

UNIT-IV

User Search Techniques: Search statements and binding, Similarity measures and ranking, Relevance feedback, Selective dissemination of information search, weighted searches of Boolean systems, Searching the Internet and hypertext. **Information Visualization:** Introduction, Cognition and perception, Information visualization technologies.

Search Statements and Binding

Search statements are the statements of an information need generated by users to specify the concepts they are trying to locate in items.

In generation of the search statement, the user may have the ability to weight (assign an importance) to different concepts in the statement. At this point the binding is to the vocabulary and past experiences of the user. Binding in this sense 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 information needed by the user.

The next level of binding comes when the search statement is parsed for use by a specific search system.

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 and concept indexing systems.

Figure 7.1 illustrates the three potential different levels of binding. Parenthesis are used in the second binding step to indicate expansion by a thesaurus.

INPUT	Binding
"Find me information on the impact of the oil spills in Alaska on the price of oil"	User search statement using vocabulary of user
impact, oil (petroleum), spills (accidents), Alaska, price (cost, value)	Statistical system binding extracts processing tokens
impact (.308), oil (.606), petroleum (.65), spills (.12), accidents (.23), Alaska (.45), price (.16), cost (.25), value (.10)	Weights assigned to search terms based upon inverse document frequency algorithm and database

Figure 7.1 Examples of Query Binding

Similarity Measures and Ranking

A variety of different similarity measures can be used to calculate the similarity between the item and the search statement. 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. An example of a simple "sum of the products" similarity measure from the examples in Chapter 6 to determine the similarity between documents for clustering purposes is:

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

This formula uses the summation of the product of the various terms of two items when treating the index as a vector. If is replaced with then the same formula generates the similarity between every Item and The problem with this simple measure is in the normalization needed to account for variances in the length of items. Additional normalization is also used to have the final results come between zero and +1 (some formulas use the range -1 to +1)

This assumption of the availability of relevance information in the weighting process was later relaxed by Croft and Harper (Croft-79). Croft expanded this original concept, taking into account the frequency of occurrence of terms within an item producing the following similarity formula (Croft-83):

$$\text{SIM}(\text{DOC}_i, \text{QUERY}_j) = \sum_{i=1}^Q (C + \text{IDF}_i) * f_{i,j}$$

where C is a constant used in tuning, IDF_i is the inverse document frequency for term "i" in the collection and

$$f_{i,j} = K + (K - 1) \text{TF}_{i,j} / \text{maxfreq}_j$$

where K is a tuning constant, is the frequency of "i" and is the maximum frequency of any term in item "j." The best values for K seemed to range between 0.3 and 0.5. Another early similarity formula was used by Salton in the SMART system (Salton-83). Salton treated the index and the search query as n -dimensional vectors (see Chapter 5). To determine the "weight" an item has with respect to the search statement, the Cosine formula is used to calculate the distance between the vector for the item and the vector for the query:

$$\text{SIM}(\text{DOC}_i, \text{QUERY}_j) = \frac{\sum_{k=1}^n (\text{DOC}_{i,k} * \text{QTERM}_{j,k})}{\sqrt{\sum_{k=1}^n (\text{DOC}_{i,k})^2 * \sum_{k=1}^n (\text{QTERM}_{j,k})^2}}$$

where is the k th term in the weighted vector for Item "i" and is the k th term in query "j." The Cosine formula calculates the Cosine of the angle between the two vectors. As the Cosine approaches "1," the two vectors become coincident (i.e., the term and the query represent the same concept). If the two are totally unrelated, then they will be orthogonal and the value of the Cosine is "0." What is not taken into account is the length of the vectors

For example, if the following vectors are in a three dimensional (three term) system:

Item = (4, 8, 0)

Query 1 = (1, 2, 0)

Query 2 = (3, 6, 0)

$$\text{QTERM}_{i,k} = (0.5 + (0.5 \text{TF}_{i,k} / \text{maxfreq}_k)) * \text{IDF}_i$$

$$\text{SIM}(\text{DOC}_i, \text{QUERY}_j) = \frac{\sum_{k=1}^n (\text{DOC}_{i,k} * \text{QTERM}_{j,k})}{\sum_{k=1}^n \text{DOC}_{i,k} + \sum_{k=1}^n \text{QTERM}_{j,k} - \sum_{k=1}^n (\text{DOC}_{i,k} * \text{QTERM}_{j,k})}$$

The Dice measure simplifies the denominator from the Jaccard measure and introduces a factor of 2 in the numerator. The normalization in the Dice formula is also invariant to the number of terms in common.

$$\text{SIM}(\text{DOC}_i, \text{QUERY}_j) = \frac{2 * \sum_{k=1}^n (\text{DOC}_{i,k} * \text{QTERM}_{j,k})}{\sum_{k=1}^n \text{DOC}_{i,k} + \sum_{k=1}^n \text{QTERM}_{j,k}}$$

QUERY = (2, 2, 0, 0, 4)
 DOC1 = (0, 2, 6, 4, 0)
 DOC2 = (2, 6, 0, 0, 4)

	Cosine	Jaccard	Dice
DOC1	36.66	16	20
DOC2	36.66	-12	20

Figure 7.2 Normalizing Factors for Similarity Measures

Figure 7.3 illustrates the threshold process. The simple “sum of the products” similarity formula is used to calculate similarity between the query and each document. If no threshold is specified, all three documents are considered hits. If a threshold of 4 is selected, then only DOC1 is returned.

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

DOC1: geography of Mexico suggests oil reserves are available
 vector (0, 1, 0, 2, 0, 3, 1, 0)

DOC2: American geography has lakes available everywhere
 vector (1, 3, 2, 0, 0, 0, 0, 0)

DOC3: painters suggest Mexico lakes as subjects
 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

Figure 7.3 Query Threshold Process

One special area of concern arises from search of clusters of terms that are stored in a hierarchical scheme (see Chapter 6). The items are stored in clusters that are represented by the centroid for each cluster. Figure 7.4 shows a cluster representation of an item space. In Figure 7.4, each letter at the leaf (bottom nodes) represent an item (i.e., K, L, M, N, D, E, F, G, H, P, Q, R, J). 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, then that part of the tree is pruned and not searched. This continues until

the actual leaf nodes that are not pruned are compared. The problem comes from the nature of a centroid which is an average of a collection of items (in Physics, the center of gravity). The risk is that the average may not be similar enough to the query for continued search, but specific items used to calculate the centroid may be close enough to satisfy the search. The risks of missing items and thus reducing recall increases as the standard deviation increases. 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 (item) level.

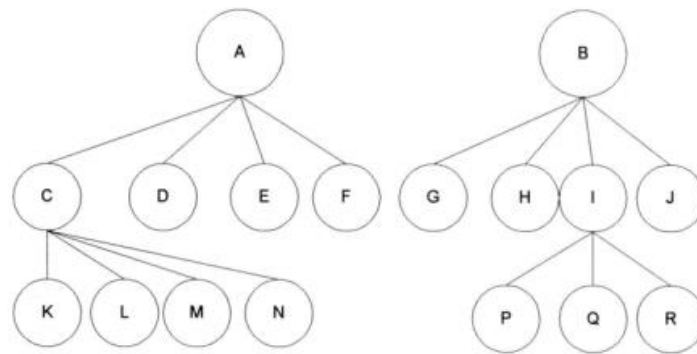


Figure 7.4 Item Cluster Hierarchy

Hidden Markov Models Techniques

Use of Hidden Markov Models for searching textual corpora has introduced a new paradigm for search. In most of the previous search techniques, the query is thought of as another "document" and the system tries to find other documents similar to it. In HMMs the documents are considered unknown statistical processes that can generate output that is equivalent to the set of queries that would consider the document relevant. Another way to look at it is by taking the general definition that a HMM is defined by output that is produced by passing some unknown key via state transitions through a noisy channel. The observed output is the query, and the unknown keys are the relevant documents. The noisy channel is the mismatch between the author's way of expressing ideas and the user's ability to specify his query. Leek, Miller and Schwartz (Leek-99) computed for each document the probability that D was the relevant document in the users mind given that Q was the query produced, i.e., $P(D \text{ is } R/Q)$. The development for a HMM approach begins with applying Bayes rule to the conditional probability

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

The biggest problem in using this approach is to estimate the transition probability matrix and the output (queries that could cause hits) for every document in the corpus.

Ranking Algorithms

A by-product of use of similarity measures for selecting Hit items is a value that can be used in ranking the output. Ranking the output implies ordering the output 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. The original Boolean systems returned items ordered by date of entry into the system versus by likelihood of relevance to the user’s search statement. With the inclusion of statistical similarity techniques into commercial systems and the large number of hits that originate from searching diverse corpora, such as the Internet, ranking has become a common feature of modern systems. A summary of ranking algorithms from the research community is found in an article written by Belkin and Croft (Belkin-87).

Relevance Feedback

The first major work on relevance feedback was published in 1965 by Rocchio (republished in 1971: Rocchio-71). Rocchio was documenting experiments on reweighting query terms and query expansion based upon a vector representation of queries and items. The concepts are also found in the probabilistic model presented by Robertson and Sparck Jones (Robertson-76). The relevance feedback concept was that the new query should be based on the old query modified to increase the weight of

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

where

- Q_n = the revised vector for the new query
- Q_o = the original query
- r = number of relevant items
- DR_i = the vectors for the relevant items
- nr = number of non-relevant items
- DNR_j = the vectors for the non-relevant items.

The factors r and nr were later modified to be constants that account for the number of items along with the importance of that particular factor in the equation. Additionally a constant was added to Q_o to allow adjustments to the importance of the weight assigned to the original query. This led to the revised version of the formula:

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

Terms in relevant items and decrease the weight of terms that are in non-relevant items. This technique not only modified the terms in the original query but also allowed expansion of new terms from the relevant items. The formula used is:

Where α and β are the constants associated with each factor (usually $1/n$ or $1/nr$ times a constant). The factor is referred to as positive feedback because it is using the user judgments on relevant items to increase the values of terms for the next iteration of searching. The factor γ is referred to as negative feedback. Figure 7.6 gives an example of the impacts of positive and negative feedback. The filled circles represent non-relevant items; the other circles represent relevant items. The oval represents the items that are returned from the query. The solid box is logically where the query is initially. The hollow box is the query modified by relevance feedback (positive only or negative only in the Figure).

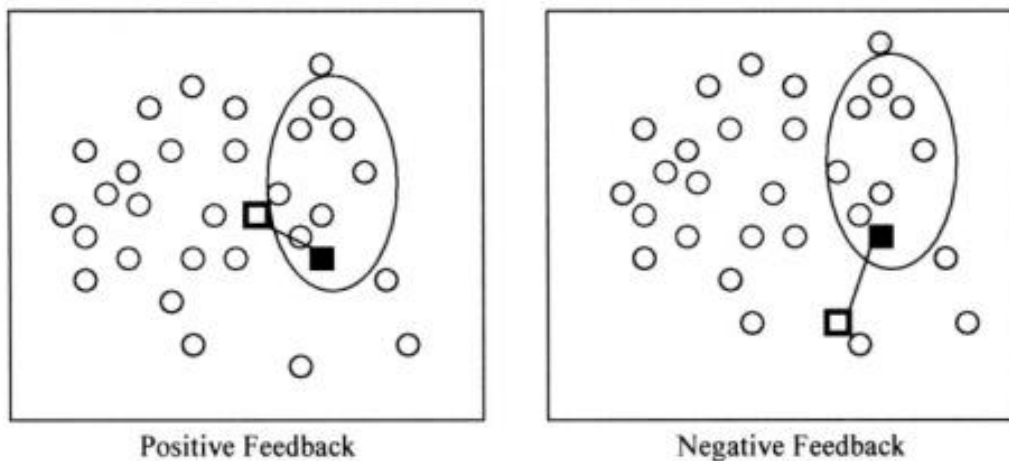


Figure 7.6 Impact of Relevance Feedback

Relevance feedback, in particular positive feedback, has been proven to be of significant value in producing better queries. Some of the early experiments on the SMART system (Ide-69, Ide-71, Salton-83) indicated the possible improvements that would be gained by the process. But the small collection sizes and evaluation techniques put into question the actual gains by using relevance feedback. One of the early problems addressed in relevance feedback is how to treat query terms that are not found in any retrieved relevant items. Just applying the algorithm would have the effect of reducing the relative weight of those terms with respect to other query terms. From the user's perspective, this may not be desired because the term may still have significant value to the user if found in the future iterations of the search process. Harper and van Rijisbergen addressed this issue in their proposed EMIM weighting scheme (Harper-78, Harper-80). Relevance feedback has become a common feature in most information systems. 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 negative feedback would have eliminated them. In some systems the modified query is presented to the user to allow the user to readjust the weights and review the new terms added.

Selective Dissemination of Information Search

Selective Dissemination of Information, frequently called dissemination systems, are becoming more prevalent with the growth of the Internet. A dissemination system is sometimes labeled a “push” system while a search system is called a “pull” system. The differences are that in a search system the user proactively makes a decision that he needs information and directs the query to the information system to search. In a dissemination system, the user defines a profile (similar to a stored query) and as new information is added to the system it is automatically compared to the user's profile.

Weighted Searches of Boolean Systems

The two major approaches to generating queries are Boolean and natural language. Natural language 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 systems. Some of the issues are associated with how the logic (AND, OR, NOT) operators function with weighted values and how weights are associated with the query terms. If the operators are interpreted in their normal interpretation, they act too restrictive or too general (i.e., AND and OR operators respectively). Salton, Fox and Wu showed that using the strict definition of the operators will suboptimize the retrieval expected by the user (Salton-83a). Closely related to the strict definition

problem is the lack of ranking that is missing from a pure Boolean process. Some of the early work addressing this problem recognized the fuzziness associated with mixing Boolean and weighted systems (Brookstein-78, Brookstein-80) To integrate the Boolean and weighted systems model, Fox and Sharat proposed a fuzzy set approach (Fox-86). Fuzzy sets introduce the concept of degree of membership to a set (Zadeh-65). The degree of membership for AND and OR operations are defined as:

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

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

$$SIM(Q_{OR}, DOC) = C_{OR1} * \max(DOC_1, DOC_2, \dots, DOC_n) + C_{OR2} * \min(DOC_1, DOC_2, \dots, DOC_n)$$

$$SIM(Q_{AND}, DOC) = C_{AND1} * \min(DOC_1, DOC_2, \dots, DOC_n) + C_{AND2} * \max(DOC_1, DOC_2, \dots, DOC_n)$$

The MMM technique was expanded by Paice (Paice-84) considering all item weights versus the maximum/minimum approach. The similarity measure is calculated as:

$$SIM(Q_{OR}, DOC) = \frac{\sum_{i=1}^n r^{i-1} d_i}{\sum_{i=1}^n r^{i-1}}$$

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

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

Searching the INTERNET and Hypertext

The Internet has multiple different mechanisms that are the basis for search of items. The primary techniques are associated with servers on the Internet that create indexes of items on the Internet and allow search of them. Some of the most commonly used nodes are YAHOO, AltaVista and Lycos. In all of these systems there are active processes that visit a large number of Internet sites and retrieve textual data which they index. The primary design decisions are on the level to which they retrieve data and their general philosophy on user access. LYCOS (<http://www.lycos.com>) and AltaVista automatically go out to other Internet sites and return the text at the sites for automatic indexing (<http://www.altavista.digital.com>). Lycos returns home pages from each site for automatic indexing while Altavista indexes all of the text at a site. The retrieved text is then used to create an index to the source items storing the Universal Resource Locator (URL) to provide to the user to retrieve an item. All of the systems use some form of ranking algorithm to assist in display of the retrieved items. The algorithm is kept relatively simple using statistical information on the occurrence of words within the retrieved text

Closely associated with the creation of the indexes is the technique for accessing nodes on the Internet to locate text to be indexed. This search process is also directly available to users via Intelligent Agents. Intelligent Agents provide the capability for a user to specify an information

need which will be used by the Intelligent Agent as it independently moves between Internet sites locating information of interest. There are six key characteristics of intelligent agents (Heilmann-96):

1. **Autonomy** - the search agent must be able to operate without interaction with a human agent. It must have control over its own internal states and make independent decisions. This implies a search capability to traverse information sites based upon pre-established criteria collecting potentially relevant information.
2. **Communications Ability** - the agent must be able to communicate with the information sites as it traverses them. This implies a universally accepted language defining the external interfaces (e.g., Z39.50).
3. **Capacity for Cooperation** - this concept suggests that intelligent agents need to cooperate to perform mutually beneficial tasks.
4. **Capacity for Reasoning** - There are three types of reasoning scenarios (Roseler-94): Rule-based - where user has defined a set of conditions and actions to be taken Knowledge-based - where the intelligent agents have stored previous conditions and actions taken which are used to deduce future actions Artificial evolution based - where intelligent agents spawn new agents with higher logic capability to perform its objectives.
5. **Adaptive Behavior** - closely tied to 1 and 4 , adaptive behavior permits the intelligent agent to assess its current state and make decisions on the actions it should take
6. **Trustworthiness** - the user must trust that the intelligent agent will act on the user's behalf to locate information that the user has access to and is relevant to the user.

Information Visualization

Functions that are available with electronic display and visualization of data that were not previously provided are:

- modify representations of data and information or the display condition (e.g., changing color scales)
- use the same representation while showing changes in data (e.g., moving between clusters of items showing new linkages)
- animate the display to show changes in space and time
- Create hyperlinks under user control to establish relationships between data

Information Visualization addresses how the results of a search may be optimally displayed to the users to facilitate their understanding of what the search has provided and their selection of most likely items of interest to read. Cognitive (the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses) engineering derives design principles for visualization techniques from what we know about the neural processes involved

with attention, memory, imagery and information processing of the human visual system.

Cognitive engineering results can be applied to methods of reviewing the concepts contained in items selected by search of an information system. Visualization can be divided into two broad classes: link visualization and attribute (concept) visualization. Link visualization displays relationships among items. Attribute visualization reveals content relationships across large numbers of items.

There are many areas that information visualization and presentation can help the user:

- a. reduce the amount of time to understand the results of a search and likely clusters of relevant information
- b. yield information that comes from the relationships between items versus treating each item as independent
- c. perform simple actions that produce sophisticated information search functions

Visualization is the transformation of information into a visual form which enables the user to observe and understand the information.

Cognition (the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses)

Perception (the ability to see, hear, or become aware of something through the senses)

Proximity - nearby figures are grouped together

Similarity - similar figures are grouped together

Continuity - figures are interpreted as smooth continuous patterns rather than discontinuous concatenations of shapes (e.g., a circle with its diameter drawn is perceived as two continuous shapes, a circle and a line, versus two half circles concatenated together)

Closure - gaps within a figure are filled in to create a whole (e.g., using dashed lines to represent a square does not prevent understanding it as a square)

Connectedness - uniform and linked spots, lines or areas are perceived as a single unit

Aspects of the Visualization Process

One of the first-level cognitive processes is pre attention, that is, taking the significant visual information from the photoreceptors and forming primitives. In Figure 8.1 the visual system detects the difference in orientations between the left and middle portion of the figure and determines the logical border between them. An example of using the conscious processing capabilities of the brain is the detection of the different shaped objects and the border between them shown between the left side and middle of the Figure 8.1. The reader can likely detect the differences in the time it takes to visualize the two different boundaries.

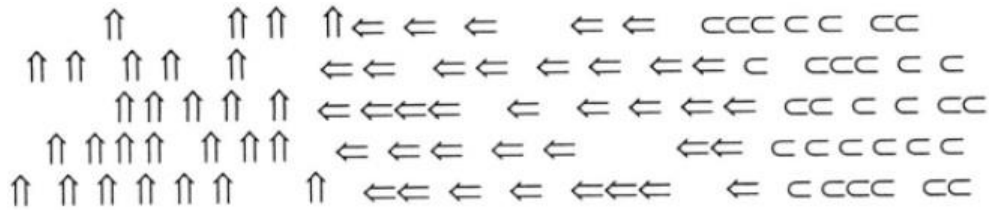


Figure 8.1 Preattentive Detection Mechanism

The preattentive process can detect the boundaries between orientation groups of the same object. A harder process is to identify the equivalence of rotated objects. For example, a rotated square requires more effort to recognize it as a square. As we migrate into characters, the problem of identification of the character is affected by rotating the character in a direction not normally encountered. It is easier to detect the symmetry when the axis is vertical. Figure 8.2 demonstrates these effects.

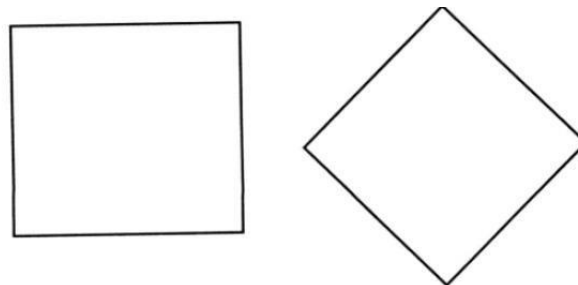


Figure 8.2 Rotating a Square and Reversing Letters in "REAL"

Color is one of the most frequently used visualization techniques to organize, classify, and enhance features.

The goals for displaying the result from searches fall into two major classes: document clustering and search statement analysis. The goal of document clustering is to present the user with a visual representation of the document space constrained by the search criteria. Within this constrained space there exist clusters of documents defined by the document content. Visualization tools in this area attempt to display the clusters, with an indication of their size and topic, as a basis for users to navigate to items of interest.

The second goal is to assist the user in understanding why items were retrieved, thereby providing information needed to refine the query. Visualization techniques approach this problem by displaying the total set of terms, including additional terms from relevance feedback or thesaurus expansion, along with documents retrieved and indicate the importance of the term to the retrieval and ranking process.

Link analysis is also important because it provides aggregate-level information within an information system. One way of organizing information is hierarchical. A two-dimensional representation becomes difficult for a user to understand as

The hierarchy becomes large. One of the earliest experiments in information visualization was the Information Visualizer developed by XEROX PARC. It incorporates various visualization formats such as DataMap,

InfoGrid, ConeTree, and the Perspective wall. The Cone-Tree is a 3-Dimensional representation of data, where one node of the tree is represented at the apex and all the information subordinate to it is arranged in a circular structure at its base.

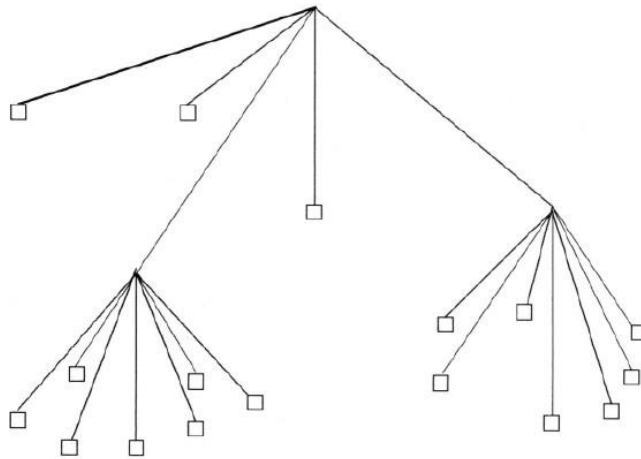


Figure 8.4 Cone Tree

Thus a six-dimensional coordinate space may have three of the coordinates defined as a subspace within the other three coordinate spaces. This has been called Feiner's "worlds within worlds"

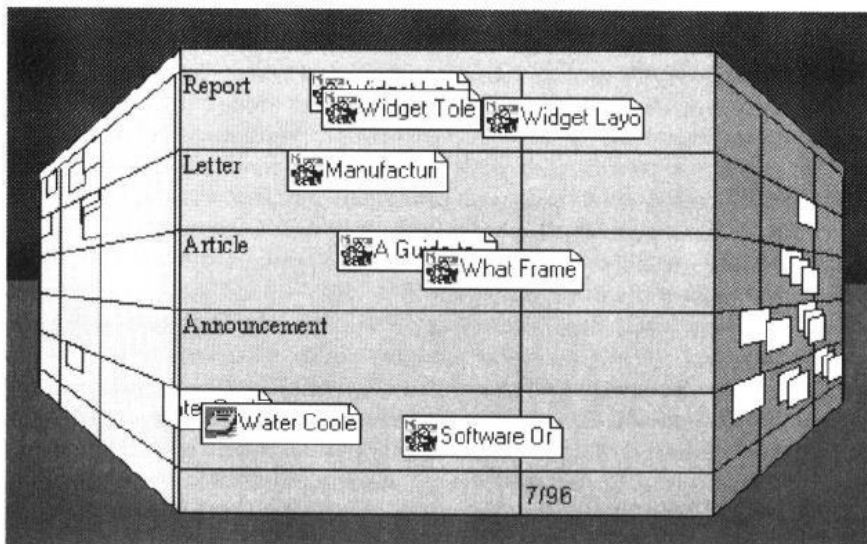


Figure 8.5 Perspective Wall
From inXight web site - www.inxight.com

approach.

Another clustering system that uses statistical information for a small number of items (50 - 120) to show term relationships via spatial positioning is the VIBE system . The VIBE system allows users to associate query terms with different locations in the visual display.

Lin has taken the self-organization concept further by using Kohonen's algorithm to automatically determine a table of contents (TOC) and display the results in a map display. Visualization tools need to assist the user in understanding the effects of his search statement even to the level of identifying important terms that are not contributing to the search process. One solution is a graphical display of the characteristics of the retrieved items which contributed to their selection. This is effected in the Envision system.

Figure 8.7 shows Envision's three interactive windows to display search results: Query window, Graphic View window, and Item Summary window. The Query window provides an editable version of the query. The Item Summary window provides bibliographic citation information on items selected in the Graphic View window. The Graphic View window is similar to scatterplot graphs. Each item in the Hit file is represented by an icon in the window. Selecting an item in the window provides bibliographic information on the same display. Circles represent single items with the relevance weights displayed below them.

Ellipses represent clusters of multiple items that are located at the same point in the scatterplot with the number of items in the ellipse and their weights below the ellipse. In this example, estimated relevance is on the X-axis and author's name is on the Y-axis. This type of interface provides a very user friendly environment but encounters problems when the number of relevant items and entries for an axis becomes very large. Envision plans to address this issue by a "zoom" feature that will allow seeing larger areas of the scatterplot at lesser detail.

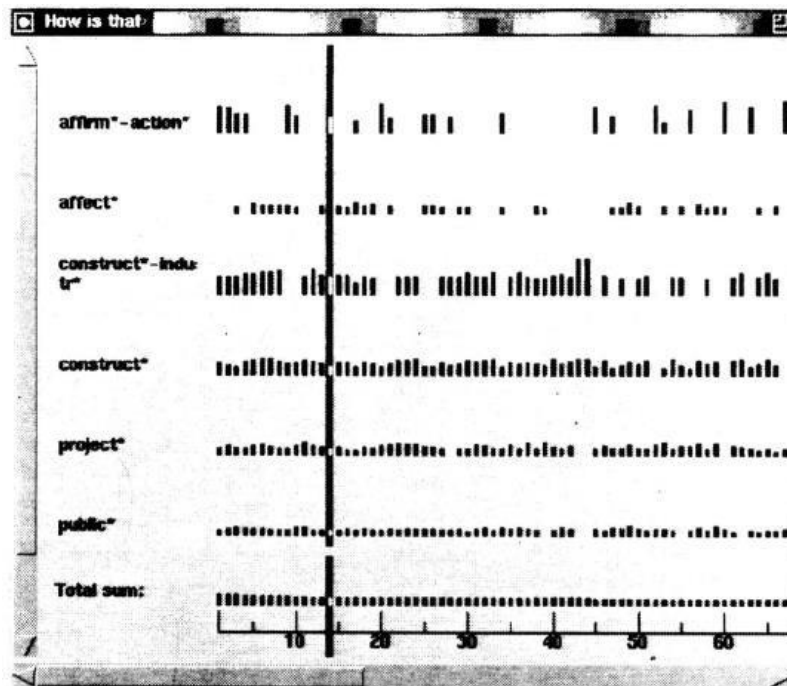


Figure 8.8 Visualization of Results
(from SIGIR 96, page 88)

A similar technique is used by Veerasamy and Belkin (Veerasamy-96). They use a series of vertical columns of bars. The columns of bars represent documents, and the rows represent index terms. The height of the bar corresponds to the weight of the corresponding term (row) in the corresponding item (column). In addition to the query terms, the system shows the additional words added to the system by relevance feedback. Figure 8.8 provides an example for a search statement of “How affirmative action affected the construction industry.” This approach quickly allows a user to determine which terms had the most effect on retrieving a specific item (i.e. by scanning down the column). It also allows the user to determine how the various terms contributed to the retrieval process (i.e. by scanning a row).

This latter process is very important because it allows a user to determine if what he considers to be an important search term is not contributing strongly or not found at all in the items being retrieved. It also shows search terms that are causing items to be retrieved allowing their removal or reduction in query weight if they are causing false hits. In the Boolean environment this function was accomplished by vocabulary browsing (see Chapter 2) that allows for a user to see the number of items a particular term is in prior to including it in a search.

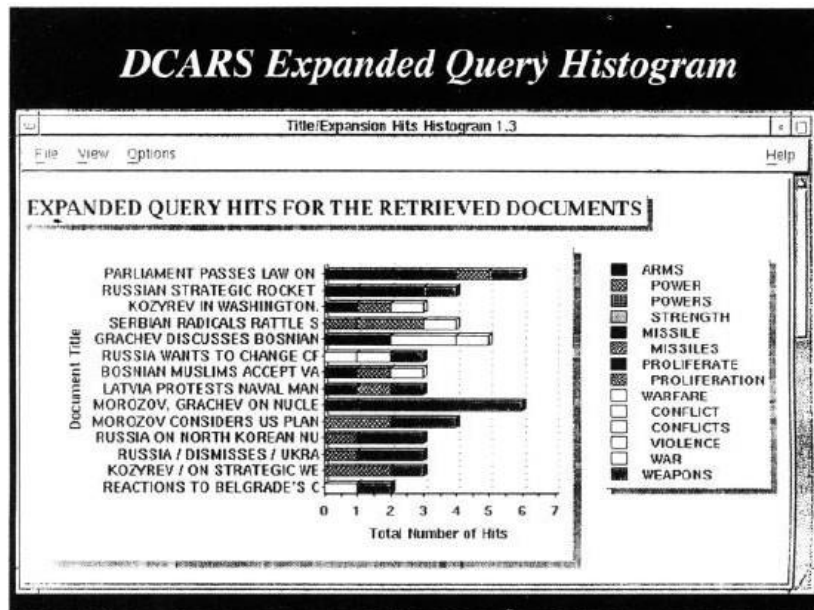


Figure 8.9 Example of DCARS Query Histogram
(from briefing by CALSPAN)

A slightly different commercial version having properties similar to the systems above is the Document Content Analysis and Retrieval System (DCARS) being developed by Calspan Advanced Technology Center. Their system is designed to augment the RetrievalWare search product. They display the query results as a histogram with the items as rows and each term's contribution to the selection indicated by the width of a tile bar on the row (see Figure 8.9). DCARS provides a friendly user interface that indicates why a particular item was found, but it is much harder to use the information in determining how to modify search statements to improve them.

Another representation that is widely used for both hierarchical and network related information is the "cityscape" which uses the metaphor of movement within a city. In lieu of using hills, as in the terrain approach, skyscrapers represent the theme (concept) area as shown in Figure 8.10. This is similar to extending bar charts to three dimensions. Buildings can be connected by lines which can vary in representation to describe interrelationships between themes. Colors or fill designs can be used for the visualization presenting another layer of information (e.g., the building having the same color may be members of a higher concept). Movement within the cityscape (or terrain) of the viewer perspective allows zooming in on specific information areas that will bring into view additional structures that might have been hidden by the previous viewpoint.

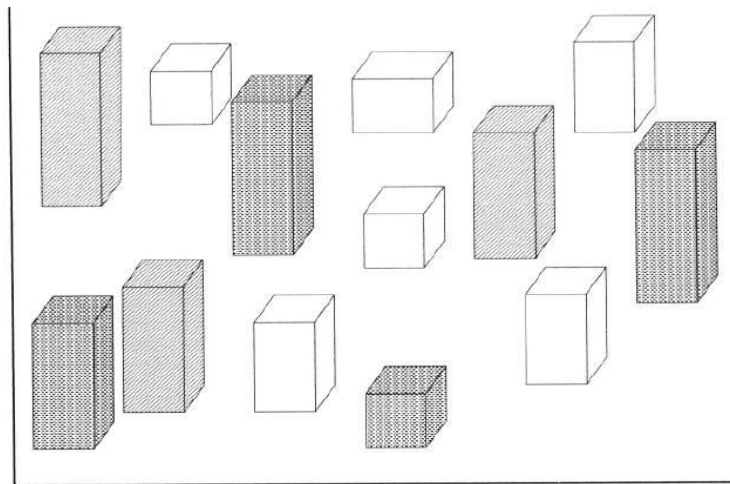


Figure 8.10 CityScape Example

Another task in information systems is the visualization of specific text within an item versus between items. In some situations, items are allowed to change over time via editing. Thus, there is both the static representation and a time varying representation. Text changing representations are very important when the text being represented is a software program of millions of lines of code.

AT&T Bell laboratories created the SeeSoft system which uses columns and color codes to show when different lines of code have been changed. This technique was used as a basis for a similar code visualization tool, DEC FUSE/SoftVis (Zaremba-95). They created small pictures of files that represent the code in the file with the size of the picture scaled to the number of lines of code in the file. Color coding indicates different characteristics of the code (e.g., green is comments). A user can quickly see the relative structure of all of the code files composing a system along with the complexity of each of the modules. The TileBars tool from Xerox PARC provides the user with a visualization of the distribution of query terms within each item in a Hit file. Using this tool, the user can quickly locate the section of the item that is most likely to be of interest.

The first visualization tool was the Query By Example user interface developed by IBM (Zloof-75). The interface presented the user with two dimensional tables on the display screen and based upon the user is defining values of interest on the tables, the system would complete the search. The Information Visualization and Exploration Environment (IVEE) makes use of the three dimensional representation of the structured database as constrained by

the user's search statement. MITRE Corporation has developed a tool used with web browsers that enables a user to see a tree structure visual representation of the information space they have navigated through.

Another area in information visualization is the representation of pattern and linkage analysis. A system that incorporates many information visualization techniques including those used to represent linkage analysis is the Pathfinder Project sponsored by the Army (Rose-96). It contains the Document Browser, CAMEO, Counts, CrossField Matrix, OILSTOCK and SPIRE tools. The Document Browser uses different colors and their density for words in the text of items to indicate the relative importance of the item to their profile of interest.

CAMEO models an analytic process by creating nodes and links to represent a problem. Queries are associated with the nodes. The color of the nodes change based on how well the found items satisfy the query. Counts uses statistical information on words and phrases and plots them over time. Time is used as a parameter to show trends in development of events. The display uses a three dimensional cityscape representation of the data. The Cross Field Matrix creates a two-dimensional matrix of two fields in a dataset.

The SPIRE tool is a type of scattergraph of information. Items are clustered and displayed in a star chart configuration. Distance between two points is indicative of their similarity based upon concurrence of terms.