithms summarize by calculating a score for each sentence and then extracting the sentences with the highest scores.

Data Structure: Introduction to Data Structure

- From an Information Retrieval System perspective, the two aspects of a data structure are
  - its ability to represent concepts and their relationships
  - how well it supports location of those concepts
- There are two major data structures in any information system.
  - One structure stores and manages the received items in their normalized form; this process is called document manager.
- The other contains the processing tokens and associated data to support search.

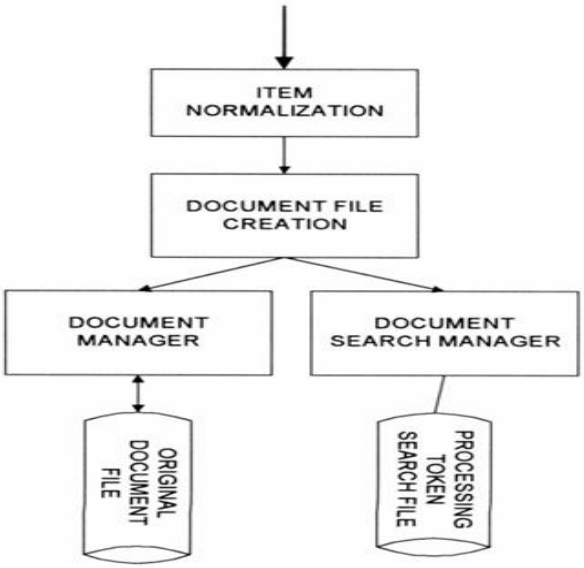


Fig Major Data Structures

Stemming Algorithms

- Goal of stemming was to improve performance and reduce system resources by reducing the number of unique words that a system has to contain.
- The stemming process creates one large index for the stem.
  1. Introduction to the Stemming Process
  2. Porter Stemming Algorithm
  3. Dictionary Look-Up Stemmers
  4. Successor Stemmers
     a) Cutoff method
     b) Peak and Plateau
     c) Complete word method
     d) Entropy method

1. *Introduction to the Stemming Process*
   - Stemming algorithms are used to improve the efficiency of the information system and to improve recall.
   - Conflation: the term used to refer to mapping multiple morphological variants to a single representation (stem). Thus, exceptions and non-consistent variants are always

present in languages that typically require 'exception look-up tables' in addition to the normal reduction rules.

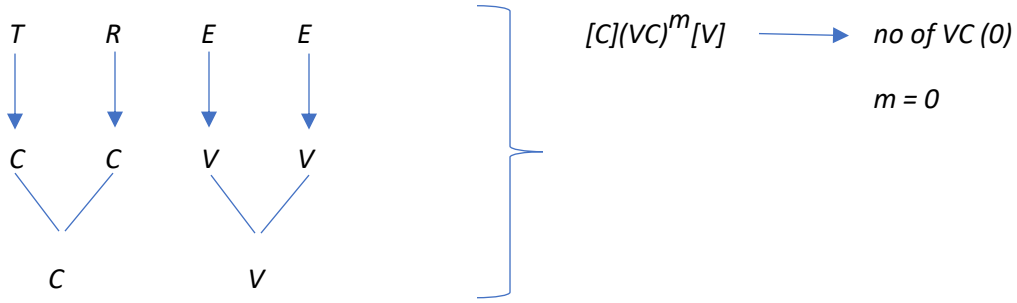
- For example, the stem "comput" could associate "computable, computability, computation, computational, computed, computing, computer, computerese, computerize" to one compressed word.
- Compression of stemming does not significantly reduce storage requirements. Misspellings and proper names reduce the compression even more.
- Another major use of stemming is to improve recall. As long as a semantically consistent, stem can be identified for a set of words.
- Stemming of the words "calculate, calculates, calculation, calculations, calculating" to a single stem "calculat" assures whichever of those terms is entered by the user, it is translated to the stem and finds all the variants in any items they exist.
- Other techniques such as table lookup and successor stemming provide alternatives that require additional overheads. Successor stemmers determine prefix overlap as the length of a stem is increased.

2. *Porter Stemming Algorithm*
   - The Porter Algorithm is based upon a set of conditions of the stem, suffix and prefix and associated actions given the condition. Some examples of stem conditions are:
     1. The measure, m, of a stem is a function of sequences of vowels (a, e, i, o, u, y) followed by a consonant. If V is a sequence of vowels and C is a sequence of consonants, then m is

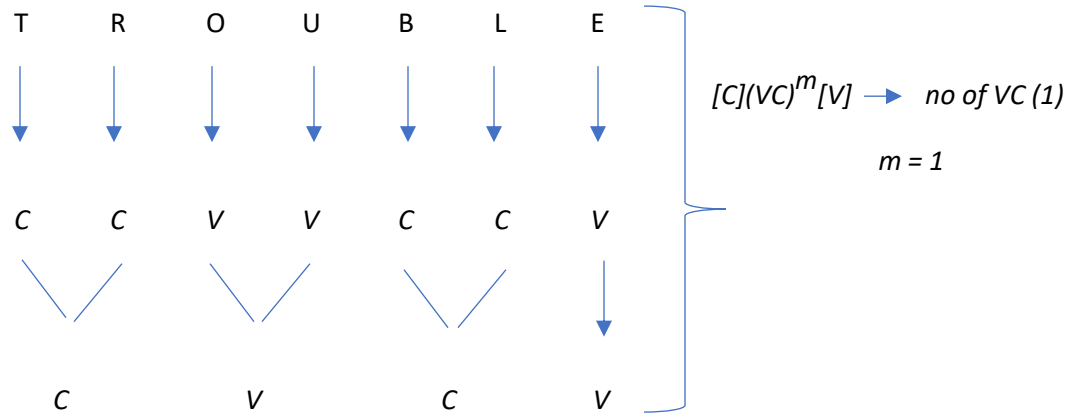
        $$C(VC)^m V$$

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

Example 1



Example 2



Example 3

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[1]	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 word "duplicatable," the following are the steps in the stemming process:

duplicat	rule 4
duplicate	rule 1b1
duplic	rule 3

3. Dictionary Look-Up Stemmers
    - In this approach, simple stemming rules may be applied.
    - The rules are taken from those that have the fewest exceptions (e.g., removing pluralization from nouns).
    - Even the most consistent rules have exceptions.
    - 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 INQUERY system uses a stemming technique called Kstem.
    - Kstem is a morphological analyzer that that reduces morphological variants to a root form.
    - For example,
        - `elephants'->`elephant', `amplification'->`amplify', and `european' >`europe'.
            - 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”).

4. Successor Stemmers

The process determines the successor varieties for a word and 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 is the number of distinct letters that occupy the segment length plus one character.

- A graphical representation of successor variety is shown in a symbol tree.
- Figure 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

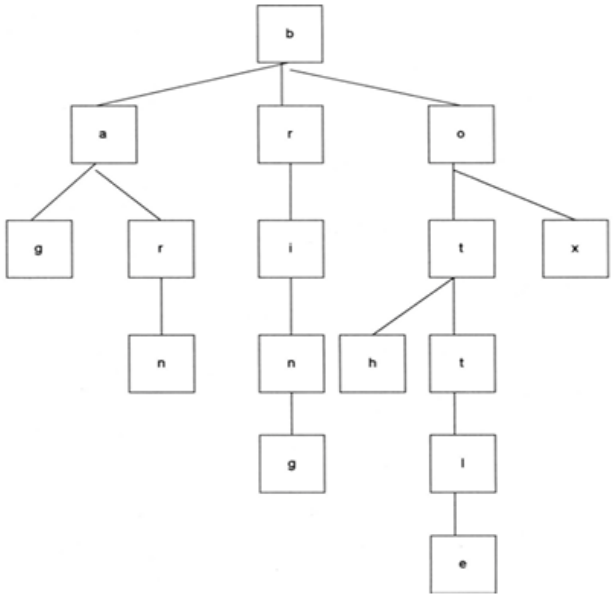


Figure 4.2 Symbol Tree for terms bag, barn, bring, box, bottle , both

- 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 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 entropy (Average Information) 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.

To illustrate the use of successor variety stemming, consider the example below where the task is to determine the stem of the word READABLE.

```
        Test Word:  READABLE
        Corpus:     ABLE, APE, BEATABLE, FIXABLE, READ,
                    READING, READS, RED, ROPE, RIPE.
  Prefix     Successor Variety      Letters
----------------------------------------
R                   3               E,I,O
RE                  2               A,D
REA                 1               D
READ                3               A,I,S
READA               1               B
READAB              1               L
READABL             1               E
READABLE            1               BLANK
```

Inverted File Structure

- Inverted file structure used in both database management and Information Retrieval Systems.
- Inverted file structures are composed of three basic files:
    - *Document file*
    - *Inversion lists (posting files)*
        - For each word, a list of documents in which the word is found in is stored (the inversion list for that word).
        - Each document in the system is given a unique numerical identifier that is stored in the inversion list.
    - *Dictionary*
        - Dictionary is used to locate the inversion list for a particular word.
        - Dictionary is a sorted list of all unique words (processing tokens) in the system and a pointer to the location of its inversion list.

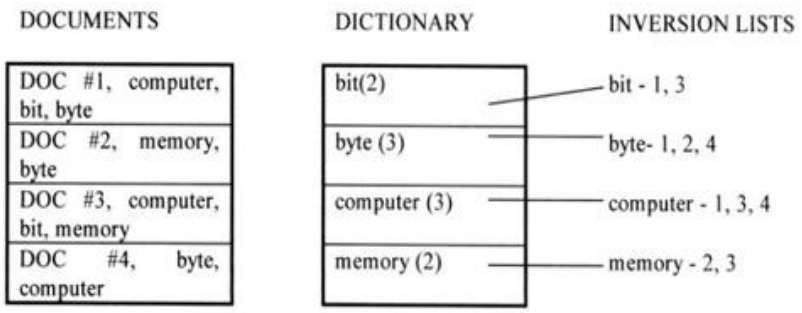


Fig Inverted File Structure

- Additional information may be used from the item to increase precision and provide a more optimum inversion list file structure.
- For example, if zoning issued, the dictionary may be partitioned by zone.
- There could be a dictionary and set of inversion lists for the “Abstract” zone in an item and another dictionary and set of inversion lists for the “Main Body” zone.
- The inversion list contains the document identifier for each document in which the word is found.
- To support proximity, contiguous word phrases, and term weighting algorithms, all occurrences of a word are stored in the inversion list along with the word position.
- Thus, if the word “bit” was the tenth, twelfth and eighteenth 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.
- For systems that support ranking, the list is reorganized into ranked order.
- The document numbers are used to retrieve the documents from the Document File.

```
Documents Collection:
-------------------------------------
Doc 1: "AI is the future of technology"
Doc 2: "AI and ML are changing technology"
Doc 3: "AI is used in various fields of technology"

Step 1: Build Dictionary (Lexicon)
-------------------------------------
Unique Terms: { AI, is, the, future, of, technology, and, ML, are, changing, used, in, various, fields }

Step 2: Create Posting List
-------------------------------------
Word          | Posting List (Document IDs)
-------------------------------------
AI            | { Doc1, Doc2, Doc3 }
is            | { Doc1, Doc3 }
the           | { Doc1 }
future        | { Doc1 }
of            | { Doc1, Doc3 }
technology    | { Doc1, Doc2, Doc3 }
and           | { Doc2 }
ML            | { Doc2 }
are           | { Doc2 }
changing      | { Doc2 }
used          | { Doc3 }
in            | { Doc3 }
various       | { Doc3 }
fields        | { Doc3 }
```

- Instead of using a dictionary to point to the inversion list, B-trees can be used.
- The inversion lists at the leaf level or referenced in higher level pointers.
- "Figure B-Tree Inversion Lists" shows how the words in "Figure Inverted File Structure "would appear.
- 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.

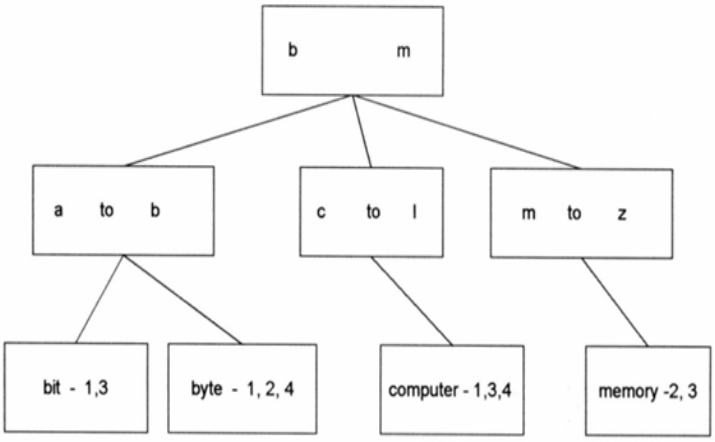


Fig B-Tree Inversion Lists

N-Gram Data Structures

- A special technique for conflation (stemming).
- A unique data structure in information systems that ignores words and treats the input as a continuous data, optionally limiting its processing by interword symbols.
- A fixed length consecutive series of "n" characters or fixed length overlapping symbol segments that define the searchable processing tokens.
- These tokens have logical linkages to all the items in which the tokens are found.
- To store the linkage data structure, inversion lists, document vectors and other data structures are used in the search process.
- N-grams do not care about semantics, unlike stemming (that determine the stem of a word that represents the semantic meaning of the word).

- Examples of bigrams, trigrams and pentagrams are given in Figure for the word phrase "sea colony".
- n-grams with n greater than two allow interword symbols to be part of the n-gram set.
- The symbol # is used to represent the interword symbol (e.g., blank, period, semicolon, colon, etc.). Each of the n-grams created becomes a separate processing token 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 #)

Fig Bigrams, Trigrams and Pentagrams for "sea colony"

- *The advantage of n-grams*
  - Limits the number of searchable tokens.
  - Maximum number of unique n-grams generated.
  - Fast processing on minimally sized machines
- *Disadvantage*
  - False hits can occur under some architectures.
  - Increased size of inversion lists that store the linkage data structure.
  - N-grams expands the number of processing tokens.
  - There is no semantic meaning in a particular n-gram since it is a fragment of processing token and may not represent a concept.
  - Poor representation of concepts and their relationships.

PAT Data Structure

- A Continuous text input data structure is indexed in contiguous "n" character tokens using n-grams with interword symbols between processing tokens.
- A continuous text input data structure is addressed differently using PAT trees and PAT arrays.
- The name PAT is short for PAtriciaTrees (PATRICIA stands for Practical Algorithm To Retrieve Information Coded In Alphanumerics.)

Sistring (Semi-infinite string)

- The input stream is transformed into a searchable data structure consisting of substrings called sistring or semi-finite 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 includes all new text up to the end of the input.

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.

Fig Examples of 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

Fig Sistrings for input "100110001101"

- All substrings are unique. A substring can start at any point in the text and can be uniquely indexed by its starting location and length.
- Substring may go beyond the length of the input stream by adding additional null characters.

PAT tree

- It 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

Above 1st shows some possible sistrings for an input text and 2nd Figure gives an example of the sistrings used in generating a PAT Tree.

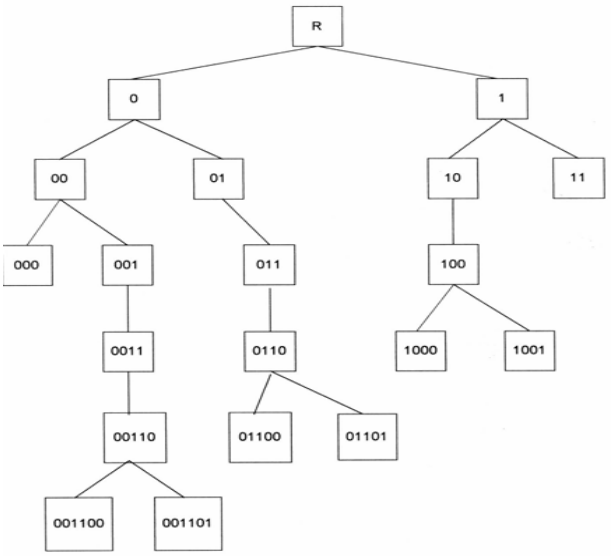


Figure 4.11 PAT Binary Tree for input "100110001101"

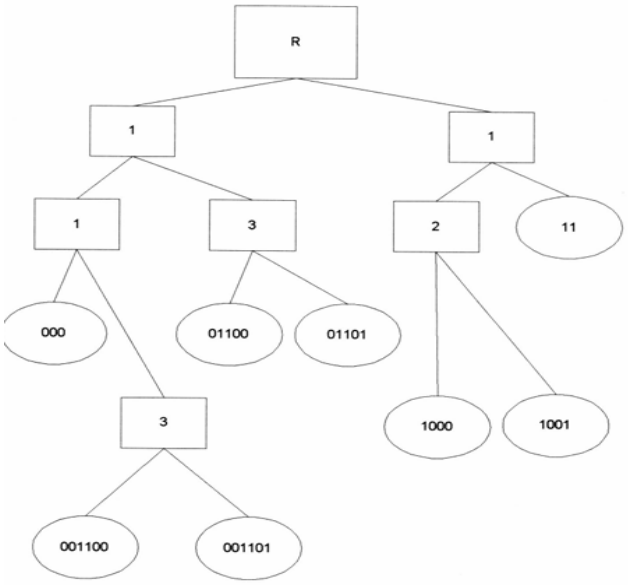


Figure 4.12 PAT Tree skipping bits for "100110001101"

- If the binary representations of "h" is (100), "o" is (110), "m" is (001) and"e" is (101) then the word "home" produces the input 100110001101.
- Using the sistrings, the full PAT binary tree is shown in Figure 4.11.
- A more compact tree where skip values are in the intermediate nodes is shown in Figure 4.12

Advantages and Disadvantages

- PAT trees are ideal for prefix searches.
- Suffix, imbedded string, and fixed length masked searches are easy if the total input stream is used in defining the PAT tree.
- Fuzzy searches are very difficult because large number of possible sub-trees could match the search term.
- PAT arrays have more accuracy than Signature files
- Ability to string searches that are inefficient in inverted files (e.g., suffix searches, approximate string searches, longest repetition).
- It is not used in any major commercial products.

Signature File Structure

The goal of a signature file structure

- To provide a fast test to eliminate the majority of items that are not related to a query.
- The items that satisfy the test can either be evaluated by another search algorithm to eliminate additional false hits or delivered to the user to review.
- The text of the items is represented in a highly compressed form that facilitates the fast test.

Signature file search

- Signature search file is created using superimposed coding.

Superimposed coding

- The coding is based upon words in the item.
- The words are mapped into a “word signature”.

Word signature

- A fixed length code with a fixed number of bits.
- A bit pattern of size f, with m bits set to "1", while the rest are "0".
- The bit positions are determined using a hash function of the word.
- The word signatures are “or “ ed together to create the signature of an item.
- 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.
  - In Figure on right the block size is set at five words
  - Code length is 16 bits
  - Number of bits that are allowed to be “1” for each word is five.
  - The words in a query are mapped to their signature.

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

Fig Superimposed Coding

Signature file

- Can be stored as a signature with each row representing signature block.
- Associated with each row is a pointer to the original text block.

Design objective of a signature file system

- Trading off the size of the structure versus the density of the final created signatures.
- Longer code lengths reduce the probability of collision in hashing the words (i.e., two different words hashing to the same value).
- Fewer bits per code reduce the effect of a code word pattern being in the final block signature.

Application / Advantages

- Signature files provide a solution for storing and locating information in a number of different situations.
- Signature files are applied as medium size databases, WORM devices, parallel processing machines, and distributed environments.

Hypertext and XML Data Structures

Hypertext:

- A mechanism for representing information structure.
- Hypertext is a non-sequential directed graph structure, where each node contains its own information.
- 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.
- Hypertext references are used to include information that is other than text (e.g., graphics, audio, photograph, video) in a text item.
- Hypertext is stored in Hypertext Markup Language (HTML) and extensible Markup Language (XML).
- HTML and XML provide detailed descriptions for subsets of text similar to the zoning that increase search accuracy and improve display of hit results.

Hypertext Structure

- Used in the Internet environment
- Requires 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.
- Reference pointer is called a link.
- Each node is displayed by a viewer that is defined for the file type associated with the node.

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>)
- 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 (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:

- Access method the client used to retrieve the item
- Internet address of the server where the item is stored
- Address of the item at the server

Dynamic HTML

- Combination of the latest HTML tags, style sheets and programming that help to create WEB pages that are more animated and responsive to user interaction.
- Supports features such as object-oriented view of a WEB page and its elements, cascading style sheets, programming that can address most page elements add dynamic fonts.
- Allows the specification of style sheets in a cascading fashion.

```
<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">
```

Fig Example of Segment of HTML

XML

- The extensible Markup Language (XML) is a standard data structure on the WEB.
- Objective is to extend HTML with semantic information.
- The logical data structure within XML is defined by a Data Type Description (DTD)
- The user can create any tags needed to describe and manipulate their structure.

The following is a simple example of XML tagging:

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

Resource Description Format (RDF)

- Used to represent 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 content filtering (e.g., unsuitable for children).

Hypertext links for XML

- These are defined in the Xlink (XML Linking Language) and Xpoint (XML Pointer language) specifications.
- Allow different types of links to locations within a document and external to the document.

- Allow an application to know if a link is positioning reference within an item or link to another document.
- Help in determining what needs to be retrieved to define the total item.

XML Style Sheet Linking

- Define how to display items on a particular style sheet and handle cascading stylesheets.
- Allow designers to limit what is displayed to the user and allow expansion to the whole item if desired.

Hidden Markov Models

Markov process assumption

- Future is independent of the past given the present.
- In other words, assuming we know our present state, we do not need any other historical information to predict the future state.

Example of a three state Markov Model of the Stock Market

State 1 (S1): market decreased
State 2 (S2): market did not change
State 3 (S3): market increased in value

The states will be one of the above that is observed at the closing of the market.

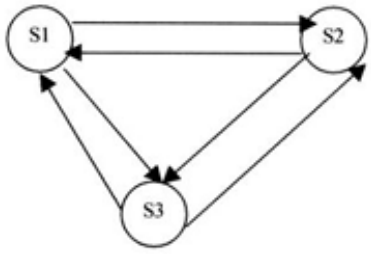
The movement between states can be defined by a state transition matrix with state transitions.

$$A = \{a_{ij}\} = \begin{matrix} .5 & .3 & .4 \\ .1 & .6 & .3 \\ .6 & .7 & .5 \end{matrix}$$

- Given that the market fell on one day (State 1), the matrix suggests that the probability of the market not changing the next day is 1.
- This then allows questions such as the probability that the market will increase for the next 4 days then fall.
- This would be equivalent to the sequence of SEQ = {S3, S3, S3, S3, S1}.
- Let's assume that instead of the current state being dependent upon all the previous states, let's assume it is only dependent upon the last state.

$$\begin{aligned} P(SEQ) &= P[S3, S3, S3, S3, S1] \\ &= P[S3] * P[S3/S3] * P[S3/S3\} * P[S3/S3] * P[S1/S3] \\ &= S3(init) * a_{3,3} * a_{3,3} * a_{3,3} * a_{1,3} \\ &= (1.0) * (.5) * (.5) * (.5) * (.4) \\ &= .05 \end{aligned}$$

- The following graph depicts the model.
- The directed lines indicate the state transition probabilities $a_{i,j}$.
- In the example, every state corresponded to an observable event (change in the market).
- To add more flexibility, probability function was allowed to be associated with the state. The result is called the Hidden Markov Model.



Hidden Markov Models (HMMs)

- A class of probabilistic graphical model that allow us to predict a sequence of unknown (hidden) variables from a set of observed variables.
- Allow us to compute the joint probability of a set of hidden states (latent states) given a set of observed states.
- Once we know the joint probability of a sequence of hidden states, we determine the best possible sequence of hidden states.

Definition of a discrete Hidden Markov Model is summarized as consisting of the following:

1. $S = \{ S_0, , \ldots, S_{n-1}\}$ 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. $V = \{ v_0, , \ldots, v_{m-1}\}$ is a finite set of output symbols. This will correspond to the physical output from the system being modeled.
3. $A = S \times S$ a transition probability matrix where $a_{i,j}$ represents the probability of transitioning from state i to state j such that $\sum_{j=0}^{n-1} a_{i,j} = 1$ for all $i = 0, \ldots, n-1$. Every value in the matrix is a positive 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 output probability matrix where element $b_{j,k}$ is a function 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 output symbol at every state transition.
- The transition probability is the probability of the next state given the current state.
- The output probability is the probability that a given output is generated upon arriving at the next state.

The complete specification of a HMM requires

- Specification of the states
- The output symbols
- Three probability measures for the state transitions
- Output probability functions and the initial states

The distributions are frequently called A, B, and ø and the following notation is used to define the model:

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

Issues with HMM are

- How to efficiently calculate the probability of a sequence of observed outputs given the HMM model.
- How to determine which of a number of competing models should be selected given an observed set of outputs.
- How best to tune the λ model to maximize the probability of the output sequence given λ.

Applications of Hidden Markov Models (HMM)

- Speech recognition
- Optical character recognition
- Topic identification
- Information retrieval search.