MODULE III

Document and Term Clustering

Introduction :-

Clustering :-

↳ Goal - to assist in the location of info.
↳ of words originated with the generation of thesauri.
Thesaurus provides the synonyms and antony of words.

↳ Aim is to provide a grouping of simil objects into a class under a general title.

↳ Allows linkages b/w clusters to be specified.

The steps of clustering :-

(a) Define the domain for the clustering effort :-
↳ identifies those objs to be used in the clustering process and reduce the potential for erroneous data that could induce error in the clustering process.

(b) Once the domain is determined, determine the attributes of the objs to be clustered

(c) Determine the strength of the relationships b/w the attributes whose co-occurrence in objs suggest those objs should be in the same class.

(d) At this point, the total set of objs and the strengths of the relationships b/w the objs have been determined. The final step is applying some alg to determine the class to which each item will be assigned.

→ Guidelines on the characteristics of classes :-

* A well-defined semantic def should exist for each class.
* The size of the classes should be within the same order of magnitude.
* Within a class, one obj should not dominate the class.
* Whether an obj can be assigned to multiple classes or just one must be decided at creation time.

→ Decisions associated with the generation of thesauri and not part of item clustering :-

* Word coordination approach :- specifies if phrases as well as individual terms are to be clustered.
* Word relationships :- when the generation of a thesaurus includes a human interface, a variety of relationships b/w words are possible.
  - ↳ equivalence
  - ↳ hierarchical
  - ↳ non-hierarchical.
* Homograph resolution
* Vocabulary constraints.

I. Thesaurus Generation :- Manual Clustering
Automatic Term Clustering
(i) Complete Term Relation Method.
(ii) Clustering using existing clusters.
(iii) One Pass Assignment

Manual Clustering :-

↳ 1st step is to determine the domain for the clustering.
Defining the domain assists in reducing ambiguities caused by homographs and helps focus the creator.
A concordance is an alphabetical listing of words from a set of items along with their freq of occurrence and references of which items in which they are found.

↳ Main focus in manual thesaurus construct is the selection of the set of words to be include; where words that are not part of the domain are not included.
Also words that have less freq of occurrence are also not included.

↳ A Keywords out of Context (KWOC) is another name of concordance.
Keyword in context (KWIC) displays a possible term in its phrase context
↳ is used to identify easily the locat

Key Word and Context (KWAC) displays the words followed by their content

Eg: "Computer design contains memory chips".

KWOC

Term	Freq	Item Ids
chips	2	doc 2, 4
computer	3	doc 1, 4, 10
design	1	doc 4
memory	3	doc 3, 4, 8, 12

KWIC

chips/	computer design contains memory
Computer	design contains memory chips/
design	contains memory chips/ computer
memory	chips/ computer design contains

KWAC

chips	computer design contains memory chips
computer	"
design	"
memory	"

KWIC & KWAC - determines the meaning of homographs.

Once the terms are selected they are clustered based upon the word relationship guidelines and the interpretation of the strength of the relationship.

Automatic Term Clustering :-

This is based on the concept that the more frequently two terms co-occur in the same items, the more likely they are about the same concept.

The most complete process computes the strength of relationships b/w all the combinations of the 'n' unique words with an overhead of $O(n^2)$.

The simplest case employs one pass of the data in creation of the clusters. When the number of clusters created is very large, the initial clusters may be used as a starting point to generate more abstract clusters creating a hierarchy.

↳ The basis for automatic generation of a thesaurus is a set of items that represents the vocabulary to be included in the thesaurus. The processing tokens in the set of selected items are the attributes to be used to create the clusters.

↳ The automated method of clustering does is based upon the polythetic clustering, where each cluster is defined by a set of words and phrases.

(i) Complete Term Relation Method :-

Here, the similarity b/w every term pair is calculated as a basis for determining the clusters, using the vector model.

The vector model is represented by a matrix where the rows are individual items and the cols are the unique words in the items.

The values in the matrix represents how strongly that a particular word represents the concepts in the term, item.

Eg:

	T1	T2	T3	T4	T5
I1	0	4	0	0	0
I2	3	1	4	3	1
I3	3	0	0	0	3
I4	0	0	0	3	0
I5	2	2	2	3	1

$sim(T_1, T_2) = (0\times4) + (3\times1) + 3\times0 + 0\times0 + 2\times2$

$= 7$

To determine the relationship b/w terms, a similarity measure is used.

$$SIM(Term_i, Term_j) = \Sigma (Term_{k,i})(Term_{k,j})$$

where "k" summed across the set of all items.

These results can be placed in Term - Term matrix.

Using then Item to Term matrix, Term - Term matrix is produced; as show below:

	Term1	2	3	4
Term1	No similarity from t_i to t_i	7	16	15
2	7		8	12
3	16	8		18
4	15	12	18	

$sim(T_1, T_2) = 7$

To obtain this ~~Ede~~ consider the 2 cols of T_1 & T_2 from the above matrix & multi the corresponding freq

There are no values on the diagonal since that represents the autocorrelation of a word to itself.

The next step is to select a threshold that determines if two terms are considered similar enough to each other to be in the same class. This produces a new binary matrix called Term-Relationship matrix, defining similar terms. This is shown below:-

	T1	2	3	4	5
T1		0	1	1	1
2	1		0	1	0
3	1	0			
4	1				
5	1				

1- similar
0- dissimilar

The final step in creating clusters is to determine when two objs are in the same cluster. from the above matrix.

The foll alg are the most common:-
(a) cliques, (b) single link, stars and (c) connected components.

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

Algorithm for cliques:-

1. Let $i = 1$
2. Select $term_i$ and place it in a new class
3. Start with $term_k$ where $k = i + 1$
4. Validate if $term_k$ is within the threshold of all the terms within the current class.

5. If not, let k = k+1

6. If k > m (# of words)

then u = u+1

If u = m then goto 6 else

k = u

creati a new clas with term; in it

goto 3 4

else goto 3f

7. If current class only has term; in it and there are other classes with term; in them

then delete current class

else i = i+1

8. If i = m+1 then goto 9

else goto 2.

9. Eliminati any classes that duplicati or are subsets of other classes.

By applying the alg over the Term-Term matrix, it can be found that the same term can be found in multiple classes.

(b) In single link clustering, the strong constraint that every term in a class is similar to every other term is relaxed.

The rule to generate single link cluster is that any term that is similar to any term in the cluster can be added to the cluster.

Now, it becomes impossible for a term to be in two different clusters.

The algorithm for single-link is as follows:-

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 2, goto 1.

(c) The star technique selects 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. This technique also allows terms to be in multiple clusters.

The string technique starts with a term and inclu in the class one additional term that is simil 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 bcoz the term bei analyzed does not have another term related to it or the terms related to it are already in the class.

(ii) Clustering Using Existing Clusters :-

This methodology reduces the number of similarity calculations required to determine the clusters. The initial assignment of terms to the clusters is revised by revalidating every term assignment to a cluster.

The process stops when minimal movement b/w clusters is detected.

To minimize calculations, centroids are calculated for each cluster, which is the average of all of the vectors in a cluster.

The similarity b/w all existing terms and the centroids of the clusters can be calculated. The term is reallocated to the clusters that has the highest similarity.

Each value in the centroid is the avg of the wts of the terms in the cluster for each item in the database.

The problem is that the no of classes is defined at the start of the process and can not grow. // ①

(iii) One Pass Assignments :-

Here, The 1st term is assigned to the 1st 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 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.

	T1	2	3	4	5	6	7	8
T1		7	16	15	14	14	9	7
2	7		8	12	3	18	6	17
3	16	8		18	6	16	0	8
4	15	12	18			18	6	9
5	14	3	6	6			9	3
6	14	18	16	18	6			16
7	9	6	0	6	9	2		
8	7	17	8	9	3	16	3	

From the above Term x Term matrix, with a threshold to the following classes would be generated :-

class 1 = Term 1, 3, 4
Class 2 = Term 2, 6, 8
class 3 = Term 5
class 4 = Term 7

class 1 (Term 1, 3) = 0, 7/2, 3/2, 0, 4/2
class 1 (Term 1, 3, Term 4) = 0, 10/3, 3/3, 3/3, 7/3
class 2 (Term 2, Term 6) = 6/2, 3/2, 0/2, 4/2, 6/2

II. Item clustering:-

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.

Similarity b/w doc is based upon two items that have terms in common ie terms with items in common.

Thus, the similarity fn is performed b/w rows of the item matrix.

$$SIM(Item_i, Item_j) = \sum (Term_{i,k})(Term_{j,k})$$

k - ranges from 1 to 8 for eight terms. (k^{th} term)

Now an Item-Item matrix is created.

	I_1	I_2	I_3	I_4	I_5
I_1		11	3	6	22
I_2	11		12	10	36
I_3	3	12		6	9
I_4	6	10	6		11
I_5	22	36	9	11	

($Term_1$ in $Item_1$, $Term_1$ in $Item_2$)

$$S(I_1, I_2) = (T_{11} \times T_{21}) + (T_{12} \times T_{22}) + (T_{13} \times T_{23}) + (T_{14} \times T_{24}) \times \cdots (T_{18} \times T_{28})$$

$$= 0 + 4 + 0 + 0 + 0 + 4 + 0 + 3$$

$$= 11.$$

From the above matrix, let us now create an Item Relationship matrix, as shown below by applying a User defined threshold of '10', as shown below:-

	I_1	I_2	I_3	I_4	I_5
I_1		1	0	0	1
I_2	1		1	1	1
I_3	0	1		0	0
I_4	0	1	0		1
I_5	1	1	0	1	

Using clique alg for assigning items to classes produces the foll classes based upon the above matrix :-

Class 1 = Item 1, Item 2, Item 5

Class 2 = Item 2, Item 3

Class 3 = Item 2, Item 4, Item 5.

Applying single link technique produces :-

Class 1 = Item 1, Item 2, Item 5, Item 3, Item 4.

Here Item 3 & 4 are added because of their similarity to Item 2.

The star technique produces :

Class 1 - Item 1, 2, 5

Class 2 - Item 3, 2

Class 3 - Item 4, 2, 5.

clustering by starting with existing clusters can be performed in a manner similar to term model.

Lets start with item1 & 3 in class 1, and item 2 & 4 in class 2. Then the centroids are:-

class1 = 8/2, 4/2, 0/2, 0/2, 3/2, 2/2, 4/2, 3/2

class2 = 8/2, 2/2, 4/2, 6/2, 1/2, 2/2, 2/2, 1/2.

The results of recalculating the similarities of each item to each centroid and reassigning items is as shown:-

	class 1	class 2	Assign
I_1	38/2	19/2	class1
I_2	23/2	51/2	class 2
I_3	30/2	18/2	{class 1} → class 2.
I_4	8/2	24/2	class 2
I_5	31/2	47/2	class 2

After calculating the centroid of class 2, item 3 will also take place in class 2.

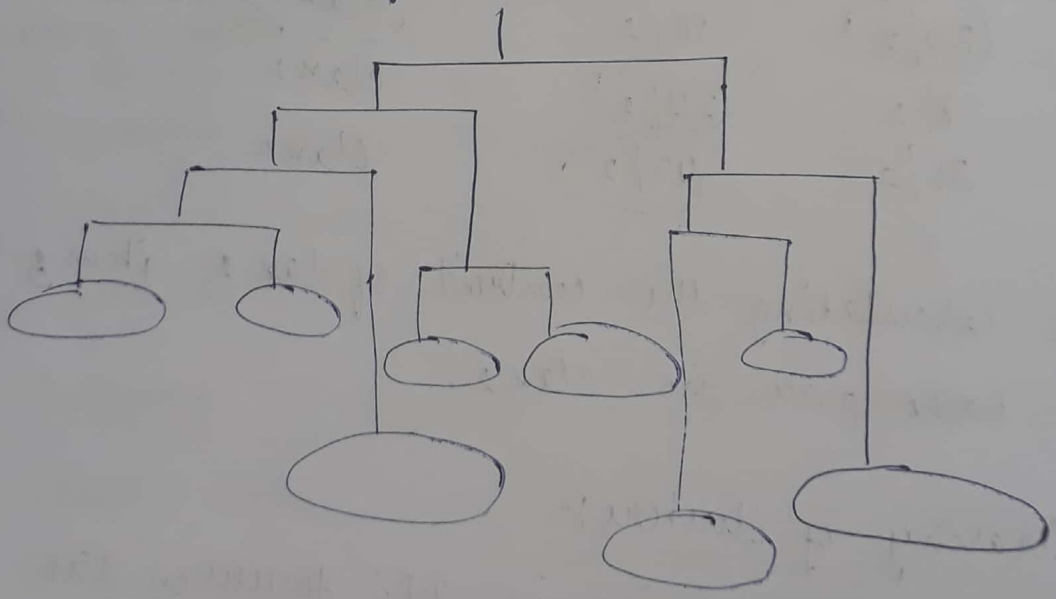
III. Hierarchy of clusters:-

Hierarchical clustering in IRS focuses on the area of hierarchical agglomerative clustering methods. The term agglomerative means the clustering process starts with unclustered items and performs pairwise similarity measures to determine the clusters. Divisive is the term applied to starting with a cluster and breaking it down into smaller clusters.

Objectives:- * Reduce the overhead of search.
* Provide for a visual rep of info space
* Expand the retrieval of relevant items

Search overhead is reduced by performing top-down searches of the centroids of the clusters in the hierarchy and trimming those branches that are not relevant.

Use of dendograms along with visual cues on the size of clusters and strengths of linkage b/w clusters allows a user to determine alternate paths of browsing database.



Even without the visual display of the hierarchy, a user can use the logical hierarchy to browse items of interest.
A user, once having identified an item of interest, can request to see other items in the cluster.

Sanderson & Croft proposed the following methodology to build a concept hierarchy. This had also considered the ways of extracting terms from the docs to rep the hierarchy.

The terms had the foll characteristics:-

* Terms had to best reflect the topics.
* A parent term would refer to a more general concept than its child.
* A child would cover a related subtopic of the parent.
* A directed acyclic graph would represent relationships as a pure hierarchy.
* Ambiguous terms would have seperate entries in the hierarchy for each meaning.

User Search Techniques

To understand the search process, it is first necessary to look at the different binding levels of the search st entered by the user to the db being searched.

The selection and ranking of items is accomplished via similarity measures that calculate the similarity b/w the user's search st and the wt'd stored rep of the semantics of an item. Searching centroids can reduce search computation, but there is an associated risk of missing relevant items because of the averaging nature of centroid.

Hyperlinked items introduced new concepts in search originating from the dynamic nature of

(1) Search Statements & Binding:-

Search st are the sts of an info need generated by users to specify the concepts they are trying to locate in items. The search st uses traditional boolean logic and/or Natural lang.

Binding is when a more abstract form is redefined into a more specific form. The search statement is the user's attempt to specify the conditions needed to subset logically the total item space to that cluster of items that contains the info needed by the user. The next level of binding comes when the search st is passed from use by a specific search sys. The search st sys translates the query to its own meta language. This process is similar to the indexing of item processes.

The final level of binding comes as the search is applied to a specific database. This binding is based upon the statistics of the processing tokens in the database and the semantics used in the database. This is especially true in statistical & concept indexing sys. Frequently, in concept indexing sys, the concepts that are used as the basis for indexing are determined by applying a statistical alg against a rep sample of database.

Input	Binding
"Find me info on the impact of the oilspills in Alaska on the price of oil"	User search it using vocabulary of user.
impact, oil (petroleum), spills (accidents), Alaska, price (cost, value)	statistical binding extracts processing tokens.
impact (0.308), oil (0.606),	Weights assigned to search terms based upon inverse doc freq alg and database.

The above mentioned are the 3 potential levels of binding.

Parenthesis are used in the second binding step to indicate expansion by a thesaurus.

Ⓐ The length of search str directly affect the ability of IRS to find relevant items.

The longer the search query, the easier it is for the sys to find items.

* Profiles ~~are~~ used as search str for selective Dissimination of Info sys. are usually very long, typically 75 to 100 terms.

(2) Similarity Measures & Ranking :- Similarity Measures
HMM Techniques
Ranking Alg.

Searching in general is concerned with calculating the similarity b/w a user's search & the items in the database.
The similarity may be applied to the total item or constrained to logical passages in the item.

↳ Limiting the def of a passage to a fixed length size, locality based searching and similarity allows variable length passages based upon similarity of content.
This then leads to the ability to define locality based searching and retrieval of the precise locations of info that satisfies the query.

* Restricting the similarity measure to passages gains significant precision with minimal impact on recall.
The lack of a large no of terms makes it harder to find shorter passages that contain the search terms expanded from the shorter queries.

⇒ Once items are identified as possibly relevant to the user's query, it is best to present the most likely relevant items first. This process is called "Ranking".

(1) Similarity Measures;-
A characteristic of a similarity formula is that the results of the formula increase as the items become more similar.
The value is zero if the items are totally dissimilar.

$SIM(Item_i, Item_j) = \Sigma (Term_{i,k})(Term_{j,k})$

If $Item_j$ is replaced with $Query_j$ then the same formula generates the similarity b/w every Item and $Query_j$.

The ~~bas~~ problem with this simple measure is in the normalization needed to account for variances in the length of items.

→ Robertson & Spark Jones suggests that knowledge of terms in relevant items retrieved from a query should adjust the weights of those terms in the weighting process.

Taking into account the freq of occurrence of terms within an item producing the following similarity formula, developed by Croft.

$$SIM(DOC_i, Query_j) = \sum_{i=1}^{Q} (CC + IDF_i) * f_{i,j}$$

where C - constant used in tuning.

IDF_i - inverse doc freq for term i.

$$f_{i,j} = k + (k-1) TF_{i,j} / maxfreq_j$$

where k - tuning constant.

$TF_{i,j}$ - freq of $term_i$ in item j

$maxfreq_j$ - max freq of any term in item j.

The best values of k ranges b/w 0.3 & 0.5.

To determine the "weight", an item has w.r.t. the search st, the cosine formula is used to calculate the distance b/w the vector for the item and the vector for the query i-

$$SIM(Doc_i, Query_j) = \frac{\sum_{k=1}^{n} (Doc_{i,k} * QTerm_{j,k})}{\sqrt{\sum_{k=1}^{n} (Doc_{i,k})^2 * \sum_{k=1}^{n} (QTerm_{j,k})^2}}$$

where $Doc_{i,k}$ – k^{th} term in the weighted vector for Item "i".

$QTerm_{j,k}$ – k^{th} term in the query "j"

The cosine formula calculates the cosine of the angle b/w the two vectors.

If the two vectors are totally unrelated, then they will be orthogonal and the value of COSINE is "0" otherwise "1".

Salton & Buckley improved the above formula as :-

$$QTerm_{i,k} = (0.5 + (0.5\, TF_{i,k} / maxfreq_k)) * IDF_i$$

where $TF_{i,k}$ is the freq of term 'i' in the query 'k'

$maxfreq_k$ is the max freq of any term in query 'k' and IDF_i is the Inverse Doc Freq for term 'i'.

→ Commonly used other measures are the Jaccard & the Dice similarity measures. Here, there is a change in the normalizing factor in the denominator to account for diff characteristics of the data.

The denominator in the cosine formula is invariant to the no of terms in common and produces very small nos when the vectors are large and the no of common elts is small.

* The Jaccard formula is as follows:-

$$SIM(DOC_i, Q_j) = \frac{\sum_{k=1}^{n} (DOC_{i,k} * QTerm_{j,k})}{\sum_{k=1}^{n} DOC_{i,k} + \sum_{k=1}^{n} QTerm_{j,k} - \sum_{k=1}^{n} (DOC_{i,k} * QTerm_{j,k})}$$

As the common elts increase, the similarity value decreases, but is always in the range -1 to +1.

* The Dice formula is :-

$$SIM(DOC_i, Query_j) = \frac{2 * \sum_{k=1}^{n} (DOC_{i,k} * QTerm_{j,k})}{\sum_{k=1}^{n} DOC_{i,k} + \sum_{k=1}^{n} QTerm_{j,k}}$$

Eg:- Query = (2, 2, 0, 0, 4)

DOC_1 = (0, 2, 6, 4, 0)

DOC_2 = (2, 6, 0, 0, 4)

	Cosine	Jaccard	Dice
DOC_1	36.66	16	20
DOC_2	36.66	-12	20

Note that : as long as the vector values are same independent of their order, the cosine and Dice normalization factor do not change.

Many of the items have a similarity close or equal to zero.
For this reason, thresholds are usually associated with the search process.
The threshold defines the items in the result - ant hit file from the query.

→ A dysart is always the case where the similarity is greater than zero.
The foll Eg illustrates the threshold process:-

Vector: American, geography, lake, Mexico, painter, oil, reserve, subject

DOC1 geography of Mexico suggests oil reserves are available vector (0, 1, 0, 1, 0, 1, 0, 1)

DOC2 American geography has lakes available everywhere.
Vector (1, 3, 2, 0, 0, 0, 0, 0)

DOC3 painters suggest Mexico lakes as subject
vector (0, 0, 1, 3, 3, 0, 0, 2)

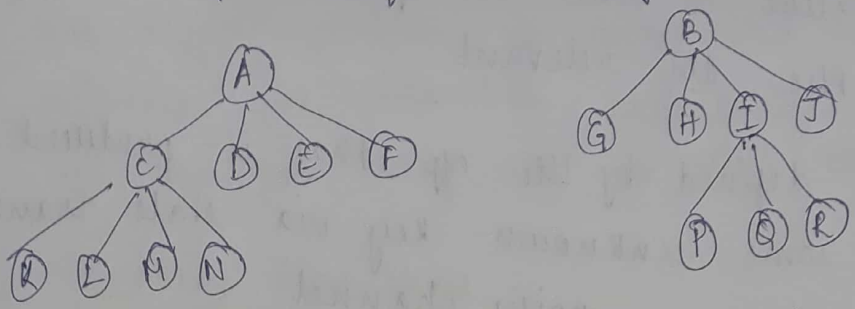
Query oil reserves in Mexico
vector (0, 0, 0, 1, 0, 1, 1, 0)

SIM(Q, DOC1) = 6
SIM(Q, DOC2) = 0
SIM(Q, DOC3) = 3

Here, the simple "sum of the products" similarity formula is used to calculate similarity b/w the query and each document.

If no threshold is specified, all three docs are considered hits. If a threshold of 4 is selected, then only DOC1 is returned.

Another ipl technique is to represent the clusters in the form of hierarchy.



The items are stored in clusters that are represented by the centroid for each cluster. In each letter at the leaf node represent an item ie K, L, M... The letters at the higher nodes A, C, B, I represent the centroid of their immediate children nodes.

The hierarchy is used in search by performing a top-down process. The query is compared to the centroids "A" and "B".

If the results of the similarity measure are above the threshold, the query is then applied to the nodes children. If not, that part of the tree is pruned and not searched. This continues until the actual leaf nodes that are not pruned are compared.

Use of centroids reduces the similarity computations but could cause a decrease in recall. It should have no effect on precision since that is based upon the similarity calculations at the leaf level.

(ii) Hidden Markov Models Techniques:-

↳ used for searching textual corpora.

Here, the docs are considered as unknown statistical processes that can generate o/p that is equivalent to the set of queries that would consider the doc relevant.

HMM is defined by the o/p that is produced by passing some unknown key via state transitions through a noisy channel.

The observed o/p is the query, and the unknown keys are the relevant documents.

The noisy channel is the mismatch b/w the author's way of expressing ideas and the user's ability to specify his query.

For each doc the probability that D was the relevant doc in the user's mind given that Q is the query produced is:

$$P(D \text{ is } R/Q) = P(Q/D \text{ is } R) * P(D \text{ is } R) / P(Q)$$

↳ This is application of Bayes rule to conditional probability.

Relevant doc sets seem to be so sensitive to the specific queries, that trying to estimate P(D) doc not return any noticeable improvements in query resolution. Thus the probability that a doc is relevant given a specific query can be estimated by calculating the probability of the query given the doc is Relevant, i.e. P(Q/D

A HMM is defined by a set of states, a transition matrix defining the probability of moving b/w states, a set of o/p symbols and the probability of the o/p symbols given a particular state.

The biggest problem in using this approach is to estimate the transition probability matrix and the o/p for every doc in the corpus.

(iii) Ranking Algorithms :-

Ranking the o/p implies ordering the o/p from most likely items that satisfy the query to least likely items. This reduces the user overhead by allowing the user to display the most likely relevant items first.

→ With the inclusion of statistical similarity techniques into commercial systems and the large no of hits that originate from searching device corpora, such as the Internet, ranking has become a common feature of modern systems.

In commercial systems, heuristic rules are used to assist in the ranking of items.

Ranking < Coarse grain ranking
Fine grain ranking

The coarse grain ranking is based on the presence of query terms within items.

In the fine grain ranking, the exact rank of the item is calculated.

The coarse grain ranking is a weighted formula that can be adjusted based on completeness, contextual evidence or variety and semantic distance.

↳ Completeness is the proportion of the no of query terms found in the item vs the no in the query.

↳ Contextual evidence occurs when related words from the semantic n/w are also in the item.

↳ Semantic distance evaluates how close the additional words are to the query term.
Synonyms add additional wt, antonyms decrease wt.

↳ The coarse grain process provides an initial rank to the item based upon existence of words within the item.

↳ Fine grain ranking considers the physical location of query terms and related words using factors of proximity in addition to the other three factors in coarse grain evaluation.

(3) Relevance Feedback :-

One of the major problems in finding relevant items lies in the difference in vocabulary b/w the authors and the users.

The user can use relevant items that have been found by the sys to improve future searches, which is the basis behind relevance feedback. Relevant items are used to reweight the existing query terms and possibly expand the user's search st with new items.

The relevance feedback concept was that the new query should be based on the old query modified to increase the wt of terms in relevant items and decrease the wt of terms that are in non-relevant items.

$$Q_n = Q_0 + \frac{1}{r} \sum_{i=1}^{r} DR_i - \frac{1}{nr} \sum_{j=1}^{nr} DNR_j$$

where

Q_n = the revised vector for the new query

Q_0 = original query

r = # of relevant items

DR_i = vectors for the relevant items

nr = # of non relevant items

DNR_j = vectors for the non-relevant items.

The revised version of the above formula is :-

$$Q_n = \alpha Q_0 + \beta \sum_{i=1}^{r} DR_i - \gamma \sum_{j=1}^{nr} DNR_j$$

where α, β, γ are the constants associated with each factor usually $1/n$ or $1/nr$ times a const.

$\beta \sum_{i=1}^{r} DR_i$ — +ve feedback

$\gamma \sum^{nr} DR_i$ — −ve feed back

↳ +ve feedback is more likely to move a query closer to user's info needs.

-ve feedback actually reduces the effectiveness of a query.

+ve feedback moves the query to retrieve items similar to the items retrieved and thus in the direction of more relevant items.

-ve feedback moves the query away from the non-relevant items retrieved.

	T_1	T_2	T_3	T_4	T_5
Q_0	3	0	0	2	0
$DOC1_r$	2	4	0	0	2
$DOC2_r$	1	3	0	0	0
$DOC3_{nr}$	0	0	4	3	3
Q_n	$3\frac{3}{4}$	$1\frac{3}{4}$	0	$1\frac{1}{4}$	0

In the above table, Q_n is obtained as follows:-

Here there are 3 items — 2 relevant and 1 non-relevant

Let us assume $\alpha = 1$, $\beta = \frac{1}{4}$ ($\frac{1}{2}$ times $\frac{1}{2}$)

$\gamma = \frac{1}{4}$ ($\frac{1}{1}$ times a constant $\frac{1}{4}$)

Now, sub these values in the formula for Q_n

$$\Rightarrow Q_n = (3, 0, 0, 2, 0) + \frac{1}{4}(2+1, 4+3, 0+0, 0+0, \ldots) - \frac{1}{4}(0, 0, 4, 3, 3)$$

$$= (3\frac{3}{4}, 1\frac{3}{4}, 0, \{-1\}, 1\frac{1}{4}, 0)$$

↓ changed

Using the normalized similarity formula,

$$SIM(Q_k, DOC_i) = \sum_{i=1}^{5} Term_{k,i} * Term_{1,i}$$

Applying this results in :-

	DOC1	DOC2	DOC3
Q_0	6	3	6
Q_n	14½	9.0	3.75

↳ Relevance feedback, (+ve feedback) has been proven to be of significant value in producing better queries.

↳ When the original query is modified based upon relevance feedback, the systems ensure that the original query terms are in the modified query, even if -ve feedback would have elimi-nated them.

↳ Queries using relevance feedback produce significantly better results than those being manually enhanced.

↳ Pseudo- relevance feedback :- when users enter queries with a few no of terms, automatic relevance feedback based upon just the rank values of the items has been used. This is also referred to as: blind feedback, or local context analysis.

(4) Selective Dissemination of Info Search :-

A dissimination sys is sometimes labeled as a "push" sys while a search sys is called a "pull" sys. The differences are that in a search sys the user proactively makes a decision that he needs info and directs the query to the info sys to search.

In a dissemination sys, the user defines a profile and as new info is added to the sys it is automatically compared to the user profile.

A dissemination sys does not necessarily have a retrospective database associated with it.

One of the 1st commercial search techniques for dissemination was the Logicon Message Dissemination System (LMDS). It was designed for speed to support the search of thousand of profiles with items arriving every 20 sec. This sys uses a least frequently occuring trigraph (3 characters) alg that quickly identifies which profiles are not satisfied by the item.

The potential hits profiles are analyzed in detail to confirm if the item is a hit.

→ An eg of a dissemination approach is the Personal Library s/w sys (PLS). This uses the approach of accumulating newly received items into the database and periodically running users profiles against the database. This makes max use of the retrospective search s/w but loses near real time delivery of items.

Eg: Retrieval Ware & Inroute s/w sys.

Retrieval ware uses a statistical alg but it does not include any corpora data.

INQUERY sys used against retrospective database, uses IDF info. It creates this info as it processes items, storing and modifying it for use as future items arrive.

The dissemination process is continuous, and the issue is the practicality of archiving all of the relevance judgements to be used in the relevance feedback process.

→ Another approach to dissemination uses a statistical classification approach and explicit error minimization to determine the decision criteria for selecting items for a particular profile.

In this case, the classification process is related to assignment for each item into one of two classes: * relevant to a users profile
* non- relevant.

↳ Eg of alg used in linear classifiers that perform explicit error minimization are linear discriminant analysis, logistic regression and linear neural n/w.

*↳ Schutze et al. used two approaches to reduce dimensionality : * selecting a set of existing features to use
* creating a new much smaller set of features that the original features are mapped into.

→ A X^2 measure was used to determine the most imp features.
The list was applied to a table that contained the no of relevant - N_r and non-relevant N_{nr} items in which a term occurs plus the no of relevant and non-relevant items in which the term does not occur ie N_{r-}, N_{nr-} resp.

$$X^2 = \frac{N(N_r N_{nr} - N_{r-} N_{nr})^2}{(N_r + N_{r-})(N_{nr} + N_{nr-})(N_r + N_{nr})(N_{r-} + N_{nr-})}$$

The chi-squared technique provides a more effective mechanism than freq of occurrence of terms. A high X^2 score indicates a feature whose freq has a significant dependence on occurrence in a relevant or non-relevant item.

An alternative technique to identify the reduced feature (vector) set is to use a modified Latent semantic index (LSI) technique to determine a new reduced set of concept vectors.

The use of the profile to define a local region is essential when working with large databases.

Linear discriminate analysis, logistic regression and neural nw are three possible techniques that were compared by Schutze et al.
Linear discrimination analysis uses the covariance class for each doc class to detect feature dependence.

Assuming a sample of data from two groups with n_1 and n_2 members, mean vectors $\bar{x}_1$ & $\bar{x}_2$ and covariance matrices C_1 and C_2 resp, the objective is to maximize the seperation b/w the two groups.
This can be achieved by maximizing the distance b/w the vector means, scaling to reflect the structure in the pooled covariance matrix. Thus choose a such that :-

$$a^* = \arg_a \max \frac{a^T(\bar{x}_1 - \bar{x}_2)}{\sqrt{a^T C a}}$$

a^* is maximized where T is the transpose and

$(n_1 + n_2 - 2)C = (n_1 - 1)C_1 + (n_2 - 1)C_2$.

∵ C is +ve, the Cholesky decomposition of $C = R^T R$.

Let $b = Ra$, then the formula becomes:

$$a^* = \arg_b \max \frac{b^T R^{T-1} (\bar{x}_1 - \bar{x}_2)}{\sqrt{b^T b}}$$

which is maximized by choosing $b \propto R^{T-1}(\bar{x}_1 - \bar{x}_2)$

This means :-

$$a^* = R^{-1} b = C^{-1} (\bar{x}_1 - \bar{x}_2)$$

The one dimensional space defined by $y = a^{*T} x$ should cause the group means to be well seperated.

Regularized Discriminant Analysis:- is to produce a non-linear classifier, a pair of shrinkage parameters is used to create a very general family of estimators for the group convariance matrix.

→ A 2nd approach is to use logistic regression proposed by Cooper. It models a binary response variable by a linear combination of one or more predictor variables, using a logit link function :-

$$g(\pi) = \log(\pi / (1 - \pi))$$

This is achieved by modeling the dependent variable $\log(\pi/(1-\pi))$ as a linear combination of independent variables using a form $g(\pi) = x_i \beta$.

Here π - estimated response probability. ie probability of relevance.

x_i - feature vector for document I

β - weight vector which is estimated from the matrix of feature vectors.

The optimal value of β can be calculated using the max likelihood and the Newton - Raphson method of numerical optimization.

→ A 3rd technique is to use neural n/w for the learning fn.

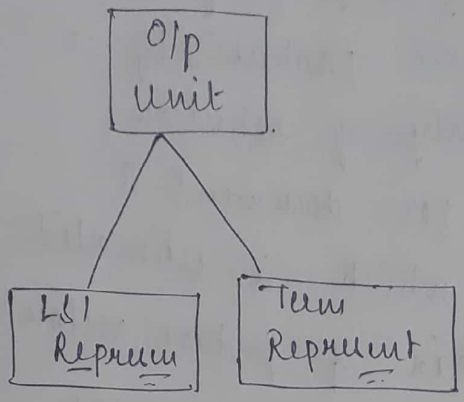
A neural n/w is a n/w of i/p and o/p cells - based upon neuron fns in the brain, originati - ng with the work of Mc Culloch and Pitts.

Each i/p pattern is propagated forward through the n/w.

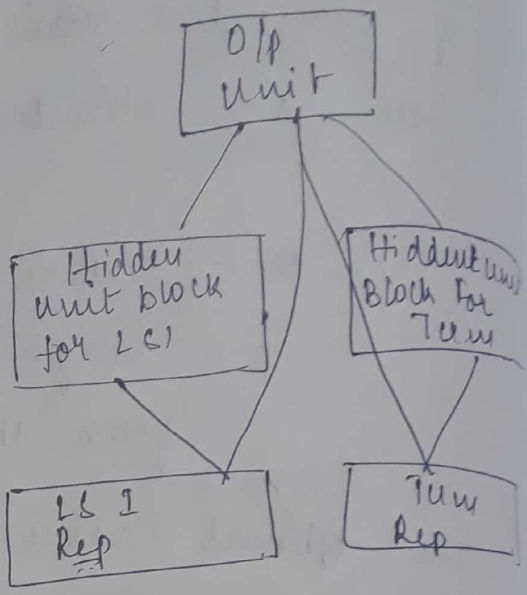
When an error is detected it is propagated backward adjusting the cell parameters to reduce the error, thus achieving learning. This technique is very flexible and can accommodate a wide range of distributions.

Whenever the errors on the validation set increase, it indicates that overfitting is occuring and establishes the no of iterations on training that improve the parameter values while not harming

The linear & non-linear architectures for an implementation of neural n/w are as follows:-



Linear neural n/w



Non linear Neural n/w

In the non-linear n/w, each of the hidden blocks consists of three hidden units. A hidden unit can be interpreted as feature detectors that estimate the probability of a feature being present in the i/p. Propagating this to the o/p unit can improve the overall estimation of relevance in the o/p unit.

The n/w show the i/p of both terms and the LSI rep.

Relevance is computed by setting the activations of the i/p units to the doci rep and propagating the activation through the n/w to the o/p unit, then propagating the error back through the n/w using a gradient descent

alg as proposed by Rumelhart.

A sigmoid was chosen as:-

$$f(x) = \frac{e^x}{1+e^x}.$$

This is an activation fn for the units of the n/w.

In this case backpropagation minimizes the same error as logistic regression.

The cross-entropy error is:-

$$L = -\Sigma (t_i \log \sigma_i + 1 - t_i) \log (1 - \sigma_i)$$

where t_i is the relevance for doc I and σ_i is the estimated relevance (activation of the o/p unit) for doc i.

The def of the sigmoid is:-

$$x = \log \left(\frac{f(x)}{1 - f(x)} \right).$$

(5) Weighted Searches of Boolean Systems:-

The two major approaches to generating queries are Boolean and natural lang.

Natural lang queries are easily represented within statistical models and are usable by the similarity measures discussed.

Issues arise when Boolean queries are associated with weighted index sys.

Some of the issues are associated with how the logic operators fn with weighted values and how weights are associated with the query terms.

To integrate the Boolean and weighted systems model, Fox and Sharat proposed a fuzzy set approach.

Fuzzy sets introduce the concept of degree of membership to a set.

The degree of membership for AND and OR operations are defined as:

$$DEG_{A \cap B} = \min(DEG_A, DEG_B)$$

$$DEG_{A \cup B} = \max(DEG_A, DEG_B)$$

where A and B are terms in an item.

DEG is the degree of membership.

The Mixed Min and Max (MMM) model considers the similarity b/w query and doc to be a linear combination of the min and max item wts.

Fox proposed the foll similarity formula:-

$$SIM(QUERY_{OR}, DOC) = C_{OR_1} * \max(DOC_1, DOC_2 \cdots DOC_n) + C_{OR_2} * \min(DOC_1, DOC_2, \cdots DOC_n)$$

$$SIM(QUERY_{AND}, DOC) = C_{AND_1} * \min(DOC_1, DOC_2 \cdots DOC_n) + C_{AND_2} * \max(DOC_1, DOC_2, \cdots DOC_n)$$

where C_{OR_1} and C_{OR_2} are wtring coefficients for the OR operation and

C_{AND1} and C_{AND2} – wt'ing coeff for the AND.

Lee and Fox found in their experiments that the best performance comes when C_{AND1} b/w 0.5 to 0.8 and $C_{OR1} > 0.2$.

The MMM technique was expanded by Paice considering all item wts as the max/min approach. The similarity measure is :-

$$SIM(QUERY, DOC) = \sum_{i=1}^{n} r^{i-1} d_i \; / \; \sum_{i=1}^{n} r^{i-1}$$

where d_i's are inspected in ascending order for AND queries and descending order for OR queries

r – wting coeff.

Best values for r are 1.0 for AND queries and 0.7 for OR queries

→ An alternative approach is using the P-norm model which allows terms within the query to have wts in additions to the terms in the items.

Similar to the cosine similarity technique, it considers the membership values $(d_{A1}, \ldots d_{AN})$ to be coordinates in an "n" dimensional space.

For an OR query, the origin is the worst possibility.

For an AND query the ideal pt is the unit vector where all the D_i values equal 1. Thus the best ranked docs will have max distance from the origin in an OR query and minimal distance from the unit vector pt. The generalized queries are:-

$$Q_{OR} = (A_1, a_1) \text{ OR } (A_2, a_2) \text{ OR } \cdots \text{ OR } (A_n, a_n)$$

$$Q_{AND} = (A_1, a_1) \text{ AND } (A_2, a_2) \text{ AND } \cdots \text{ AND } (A_n, a_n)$$

The operators will have a strictness value assigned that varies from 1 to ∞ where '∞' is the strict def of Boolean operator.

The 'a_i' values are the query term wts.

If we assign the strictness values to a parameter labeled "S" then the similarity formulas b/w queries and items are:-

$$SIM(Q_{OR}, DOC) = \sqrt[S]{(a_1^S d_{A1}^S + \cdots a_n^S d_{An}^S) / (a_1^S + a_2^S + \cdots a_n^S)}$$

$$SIM(Q_{AND}, DOC) = 1 - \sqrt[S]{a_1^S (1 - d_{A1})^S + \cdots + a_n^S (1 - d_{An})^S} \Big/ (a_1^S + a_2^S + \cdots + a_n^S)$$

$$SIM(Q_{not}, DOC) = 1 - SIM(Q, DOC)$$

→ The objective of Salton's proposal is to perform the normal Boolean operations and then refine the results using wtng techniques.

The foll procedure is a modification to his approach for defining search results.

→ Normal Boolean operations produce the foll results:

* "A OR B" retrieves those items that contain the term A or the term B or both.
* "A AND B" retrieves those items that contain both terms A and B.
* "A NOT B" retrieves those items that contain term A and not contain term B.

If wts are then assigned to the terms b/w the values 0.0 to 1.0, they may be interpreted as the significance that users are placing on each term.

The value 1.0 is assumed to be the strict interpretation of a Boolean query.

0.0 is interpreted to mean that the user places little value on the term.

Under these assumptions, a term assigned a value of 0.0 should have no effect on the retrieved set.

Thus:

* "A_1 or B_0" should return the set of items that contain A as a term.
* "A_1 AND B_0" will also return the set of items that contain term A.
* "A_1 NOT B_0" also return set A.

The alg follows the foll steps:-

1. Determine the items that are satisfied by applying strict interpretation.
2. Determine the items that are part of the set that is invariant.
3. Determine the centroid of the invariant set.
4. Determine the no of items to be added or deleted by multiplying the term wt times the no of items outside of the invariant set and rounding up to the nearest whole no.
5. Determine the similarity b/w items outside of the invariant set and the centroid.
6. Select the items to be included or removed from the final set.

Eg:- ~~Query, ends up with~~ ~~text~~

Weighted Boolean Query.

	Computer	Program	cost	Sale
D_1	0	4	0	8
D_2	0	2	0	0
D_3	4	0	2	4
D_4	0	6	4	6
D_5	0	4	6	4
D_6	6	0	4	0
D_7	0	0	0	0
D_8	4	2	0	2

Q_1 = QUERY$_1$ = Computer$_{1.0}$ OR Program$_{.333}$

Q_2 = QUERY$_2$ = Cost$_{0.75}$ AND Sale$_{1.0}$

Q_1 strict interpretation = $(D_1, D_2, D_3, D_4, D_5, D_6, D_8)$

Q_2 strict interpretation = (D_3, D_4, D_5)

Q_1 invariant = (D_8)

Q_2 invariant = (D_3, D_4, D_5)

Q_1 optional = $(D_1, D_2, D_3, D_4, D_5, D_6) \Rightarrow \lceil .333 \text{ times } 6 \text{ items} \rceil$

= 2 items

Q_2 optional = $(D_1, D_8) \Rightarrow \lceil (1 - 0.75) \text{ times } 2 \text{ items} \rceil$

= 1 item.

Here, QUERY$_1$ ends up with a set containing all of the items that contain the term "Computer" and two items from the set "computer" NOT "program". The symbol $\lceil \rceil$ stands for rounding up to the next integer.

In QUERY$_2$ the final set contains all of set "cost" AND "sale" plus 0.25 of the set of "sale" NOT "cost".

Now, using the similarity measure

$S(\text{Item}_i, \text{Item}_j) = \Sigma (\text{Term}_{i,k})(\text{Term}_{j,k})$

Give the following set of values based upon the centroids:

Centroid (Q_1) = $(D8)$ = $(4, 2, 0, 2)$

Centroid (Q_2) = $(D3, D4, D5)$

$= \frac{1}{3}(4+0+0, 0+6+4, 2+4+6, 4+6+$

Now, apply similarity,

$S(\text{centroid } Q_1, D_1) = (4\times2 + 2\times4 + 0\wedge0 + 2\times8)$

$= 24$

$S(CQ_1, D_2) = 0+4+0+0 = 4$

$S(CQ_1, D3) = 24$

$S(CQ_1, D4) = 24$

$S(CQ_1, D_5) = 16$

$S(CQ_1, D_6) = 24$

$S(CQ_2, D_1) = \frac{1}{3}(0+40+0+112)$

$= \frac{1}{3}(152)$

$S(CQ_2, D_8) = \frac{1}{3}(16+20+0+28)$

$= \frac{1}{3}(64)$

Now, For Q_1, two additional items are added to the invariant set $(D8) \cup (D_1, D3)$ by choosing the lowest no items because of the tie at 24, giving the answer of $(D_1, D3, D8$

For Q_2, one additional item is added to the invariant set $(D_3, D_4, D_5) \cup (D1)$ giving the answer $(D_1, D3, D4, D5)$.

(6) Searching the Internet and Hypertext :-

Some of the most commonly used nodes to search items on Internet are YAHOO, AltaVista and Lycos. The primary design decisions are on the level to which they retrieve data and their general philosophy on user access.

Lycos and Alta Vista automatically go out to other Internet sites and return the text at the site for automatic indexing.

Lycos returns home pages from each site for automatic indexing while Alta Vista indexes all of the text at a site.

The search process is directly available to the users via Intelligent Agents (IA).

IA provide the capability for a user to specify an info need which will be used by the IA as it independently moves b/w Internet sites locating info of interest.

There are 6 characteristics of IA :-

(i) Autonomy :- the search agent must be able to operate without interaction with a human agent. It must have ctrl over its own internal states and make independent decisions.

(ii) Communications Ability :- the agent must be able to communicate with the info sites as it traverses them. This implies a universal accepted lang defining the external interfaces.

(iii) Capacity for cooperation:- this concept suggests that intelligent agents need to cooperate to perform mutually beneficial tasks.

(iv) Capacity for Reasoning:

* Rule based – where user has defined a set of conditions and actions to be taken.
* Knowledge-based - where the IA have stored previous conditions & actions taken which are used to deduce for future actions.
* Artificial evolution based: where IA spawn new agents with higher logic capability to perform its objectives.

(v) Adaptive Behavior: closely tied to I and A, adaptive behavior permits the intelligent agent to assess its current state and make decisions on the actions it should take.

(vi) Trustworthiness - the user must trust that the intelligent agent will act on the user's behalf to locate info that the user has access to and is relevant to the user.

→ A Hyperlink is an embedded link to another item that can be instantiated by clicking on the item reference.