Unit 4

Search Statements and Binding

- Search statements are the statements of an information and generated by users to specify the concepts they are trying to locate in items.
- It uses traditional Boolean logic and/or Natural Language.
- While generating the search statement, the user have the ability to weight or 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 here it means 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 search system translates the query to its own meta language.

- This process is similar to the indexing of item processes.
- For example, statistical systems determine the processing tokens of interest and the weights assigned to each processing token based upon frequency of occurrence from the search statement.
- Natural language systems determine the syntactical and discourse semantics using algorithms similar to those used in indexing.
- Concept systems map the search statement to the set of concepts used to index items.

- 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.

- Some of the statistics used in weighting are based upon the current contents of the database.
- Some examples are Document Frequency and Total Frequency for a specific term.
- Frequently in a concept indexing system, the concepts that are used as the basis for indexing are determined by applying a statistical algorithm against a representative sample of the database versus being generic across all databases.

- Natural Language indexing techniques tend to use the most corpora-independent algorithms.

- 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

- The length of search statements directly affect the ability of Information Retrieval Systems to find relevant items.
- The longer the search query, the easier it is for the system to find items.
- Profiles used as search statements for Selective Dissemination of Information systems are usually very long, typically 75 to 100 terms.
- In large systems used by research specialists and analysts, the typical adhoc search statement is approximately 7 terms.

Similarity Measures and Ranking

- Searching is concerned with calculating the similarity between a user's search statement and the items in the database.
- Restricting the similarity measure to passages gains significant precision with minimal impact on recall.
- Once items are identified as possibly relevant to the user's query, it is best to present the most likely relevant items first- Ranking is a scalar number that represents how similar an item is to the query.

Similarity measure

- 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 to determine the similarity between documents for clustering purposes is:
- SIM(Itemi, Itemj) = Z(Termix) (Termja)

- This formula uses the summation of the product of the various terms of two items when treating the index as a vector.
- If Itemj is replaced with Queryj then the same formula generates the similarity between every Item and Queryj.
- 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).

Similarity formula by Salton in SMART system

- 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:

- $$SIM(DOC_i, QUERY_j) = \frac{\sum_{k=1}^{n} (DOC_{i,k} * QTERM_{j,k})}{\sqrt{\sum_{k=1}^{n} (DOC_{i,k})^2 * \sum_{k=1}^{n} (QTERM_{j,k})^2}}$$

Similarity formula by Salton in SMART system

- **The Jaccard formula is:**
- The denominator be varies depend upon the no of terms in common.
- Common elements common increase, the similarity value quickly decreases, in the range -1 and+1.

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

Similarity formula by Salton in SMART system

- **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.

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

Similarity formula by Salton in SMART system

- Use of a similarity algorithm returns the complete data base as search results.
- Many of the items have a similarity close or equal to zero.
- Thresholds(default is the similarity > zero) are usually associated with the search process.
- The threshold defines the items in the resultant Hit file from the query.
- Thresholds are either a value that the similarity measure must equal or exceed or a number that limits the number of items in the Hit file.

Normalizing denominator results vary with commonality of terms

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

Query Threshold process

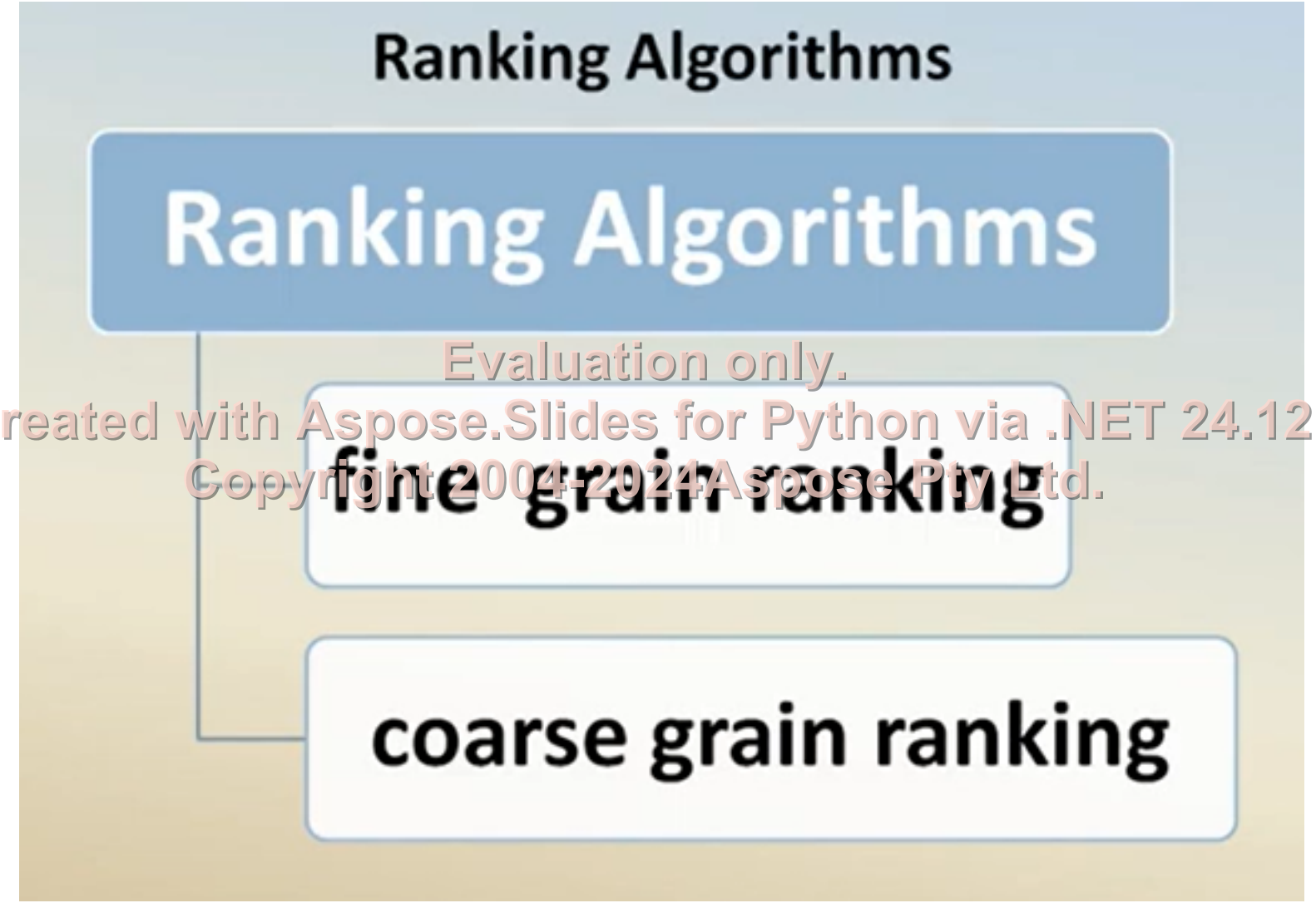
Vector:	American, geography, lake, Mexico, painter, oil, reserve, subject
DOC1	geography of Mexico *suggests* oil reserves *are available* vector (0, 1, 0, 2, 0, 3, 1, 0)
DOC2	American *geography* lakes *are available everywhere* vector (1, 3, 2, 0, 0, 0, 0, 0)
DOC3	painters *suggest* Mexico lakes *as* 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

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.



- In most of the commercial systems, heuristic rules a RetrievalWare first uses indexes (inversion lists) to identify potential relevant items.
- It then applies coarse grain and fine grain ranking.

fine grain ranking	coarse grain ranking
In the fine grain ranking, the exact rank of the item is calculated.	The coarse grain ranking is based on the presence of query terms within items.
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.	The coarse grain ranking is a weighted formula that can be adjusted based on completeness, contextual evidence or variety, and semantic distance.

- If the related terms and query terms occur in close proximity (same sentence or paragraph) the item is judged more relevant

- Ex: Computing computer
- They differ 3 units.

Relevance Feedback

- Thesuari and semantic networks provide utility in generally expanding a user's search statement to include potential related search terms.

- But this still does not correlate to the vocabulary used by the authors that contributes to a particular database.
- There is also a significant risk that the thesaurus does not include the latest jargon being used, acronyms or proper nouns.

Relevance Feedback

- In an interactive system, users can manually modify an inefficient query or have the system automatically expand the query via a thesaurus.

  Relevant items (or portions of relevant items) are used to reweight the existing query terms and possibly expand the user's search statement with new terms.

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

The formula used for Relevance Feedback

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

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.

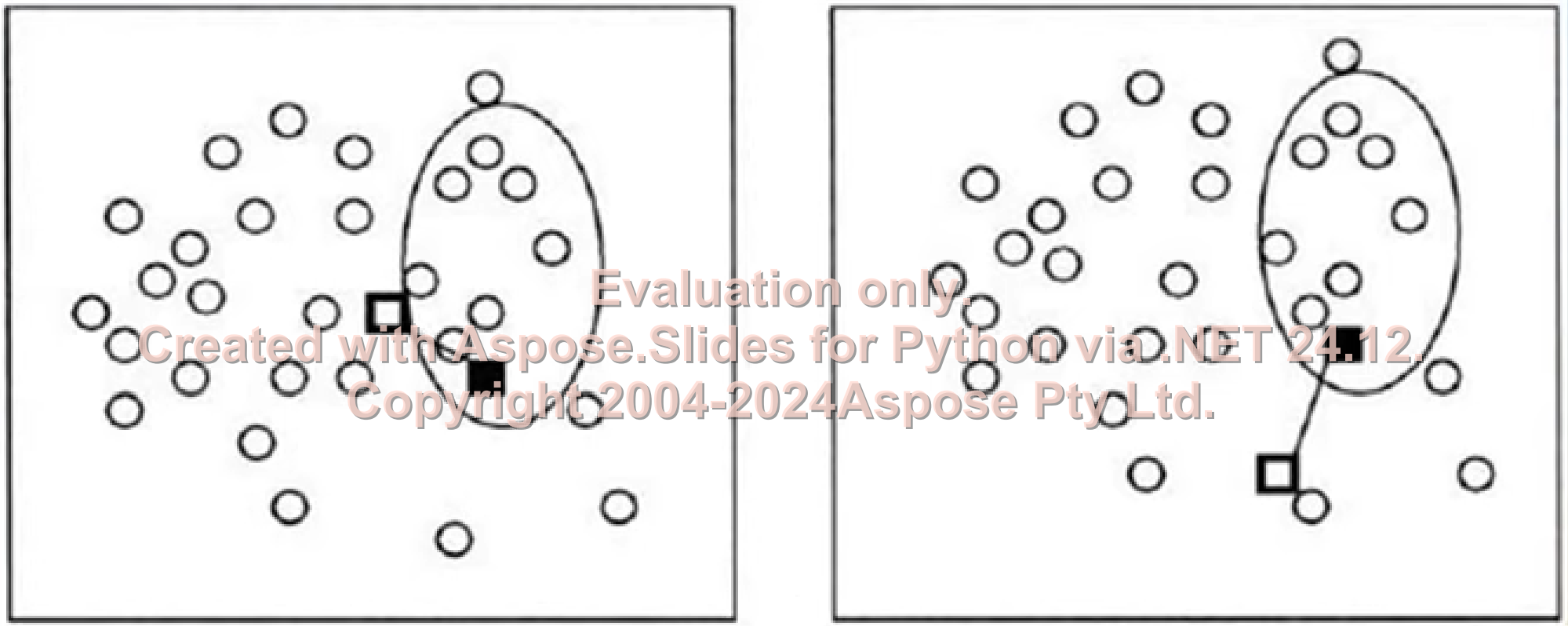


Figure 7.6 Impact of Relevance Feedback

	Term 1	Term 2	Term 3	Term 4	Term 5
Q_o	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	3¾	1¾	0	1¼	0

Figure 7.7 Query Modification via Relevance Feedback

	DOC1	DOC2	DOC3
Q_o	6	3	6
Q_n	14½	9.0	3.75

$$Q_n = (3, 0, 0, 2, 0) + ¼ (2+1, 4+3, 0+0, 0+0, 2+0) - ¼ ((0, 0, 4, 3, 2)$$

$$= (3¾, 1¾, 0\ \{-1\}, 1¼, 0)$$

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 passed a "**push**" system while a search system and while gives information called as "**pull**" system.

- In a dissemination system, the user defines a profile and as new info is added to the system it is automatically compared to the user's profile.

Selective Dissemination of Information Search

- If it is considered a match, it is asynchronously sent to the user's "**mail**"file.
- The differences between the two functions lie in the dynamic nature of the profiling process, the size and diversity of the search statements and number of simultaneous searches per item.
- In the search system, an existing database exists.

Selective Dissemination of Information Search

- These can be used for weighting factors in the indexing process and the similarity comparison (e.g., inverse document frequency algorithms).
- Profiles are relatively static search statements that cover a diversity of topics.

Selective Dissemination of Information Search

- One of the first commercial search techniques for dissemination was the Logic on Message Dissemination System (LMDS).
- The system originated from a system created by Chase, Rosen and Wallace (CRW Inc.).
- It was designed for speed to support the search of thousands of profiles with items arriving every 20seconds.
- Another approach to dissemination uses a statistical classification technique and explicit error minimization to determine the decision criteria for selecting items for a particular profile.

Selective Dissemination of Information Search

- Schutze et al. used two approaches to reduce the dimensionality:
- selecting a set of existing features to use or creating a new much smaller set of features that the original features are mapped into. χ^2 was used to determine the most important features.
- The test was applied to a table that contained the number of relevant(N_r)and non-relevant(N^n_r) items in which a term occurs plus the number of relevant and non-relevant items in which the term does not occur respectively).

Selective Dissemination of Information Search

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

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

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
- 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

- Autonomy
- Communications Ability
- Capacity for Cooperation
- Capacity for Reasoning
- Adaptive Behaviour
- Trustworthiness

THE TEXT RETRIEVAL TECHNIQUES

- Three classical text retrieval techniques have been defined for organizing items in a textual database, for rapidly identifying the relevant items
- The techniques are:
- Full text scanning(streaming)
- Word inversion
- Multi attribute retrieval
- Text streaming architecture

Software Text Search Algorithms

- In software streaming techniques, the item to be searched is read into memory and then the algorithm is applied.
- There are Three major algorithms associated with software text search:
- Brute force approach
- Boyer-Moore
- Knuth-Morris-Pratt
- Finite State Algorithm for Text Search

Software Text Search Algorithms

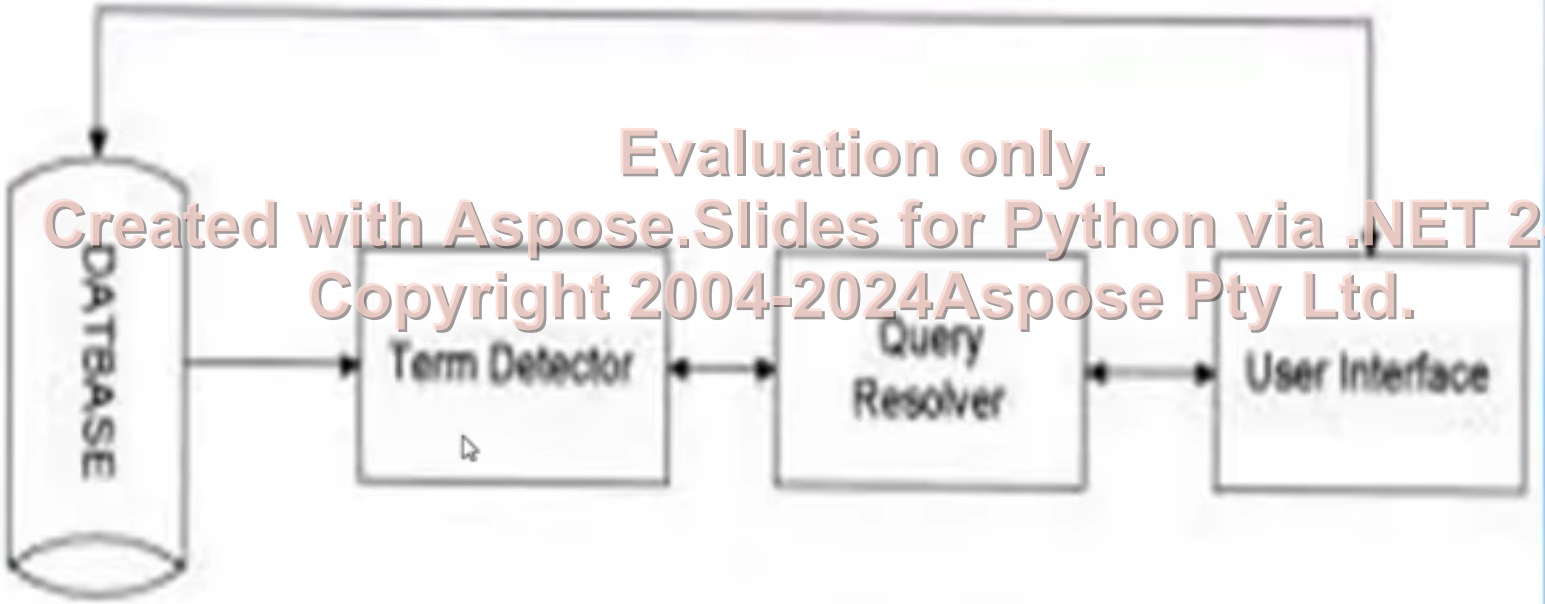


Figure 9.1 Text Streaming Architecture

Software Text Search Algorithms

- The basic concept of a text scanning system provides one or more users to enter queries, and the text to be searched is accessed and compared to the query terms.
- If all of the text has been accessed, the query is complete.
- One advantage of this type architecture is that as soon as an item is identified as satisfying a query, the results can be presented to the user for retrieval.

Software Text Search Algorithms

- There the database contains the full text of the items.
- The term detector is typically a special hardware/software that contains all of the terms being searched.
- It l input the text and detect the existence of the search terms.
- It will output to the query resolver the detected terms to allow for final logical processing of a query against an item.

Software Text Search Algorithms

- The query resolver performs two functions.
- It will accept search statements from the users, extract the logic and search terms and pass the search terms to the detector.
- The Query Resolver will pass information to the user interface that will be continually updating search status to the user and on request retrieve any items that satisfy the user search statement.
- The process is focused on finding at least one or all occurrences of a pattern of text (query term) in a text stream.

Introduction to Hardware Text search systems

- Software text search is applicable for many situations but faced some restrictions to handle many search terms simultaneously against the same text and limits due to I/O speeds.
- One approach is to have a specialized hardware machine to perform the searches and pass the results to the main computer which supported the user interface and retrieval of hits.

Introduction to Hardware Text search systems

- Since the searcher is hardware based, scalability is achieved by increasing the number of hardware search devices.
- The only limit on speed is the time it takes to flow the text off of secondary storage (i.e., disk drives) to the searchers.

Introduction to Hardware Text search systems

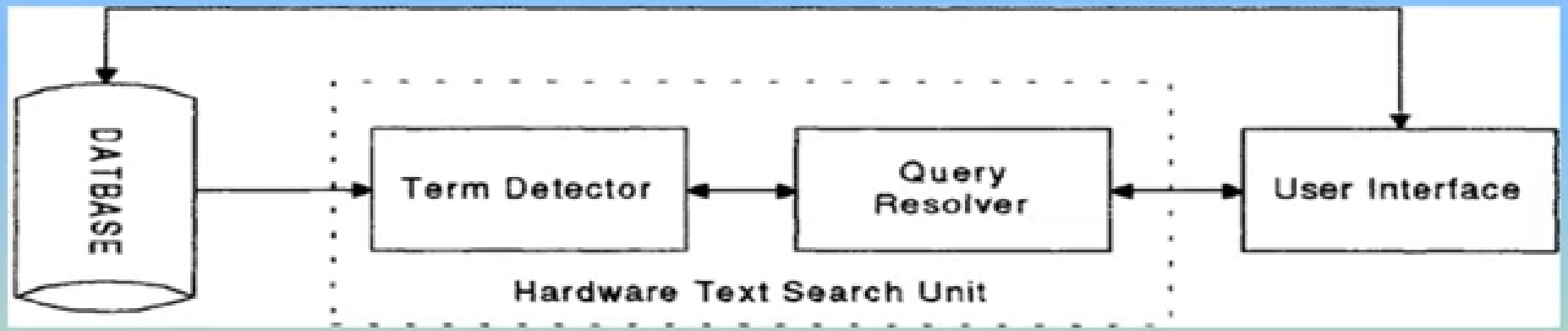
- By having one search machine per disk, the maximum time it takes to search a database of any size will be the time to search one disk.
- In some systems, the disks were formatted to optimize the data flow off of the drives.
- Another major advantage of using a hardware text search unit is in the elimination of the index that represents the document database.
- Typically the indexes are 70% the size of the actual items.

Introduction to Hardware Text search systems

- Other advantages are that new items can be searched as soon as received by the system rather than waiting for the index to be created and the search speed is deterministic.
- Even though it may be slower than using an index, the predictability of how long it will take to stream the data provides the user with an exact search time.
- As hits as discovered they can immediately be made available to the user versus waiting for the total search to complete as in index searches.

Introduction to Hardware Text search systems

- This is very basic Hardware Text Search systems.
- Contains Database,
- Term Detector,
- Query Resolver and
- User Interface.



Introduction to Hardware Text search systems

- One of the earliest hardware text string search units was the Rapid Search Machine developed by General Electric (Roberts-78).
- A more sophisticated search unit was developed by Operating Systems Inc. called the Associative File Processor (AFP) (Bird-77).
- It is capable of searching against multiple queries at the same time.