**UNIT-III**

**Syllabus**: **Automatic Indexing:** Classes of automatic indexing, Statistical indexing, Natural  
language, Concept indexing, Hypertext linkages**.**

Document and Term Clustering: Introduction, Thesaurus generation, Item clustering, Hierarchy of clusters.

**Objective:**

This unit is focuses on the process and algorithms to perform indexing. The Indexing Process is transformed of an item that extracts the semantics of the topics discussed in the item. The extracted information is used to crate the processing tokens and the searchable data structure. This Unit will focus on Automatic Indexing and User Search Techniques.

* 1. **What is Statistical Indexing? Explain**

Statistical Indexing uses frequency of occurrence of events to calculate a number that is used to indicate the potential relevance of an item.

Methods Used for Statistical Indexing is

* 1. **Probabilistic Weighting**
  2. **Vector Weighting**

**1. Probabilistic Weighting**

This is based upon the direct application of the theory of probability to IRS. It also leads to invariant results that facilitates from different databases. The use of probability theory is a natural choice because it is the basic evidential reasoning. This is summarized by Probability Ranking Principles (PRP) and its Plausible Corollary.

The Logistic Reference Model uses a random sample of query-document-term triples for which binary relevance judgments have been made from training sample. Log O is the logarithm of the odds of relevance for term tk. This is present in document Dj and Query Qi:

Log (O(R/Qi, Dj, tk) =c0+c1v1+……………..+cnvn

* 1. **Principles of the Vector Model**

**Principles of the Vector Model**

* The SMART system by Salton et al at Cornell University.
* **Vector** : a *sequence* of values(*v1, v2, …,vk*).
* Let *T1, T2, …, Tn* be the terms(tokens) in the entire vocabulary of the collection.
* Let *D1, D2, …, Dm* be the documents in the collection.
* Each item *Dj* is represented by a vector(*wj1, wj2, …, wjn*) where *wji* is a number that corresponds to the term *Ti* in document *Dj*
  + **Binary approach** : *wji* is either 0 or 1, indicating the *presence* or *absence* of the term in the document.
  + **Weighted approach** : *wji* is a positive number, indicating the *relative importance* of the term in representing the document.
* Each document becomes a vector(point) in *n*-dimensional space.

Example

* Let the vocabulary be (*n*=6) :

Petroleum, Mexico, Oil, Taxes, Refineries, Shipping.

* A document might be represented in a binary system :

(1, 1, 1, 0, 1, 0)

* And in a weighted system :

(2.8, 1.6, 3.5, 0.3, 3.1, 0.1)

* Binary systems require the use of a *threshold* to determine whether the *degree* to which a term represents the document is sufficient to merit the value 1.
* Restricting to the first three dimensions only :

**Mexico 1.6**

**Petroleum 2.8**

**2.8**

**Oil**

**3.5**

**Calculating Term Frequency in the Vector Model**

* A *statistical* approach
* Three statistics are usually available for each term :
  + **Term Frequency** *TFij* : the frequency of occurrence of a term *Ti* in a document *Dj*.
  + **Total Term Frequency** *TTFi* : the frequency of occurrence of a term *Ti* in the entire collection.
  + **Document Frequency** *DFi* : the number of unique documents in the collection that contain a term *Ti*.

1. A Simple Term Frequency Algorithm

* Algorithm:
  + Determine the set of terms.
  + For each term *Ti* and each document *Dj*, calcuate weight simply as term frequency. I.e., *TFij*, the number of occurrences of a term *Ti* in an document *Dj* becomes the weight *wji* in the vector describing *Dj*
  + If using a binary approach, choose threshold *T*, and assign to document *Dj* all terms *Ti* for which *TFij* > *T*.
* **Problem**: biases towards longer documents. The longer the document, the more often a term may occur.
* **Solution**: normalize for (e.g., divide by) the number of words in the document.

2. Inverse Document Frequency

* **Problem**: The previous weighting algorithm does not distinguish sufficiently between a term that occurs in most of the documents in the collection and a term that occurs in just a few documents.
  + A term that occurs in most documents has less “resolving” power. It results in retrieval of documents that are not useful.
* **Solution**: weight should also be inversely proportional to the document frequency.
* **Formula**: *wji* = *TFij* \*[log2(*m*) - log2(*DFi*) + *j*]
  + *m* = the number of documents in the collection.
  + *wji* = the weight of term *Ti* in document *Dj*
  + *TFij* = the frequency of term *Ti* in document *Dj*
  + *DFi* = the number of documents in which term *Ti* occurs.
  + The weight is proportional to the term frequency *TFij*.
  + The weight is proportional to the term specificity factor[…] (inverse proportional to the document frequency *DFi*).
  + log2 is a “moderating” function.

Example

* Total Number of documents: m=2048
* Document frequency of terms:
  + DF(oil)= 128
  + DF(Mexico**) = 16**
  + DF(Refinery) = 1024
* New document has these three terms with term frequencies:
  + TF(oil) = 4
  + TF(Mexico) = 8
  + TF(Refiney) = 10
* Weights vector by simple(unnormalized) term frequency:(4,8,10)
* Weights vector by inverse document frequency:(20,64,20)
  + W(oil) = 4 \* (log2(2048) - log2(128) + 1 = 4\*(11-7+1) = 20
  + W(Mexico) = 8\*(log2(2048) - log2(16) + 1) = 8 \*(11-4+1) = 64
  + W(Refinery) = 10\*(log2(2048) - log2(1024) + 1) = 10 \* (11-10+1) =20

3. Signal (Information Theoretical Approach)

* Weighting by inverse item frequency considers the *number of documents* in the

Collection that contain the term.

* It does not account for the distribution of the term *frequency* in the documents

that contain the term.

* **Example** :Assume five documents contain the terms *Saw* and *Drill*:



* The uniform distribution of the term Saw does not give any clue as to which

document is more likely to be relevant to a search for Saw.

* The weighting algorithm should take into account the non-uniform distribution

of the term Drill(would help in ranking!)

* **Information Theory**(Shannon):the information content value of an

event is inversely proportional to the probability that it occurs.

* Let e be an event, and let p(e) be the probability that it occurs. Then

Information (e) = -log2(p(e))

* Examples:
  + The information contents of an event that occurs with probability

0.0005 is -log2(0.0005) = -(-10) = 10

* + The information contents of an event that occurs with probability

0.5 is -log2(0.5) = -(-1) = 1

* + The information contents of an event that occurs with probability

1.0 is -log2(1.0) = -(0) = 0(An event fully anticipated)

* Average information content: If event e has n possible outcomes

with probabilities p1…p n then the average information value is

* Average\_Information(e) =



* This value is maximized when all the pi are identical.
* Define the probability of a term ***Ti*** occurring in a document ***Dj.***

***Pi = TFij / TTFi***

Its occurrences in ***Dj*** divided by total number of occurrences.

* Average information of Saw;

- [ 10/50 log2 (10/50) + 10/50 log2 (10/50) + … + 10/50 log2 (10/50)

**= -** [ 5 \* 0.2 log2 (0.2) ] **=** - [ -2.32] **=** **2.32**

* Average information of Drill:

- [ 2/50 log2 (2/50) + 2/50 log2 (2/50) + 18/50 log2 (18/50)

+ 10/50 log2 (10/50) + 18/50 log2 (18/50)] **=**

- [ 2 \* 0.04 log2 (0.04) + 2 \* 0.36 log2 (0.36) + 0.2 log2 (0.2)] **=**

- [ 0.08(-4.64) + 0.72(-1.47) + 0.2 (-2.32)] = **1.89**

* To use Average information as a weight we define:

***Signali = log***2 ***(TTFi) - average\_informationi***

* Signal of Saw: log2 (50) - 2.32 = 5.64 - 2.32 = **3.32**
* Signal of Drill: log2 (50) - 1.89 = 5.64 - 1.89 = **3.75** (higher!)
* Signal is combined with discrimination value: ***Wji = TFIJ \* Signali***

4. Term Discrimination Value

* Alternative to the term specificity factor in the second method that supports terms that have high specificity.
* Instead, we compute a factor that denotes the ability of the term to discriminate among documents.
* In both methods, the weight is still proportional to term frequency.
* We place all documents(vectors) in the space and consider the “spread” among documents(distance is inverse of similarity)
* When a new term is assigned to documents:

1. The distances among the documents to which it is assigned decrease.

2. The distances among the documents to which it is not assigned decrease as well.

3. The distances between these documents and the documents in the rest of the collection increase.

* Overall, does the addition of a new term increase of decrease the distances among documents?
* Denote Q the density of the document space ( however measured).
* Denote *Qi* the density of the space after term ***Ti*** is introduced and assigned to the appropriates.
* Define the discrimination value of term ***Ti***

***Dvi = Q - Qi***

* + if ***Dvi*** is positive, then ***Ti*** is a good discriminator

(the density has decreased after its introduction).

* + if ***Dvi*** is negative, then ***Ti*** is a poor discriminator

(the density has increased after its introduction).

* + if ***Dvi*** is close to zero, then ***Ti*** is a neutral discriminator

(its introduction does not affect the density).

* Term frequency is combined with discrimination value:

***Wji = Tfij \* DVi***

* One way to define the density Q of a space of m documents:



* Average of all pair-wise similarities.
* When documents are similar, Q is high.
* if Ti makes documents less similar , then Qi will be lower,

and Q-Qi will be positive.

* Another way to define the density Q:
* Define a centroid document C=(C1,C2,…,Cn) where



* The value for a term in this “document” is the average of the values in that position in the entire document in the collection.
* The density is now defined as the average of all similarities with the censored:



* Cheaper to computer.
* Findings:
  + High frequency terms yield negative Discrimination Value.
  + Low frequency terms yield about zero Discrimination Value.
  + Medium frequency terms yield positive Discrimination Value.
* Note the difference between Dvi and [log2(m)-log(Dfi)+1]

where the latter decreases strictly with frequency

**Limitations of the Vector Model**

* Weighting schemes use statistics that are extracted from the entire collection (not just from the current document). These values change continuously as new documents are received, requiring recalculation of weights for old documents.
* Every term in a description of an item is “separate” from every other term. These is no mechanism to precoordinate terms
* Every term in a description of an item is “stored” with a single value. These is no positional information that would facilitate proximity searches

**2. Explain About Hypertext Indexing**

* New class of information representation : a document is a World Wide Web page, with embedded links to other documents(pages)
* Classes of WWW indexing
  + **Manually generated** (e.g. Yahoo!) : pages are indexed manually into a linked hierarchy(an “index”). Users browse in the hierarchy by following links. Eventually, users reach the “end documents”.
  + **Automatically generated** (e.g. Alta Vista) : pages at each Internet site are indexed automatically (creating a “searchable data structure”). These structures are used for querying, rather than browsing.
  + **Crawlers** (e.g. WebCrawler) : No a *priori* indexing. Users define search terms, and the crawler goes to various sites searching for the desired information
* Issue: *could subjects be determined from links ?*
  + The links embedded in each page are indicator of the page’s contents, and could be used in its indexing
  + None of the three indexing methods considers these links to determine the subject of the page
* The dual issue is : *could links be determined from subjects?*
  + Could the system generate the links between items automatically ?
  + Related to the issue of *automatic clustering*

**3. Explain about Data Flow in Information Processing System with neat sketch.**

**Standardize i/p**

Identify Processing Token

**Logical Sub Setting**

Apply stop list

Characterize Token

Apply Stemming

Cerate searchable data structure

Update Document File

User command

Search Results

Display

Query

**4.Explain the Classes of Automatic Indexing.**

Automatic Indexing is the process of analyzing an item to extract the information to be permanently kept in an index.

Search strategies can be classified as

1. Statistical: used in commercial systems, applied to the event data probalistic, Bayesian, vector space, neural net.
2. Natural Language
3. Concept: Words with in an item to correlate to concepts discussed in the item.
4. Hypertext Linkages: Provides virtual threads of concepts between items versus directly defining the concepts with in an item.

**5.What is clustering? Explain**

Clustering: provide a grouping of similar objects into a class under a more general title. Clustering also allows linkage between clusters to be specified. An information database can be viewed as being composed of a number of independent items indexed by a series of index terms

* 1. Term clustering
     1. Used to create a statistical thesaurus
     2. Increase recall by expanding searches with related terms (query expansion)
  2. Document clustering
     1. Used to create document clusters
     2. The search can retrieve items similar to an item of interest, even if the query would not have retrieved the item (resultant set expansion)
     3. Result-set clustering

**Process of Clustering**

* Define the domain for clustering
  + Thesaurus: “medical terms”
  + Documents: set of items to be clustered
  + Identify those objects to be used in the clustering process and reduce the potential for erroneous data that could induce errors in the clustering process
* Determine the attributes of the objects to be clustered
  + Thesaurus: determine the specific words in the objects to be used
  + Documents: may focus on specific zone within the items that are to be used to determine similarity
  + Reduce erroneous association
* Determine the relationships between the attributes whose co-occurrence in objects suggest those objects should be in the same class
  + Thesaurus: determine which words are synonyms and the strength of their relationships
  + Documents: define a similarity function based on word co-occurrences that determine the similarity between two items
* Apply some algorithm to determine the class(es) to which each object will be assigned

**Characteristics of the Classes**

* A well-defined semantic definition should exist for each class
  + There is a risk that the name assigned to the semantic definition of the class could also be misleading
* The size of the classes should be within the same order of magnitude
* Within a class, one object should not dominate the class
* Whether an object can be assigned to multiple classes or just one must be decided at creation time

**6.Explain about Thesaurus Generation.**

Good clustering of terms assists file user by improving recall. But typically an increase in recall has an associated decrease in precision.

The term clustering may be categorized into two types:

1. Manual term clustering 2. Automated term clustering

Manual generation of clusters usually focuses on generating a thesaurus (i.e. on term clustering) and has been used for hundreds of years.

•As items became available in electronic form, automated term clustering techniques became available.

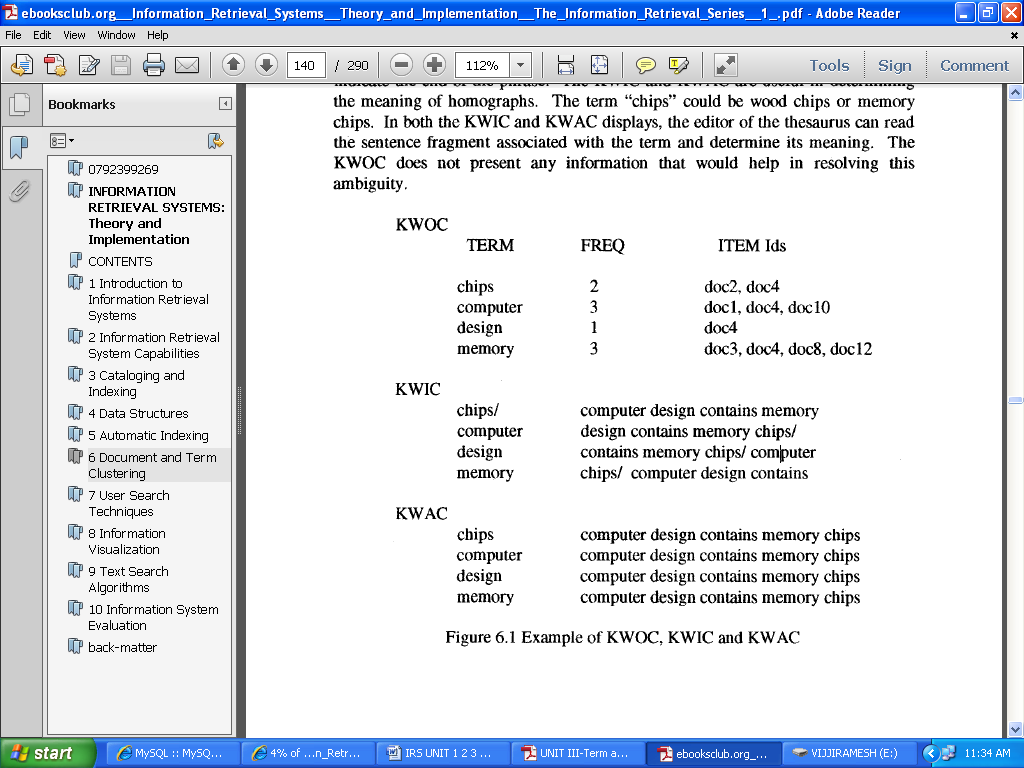
•Automatic clustering has the imprecision of information retrieval algorithms, compounding the natural ambiguities that come from language.

•Care must be taken to ensure that the increases in recall are not associated with such decreases

1. **Manual Clustering:**

The first step is to determine file domain for the clustering. Defining the domain assists in reducing ambiguities caused by Homographs and helps focus file creator. Usually existing thesauri, concordances from items that cover the domain and dictionaries are used as starting points for generating the set of potential words to be included in the new thesaurus. Concordance is an alphabetical listing of words from a set of items along with their frequency of occurrence and references of which items in which they are found. Care is taken to not include words that are unrelated to the domain of the thesaurus or those that have very high frequency of occurrence and thus hold no information value (e.g., the term Computer in a thesaurus focused on data processing machines). If a concordance is used, other tools such as KWOC, KWlC or KWAC may help in determining useful words. A Key Word Out of Context (KWOC) is another name for a concordance. Key Word In Context (KWIC) displays a possible term in its phrase context. It is structured to identify easily the location of the term under consideration in the sentence. Key Word And Context (KWAC) displays the keywords followed by their context. Figure shows the various displays for "computer design contains memory chips" (NOTE: the phrase is assumed to be from doc4; the other frequency and document ids for KWOC were created for this example.)

the character "/" is used in KWlC to indicate the end of the phrase. The KWIC and KWAC are useful in determining the meaning of homographs. The term "chips" could be wood chips or memory chips. In both the KWlC and KWAC displays, the editor of the thesaurus can read the sentence fragment associated with the term and determine its meaning. The KWOC does not present any information that would help in resolving this ambiguity.



Example of KWOC,KWIC,KWAC

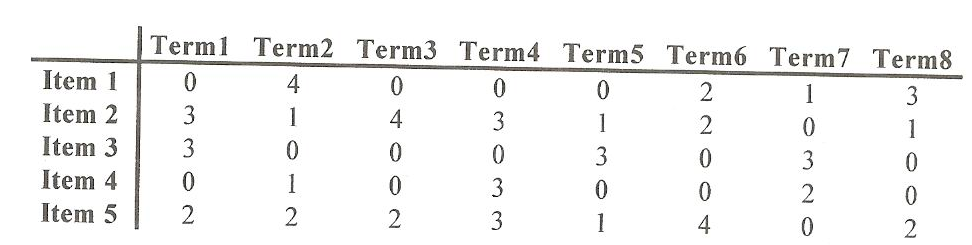
1. **Automatic Term Clustering**

Automatically generated thesauri contain classes that reflect the use of words in the corpus

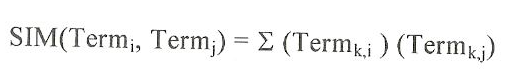
* The classes do not naturally have a name, but are just a groups of statistically similar terms
* Basic idea for term clustering: the more frequently two terms co-occur in the same items, the more likely they are about the same concept
* Term-clustering algorithms differ by the completeness with which terms are correlated
  + The more complete the correlation, the higher the time and computational overhead to create the clusters

1. **Complete Term Relation Method**

* The similarity between every term pair is calculated as a basis for determining the clusters
* Using the vector model for clustering

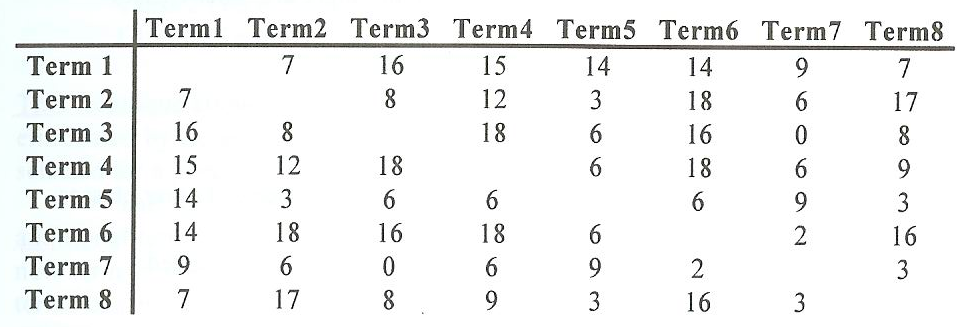


* A similarity measure is required to calculate the similarity between to terms

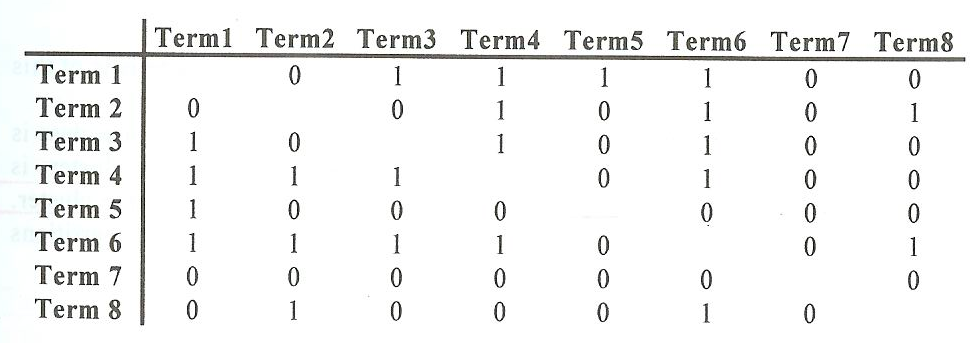


Term1-Term2= (0\*4)+(3\*1)+(3\*0)+(0\*1)+(2\*2)=7

Term1- Term 3=(0\*0)+(3\*4)+(3\*0)+(0\*0)+(2\*2)=16 and so on…



Term-Term Matrix



Term Relationship Matrix

Threshold = 10

* The final step in creating clusters is to determine when two objects (words) are in the same cluster
  + Cliques, single link, stars, and connected components

**Cliques**

* Cliques require all terms in a cluster to be within the threshold of all other terms
  + Class 1 (Term 1, Term 3, Term 4, Term 6)
  + Class 2 (Term 1, Term 5)
  + Class 3 (Term2, Term 4, Term 6)
  + Class 4 (Term 2, Term 6, Term 8)
  + Class 5 (Term 7)

**Clique Algorithm**

Cliques require all items in a cluster to be within the threshold of all other

items. The methodology to create the clusters using cliques is:

0. Leti= 1

1. Select termi and place it ill a new class

2. Start with termk where r = k = i + 1

3. Validate if tern~ is within the threshold of all terms within the current

class

4. If not, let k -- k + 1

5. If k > m (number of words)

then r=r+ 1

if r = m then go to 6 else

k=r

create a new class with termi in it

goto3

else go to 3

6. If current class only has termi in it and there are other classes

with termi in them

then delete current class

else i = i + 1

7. Ifi=m + 1 then go to 8

else go to 1

8. Eliminate any classes that duplicate or are subsets of other classes.

**Single Link**

The rule to generate single link clusters is that any term that is similar to any term in the cluster can be added to the cluster. It is impossible for a term to be in two different clusters. This in effect partitions the set of terms into the clusters. The algorithm is:

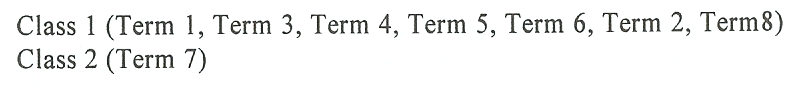
1. Select a term that is not in a class and place it in a new class

2. Place in that class all other terms that are related to it

3. For each term entered into the class, perform step 2

4. When no new terms can be identified in step 2, go to step 1.

Applying the algorithm for creating clusters using single link to the Term Relationship Matrix, the following classes are created:



* Overhead in assignment of terms to classes: O(n2)

**Star**

* Select a term and then places in the class all terms that are related to that term
* Terms not yet in classes are selected as new seeds until all terms are assigned to a class
* There are many different classes that can be created using the Star technique
* If we always choose as the starting point for a class the lowest numbered term not already in a class
  + Class 1 (Term 1, Term 3, Term 4, Term 5, Term 6)
  + Class 2 (Term 2, Term 4, Term 8, Term 6)
  + Class 3 (Term 7)
* Starts with a term and includes in the class one additional term that is similar to the term selected and not already in a class
* The new term is then used as the new node and the process is repeated until no new terms can be added because the term being analyzed does not have another term related to it or the terms related to it are already in the class
* A new class is started with any terms not currently in any existing class
* Using the guidelines to select the lowest number term similar to the current term and not to select any term already in an existing class produces the following classes
  + Class 1 (Term 1, Term 3, Term 4, Term 2, Term 6, Term 8)
  + Class 2 (Term 5)
  + Class 3 (Term 7)

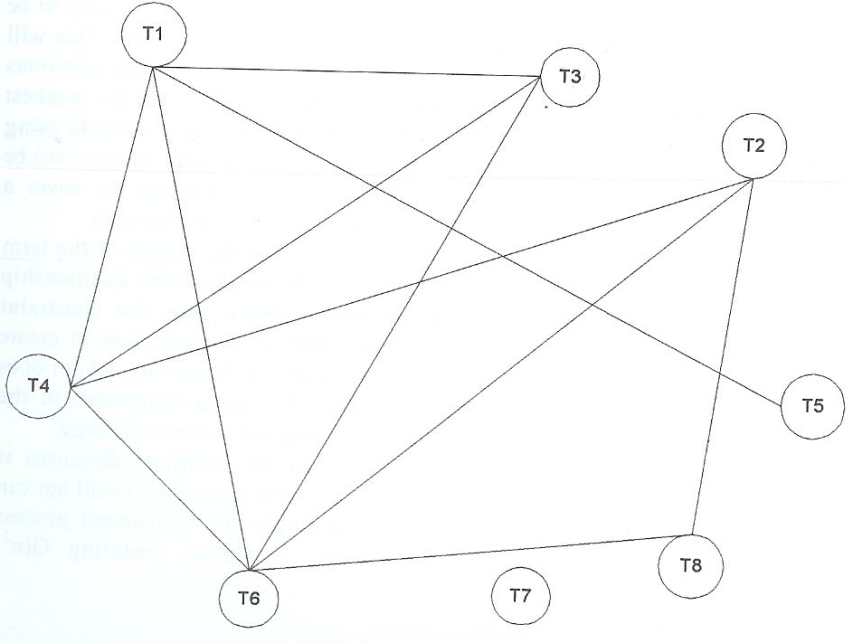
Algorithm:

1. A term not yet in a class is selected

2. All terms similar to it are then placed in its class.

3. Terms not yet in classes are selected as new seeds until all terms are assigned to a class.

**Network Diagram of Term Similarities**

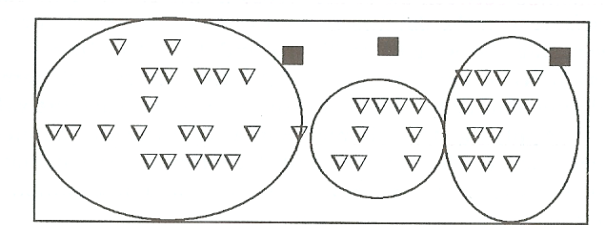
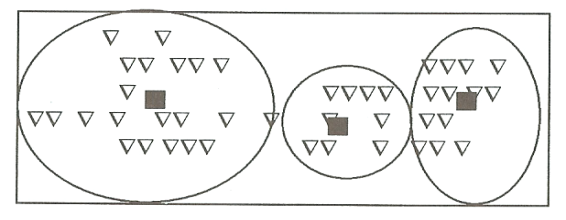


**Comparison**

* Clique
  + Produce classes that have the strongest relationships between all of the words in the class
  + The class is more likely to be describing a particular concept
  + Produce more classes than the other techniques
* Single link
  + Partition the term into classes
  + Produce the fewest number of classes and the weakest relationship between terms
  + It is possible that two terms that have a similarity value of zero will be in the same class
  + Classes will not be associated with a concept but cover diverse concepts
* The selection of the technique is also governed by the density of the term relationship matrix and objects of the thesaurus
  + Term relationship matrix
    - Sparse: toward single link
    - Dense: toward clique
  + Objects of the thesaurus
    - Cliques provide the highest precision when the statistical thesaurus is used for query term expansion
    - The single link algorithm maximize recall but can cause selection of many non-relevant items
* The single link algorithm has the least overhead in assignment of terms to classes: O(n2) comparisons

**Clustering Using Existing Clusters**

* Start with a set of existing clusters
* The initial assignment of terms to the clusters is arbitrary and revised by revalidating every term assignment to a cluster
  + To minimum calculations, centroids are calculated for each cluster
    - Centroid: the average of all of the vectors in a cluster
  + The similarity between all existing terms and the centroids of the clusters can be calculated
  + The term is reallocated to the cluster that has the highest similarity
* The process stops when minimal movement between clusters is detected

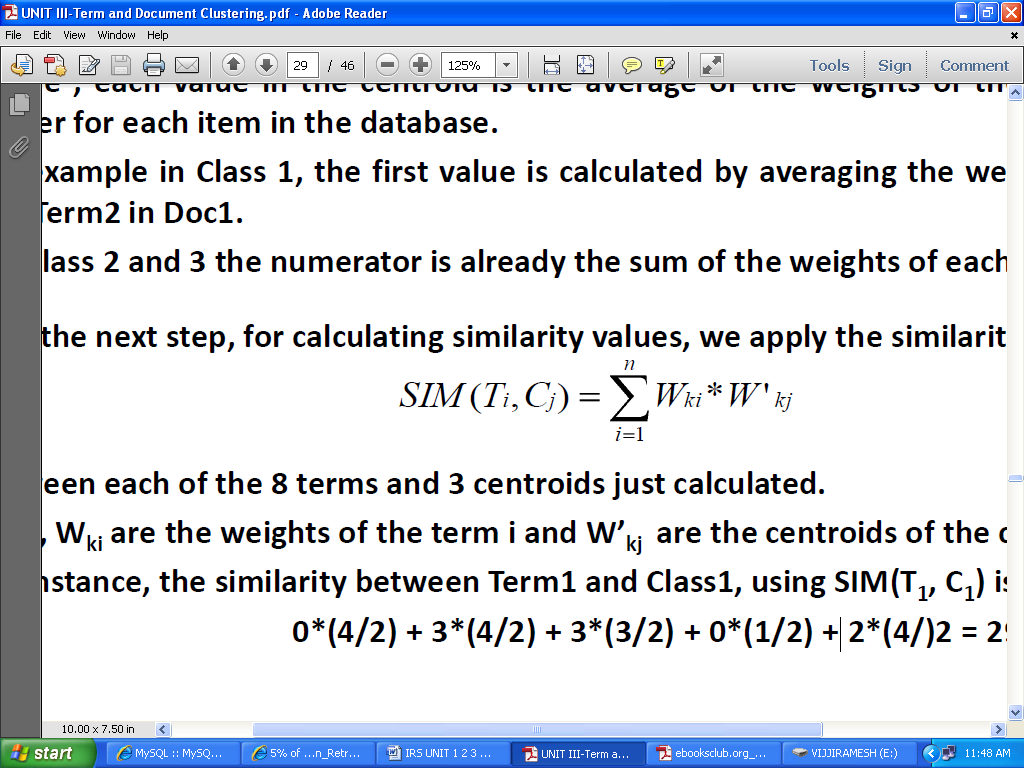


Centroids after reassigning terms

Initial centroids for clusters

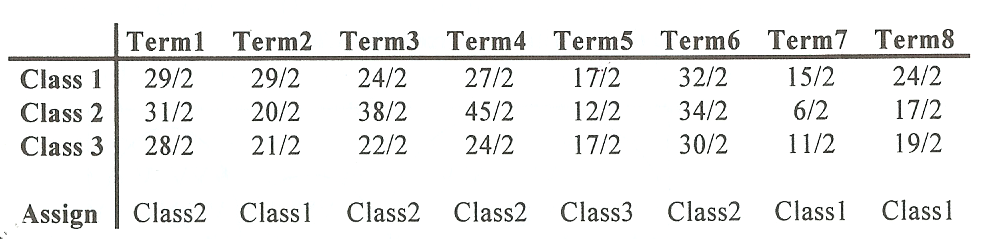
Example

1. Initial assignment
   * Class 1 = (Term 1, Term 2)
   * Class 2 = (Term 3, Term 4)
   * Class 3 = (Term 5, Term 6)
2. Initial centroid
   * Class 1 = (0+4)/2, (3+1)/2, (3+0)/2, (0+1)/2, (2+2)/2  
     = 4/2, 4/2, 3/2, 1/2, 4/2
   * Class 2 = 0/2, 7/2, 0/2, 3/2, 5/2
   * Class 3 = 2/2, 3/2, 3/2, 0/2, 5/2.
3. In the next step, for calculating similarity values, we apply the similarity measure

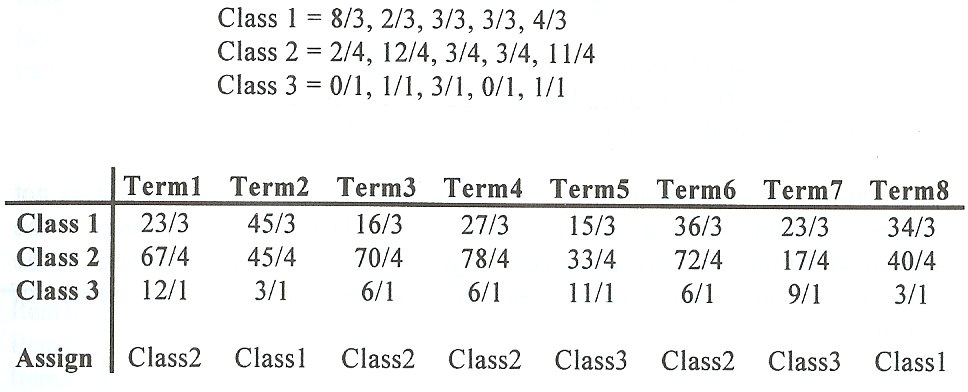


The similarity between Term1 and Class1, using SIM(T1,C1) is

0\*(4/2)+3\*(4/2)+3\*(3/2)+0\*(1/2)+2\*(4/)2=29/2.



Apply the simple similarity measure between each of the 8 terms and 3 centroids.

One technique for breaking ties is to look at the similarity weights of the other terms in the class and assign it to the class that has the most similar weights.

New Centroids and Cluster assignments

* Computation overhead: O(n)
* The number of classes is defined at the start of the process and cannot grow
  + It is possible to be fewer classes at the end of the process
* Since all terms must be assigned to a class, it forces terms to be allocated to classes, even if their similarity to the class is very weak compared to other terms assigned

**One Pass Assignment**

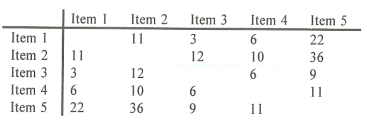
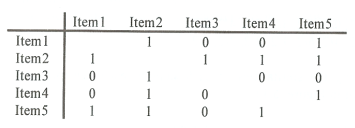
* Minimum overhead: only one pass of all of the terms is used to assign terms to classes
* Algorithm
  + The first term is assigned to the first class
  + Each additional term is compared to the centroids of the existing classes
  + A threshold is chosen. If the item is greater than the threshold, it is assigned to the class with the highest similarity
    - A new centroid has to be calculated for the modified class
  + If the similarity to all of the existing centroids is less than the threshold, the term is the first item in a new class
  + This process continues until all items are assigned to classes
* Example with threshold of 10
  + Class generated
    - Class 1 = Term 1, Term 3, Term 4
    - Class2 = Term 2, Term 6, Term 8
    - Class 3 = Term 5
    - Class 4 = Term 7
  + Centroids values used
    - Class 1 (Term 1, Term 3) = 0, 7/2, 3/2, 0, 4/2
    - Class 1 (Term 1, Term 3, Term 4) = 0, 10/3, 3/3, 3/3, 7/3
    - Class 2 (Term 2, Term 6) = 6/2, 3/2, 0/2, 1/2, 6/2
* Minimum computation on the order of O(n)
* Does not produce optimum clustered classes
* Different classes can be produced if the order in which the items are analyzed changes

7.**Explain about Item Clustering.**

Clustering of items is very similar to term clustering for the generation of thesauri. Manual item clustering is inherent in any library or filing system. In this case someone reads the item and determines the category or categories to which it belongs. When physical clustering occurs, each item is usually assigned to one category. With the advent of indexing, an item is physically stored in a primary category, but it can be found in other categories as defined by the index terms assigned to the item. Think about manual item clustering, which is inherent in any library or filing system – one item one category.

* + Automatic clustering – one primary category and several “secondary” categories
  + Similarity between documents is based on two items that have terms in common
    - The similarity function is performed between rows of the item matrix



**Item/Item and Item Relationship Matrix**

**Clustering Results**



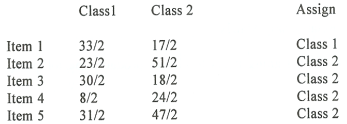
Clique:

Single Link:

Star:



String:



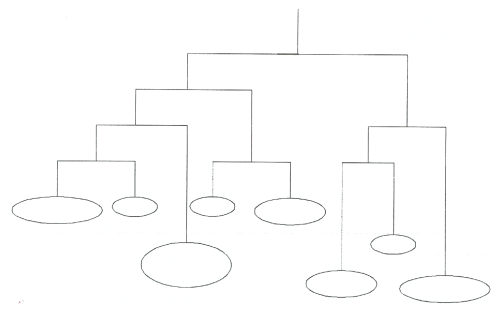
Class 1: {Item 1, Item 3}

Class 2: {Item 2, Item 4}

Clustering with existing clusters

4. Explain about Hierarchy of Clusters

* Hierarchical clustering
  + Hierarchical agglomerative clustering (HAC) – start with un-clustered items and perform pair-wise similarity measures to determine the clusters
  + Hierarchical divisive clustering – start with a cluster and breaking it down into smaller clusters
* Objectives of Create Reduce the overhead of search
  + Perform top-down searches of the centroids of the clusters in the hierarchy and trim those branches that are not relevant
* Provide for visual representation of the information space
  + Visual cues on the size of clusters (size of ellipse) and strengths of the linkage between clusters (dashed line, sold line…)
* Expand the retrieval of relevant items
  + A user, once having identified an item of interest, can request to see other items in the cluster
  + The user can increase the specificity of items by going to children clusters or by increasing the generality of items being reviewed by going to a parent clustering a Hierarchy of Clusters



**Hierarchical Agglomerative Clustering Method (HACM)**

* First the N \* N item relationship matrix is formed
* Each item is placed into its own clusters
* The following two steps are repeated until only one cluster exists
  + The two clusters that have the highest similarity are found
  + These two clusters are combined, and the similarity between the newly formed cluster and the remaining clusters recomputed
* As the larger cluster is found, the clusters that merged together are tracked and form a hierarchy

**Example**

* Assume document A, B, C, D, and E exist and a document-document similarity matrix exists
* {{A} {B} {C} {D} {E}} ⎝ {{A, B} {C} {D} {E}} ⎝ …⎝ {A, B, C, D, E}

**Similarity Measure between Clusters**

* Single link clustering
  + The similarity between two clusters (inter-cluster similarity) is computed as the maximum similarity between any two documents in the two clusters, each initially from a separate cluster
* Complete linkage
  + Inter-cluster similarity is computed as the minimum of the similarity between any documents in the two clusters such that one document is from each cluster
* Group average
  + As a node is considered for a cluster, its average similarity to all nodes in the cluster is computed. It is placed in the cluster as long as its average similarity is higher than its average similarity for any other cluster
* Ward’s method
  + Clusters are joined so that their merger minimizes the increase in the sum of the distances from each individual document to the centroid of the cluster containing it
  + If cluster A merged with either cluster B or cluster C, the centroids for the potential cluster AB and AC are computed as well as the maximum distance of any document to the centroid. The cluster with the lowest maximum is used

**Analysis of HACM**

* Ward’s method typically took the longest to implement
* Single link and complete linkage are somewhat similar in run time
* Clusters found in the single link clustering tend to be fair broad in nature and provide lower effectiveness
* Choosing the best cluster as the source of relevant documents results in very close effectiveness results for complete link, Ward’s and group average clustering
* A consistent drop in effectiveness for single link clustering is noted