ISSN: 2393-8528

IJICSE
Open Access
Journal Approved by UGC

Contents lists available at www.ijicse.in

International Journal of Innovative Computer Science & Engineering

Volume 4 Issue 4; July-August-2017; Page No. 06-12

# Information Retrieval Tools for Efficient Data Searching using Big Data

**Mrs.P.Jyothi, Mrs. K.Swathi**

Asst.Professor in Computer Science & Engineering Dept.

TKR Engineering College, Hyderabad

Received 10 May 2017; Accepted 02 July. 2017

**ABSTRACT**

An examination of the structure and components of information storage and retrieval systems and information filtering systems. Analysis of the tasks performed in such selection systems leads to the identification of thirteen components. Of these components, eight are necessarily present in all such systems, mechanized or not; the others may, but need not be, present. We argue that all selection systems can be represented in terms of combinations of these components. The components are of only two types: representations of data objects and functions that operate on them. Further, the functional components, or rules, reduce to two basic types: (i) Transformation, making or modifying the members of a set of representations and (ii) Sorting orpartitioning. The representational transformations may be in the form of copies, excerpts, descriptions, abstractions, or mere identifying references. Bypartitioning, we mean dividing a set of objects by using matching, sorting, ranking, selecting, and other logically equivalent operations. The typical multiplicity of knowledge sources and of system vocabularies are noted. Some of the implications for the study, use and design of information storage and retrieval systems are discussed.

**Keywords:** *Retrieval Tools*

## Introduction

### 1.1 Purpose

Three considerations now encourage detailed analysis of the components of selection systems:

1. Academic curiosity: Can all information storage and retrieval systems (or, better, all selection systems) be viewed as composed of a common set of components? If so, what are they and how many are there? Which are necessary and which are sufficient?

2. The recent Text REtrieval Conferences (TREC) have provided a welcome revival of interest in the comparative evaluation of retrieval and filtering systems. We suggest, however, that there are significant limits to the benefits that can be derived from comparing whole, complete systems. Sooner or later, the advanced design and evaluation of selection system performance also requires the systematic comparative evaluation of alternatives at the level of individual components within complete systems.

3. In the emerging network environment selection systems have moved away from the traditional ``unitary'' model of one retrieval (or filtering) engine operating on one dataset. We now have a situation which we have called ``extended retrieval'' It is easy to think of multiple retrieval engines connected to each other and to multiple databases over networks. But so simple a view begins to break down as soon as one begins to examine how extended retrieval might work: Where are the indexes, for example? Are they part of the respective databases on the server or part of the client retrieval engine? In the NISO Z39.50 Search and Retrieval protocol (cf. ISO 10162 & 10163) an EXPLAIN function is being developed to enable the client to ascertain the available options and constraints of the server. What, in principle, could the server explain about itself that might be useful to the client?

In brief, a general conceptual framework and vocabulary for the components of selection systems is needed. This paper seeks to analyze the ``anatomy'' of selection systems. Such analysis should advance the theory of selection systems: What are the components of retrieval and filtering systems? Which are the necessary and sufficient components and which are optional?

*Corresponding author: Mrs.P.Jyothi*

What different types of components are there? Which functionally similar techniques might be substitutable within any of the components? Which might be substitutable across different, but similar components? In what different ways can the components be combined to design more sophisticated systems? Our hope is that a functional analysis of components will stimulate the design of improved selection systems.

We will first propose a basic functional model of information storage and retrieval systems and discuss these components in some detail. Next, we reduce the non-data components of the system model to two functional types, transformers and partitioners. This is followed by a generalization of the model to other similar selection tasks. Finally, we comment on some of the implications for the study, use and design of selection systems. We approach this in the context of bibliographic and text systems, but believe the approach to be of general applicability.

**1.2 Terminology**

Throughout this paper we will be using several words and phrases in specific technical ways.

System boundaries define what is considered the ``system'' rather than the ``environment''. Inputs flow into the system, are processed, and eventually emerge as output. If the scope of the system is expanded, i.e. additional processes become incorporated into the system, then the system boundaries are moved to include more of what was previously part of the environment.

In examining the decomposition of selection systems into their functional components, a series of processes is found: objects are processed into modified objects, which are, in turn, affected by other processes to become further modified objects. The granularity of the analysis is somewhat arbitrary: processes can typically be broken down into finer and finer sub processes. Hence the level of analysis (the extent to which subsystems are defined) can reasonably depend on the purpose of the analysis.

A transforming operation in this context is the mapping of some procedure across each of the members of one set in order to derive a new transformed set of objects. It is necessarily a one-to-one mapping from the original set to the new set, where each member of the new set is a (possibly) modified copy of its corresponding member of the original set. A simple example of such an operation is copying. Each member of the original set is copied into a new derived set.

At some level of generality all information selection systems processes can be thought of transformations from one state to another, but, for the present purposes, the distinction between two types of transformation appears useful:

- Representation Making. Using rules to derive a representation (a copy or a version) of a datum into a corresponding, modified datum. Data are changed or at least copied.

- Partitioning (sorting, selecting) a subset of data objects according to some criterion expressed as a query for a matching process or as an ordering rule. Data are reorganized rather than changed.

The term ``retrieval'' tends to subsume three meanings: selecting (identifying); locating (lookup); and fetching (delivery). The first meaning -- selecting (identifying) -- is what interests us here. We follow who provide a useful classification of retrieval techniques and characterize the process as a matter of comparing and matching, either exactly or partially. The variety of retrieval techniques -- the form and degree of acceptable comparability -- is very large: exact match; partial match; match using truncation; fuzzy, positional, and other relationships; Boolean matches; etc. Multiple techniques can be combined and there are limitless degrees of progressively weaker matching. We follow regarding the retrieval process itself as a comparing or matching process. However, the purpose or function (as distinguished from the procedures) of this matching is to partition the stored representations into a set of subsets.

In information selection systems, Representations are partitioned into the two subsets: retrieved and not-retrieved, as in basic Boolean systems. But there can be degrees of matching and each different degree of matching can be used to create another partition. The limit is reached in document-ranking systems in which, at least in principle, each representation is partitioned into a separate subset with one member. We can, therefore, while accepting that the process is a matching procedure, emphasize that it is functionally a partitioning activity. With this in mind, we can regard the formal query as being a partitioning instruction. It may sound odd to refer to information retrieval as ``partitioning with

respect to relevance'' but that is an accurate statement of the intent.

In brief, while the process may be one of matching, the function is one of partitioning and we can conclude that this is a different kind of operation from, say, copying. Sorting is logically the same as partitioning: To sort into categories is to partition into categories.

## 2 A Basic Model of Information Retrieval Systems

7,]). Such models are generally in the form shown in Figure 1, with varying amounts of additional descriptive detail depending of the purpose of the description.

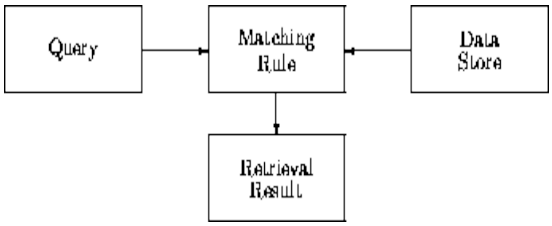


**Figure 1:** General model of information retrieval systems.

Published a ``complete formal model for information retrieval systems'' using production grammars and hypergraphs to represent text structure, indexing, and access. However this is really a procedural model of text retrieval techniques. Descriptions of the operation of individual retrieval systems are likely to have detailed flow diagrams of that particular system's components. Here, however, we are interested in developing a complete, generalized functional analysis of information selection systems.

To develop a complete and general model of the functional components of bibliographic information storage and retrieval systems we proceed by outlining a descriptive model of information storage and retrieval procedures. This illustrative model is intended to be minimally complete in that it includes all the *different types* of functional components found in all retrieval systems. The hope is that the components identified in this basic, illustrative model could be used to construct a functional representation of any information storage and retrieval systems of any complexity, including extended retrieval architectures. As a check on the adequacy of the analysis three examples of information storage and retrieval systems will be examined later.

### *2.1 Input*

System boundaries are arbitrary. Where they are drawn determines which flows of data are regarded as flowing in to and out of the system in whatever form. The inputs, queries and records, may be retained or discarded (perhaps by being relegated to other storage). There can also be feedback concerning any process.

- One kind of input supplies the stored and potentially retrievable data: documents, bibliographic records, images, etc., and/or representations of them.
- Queries constitute another kind of input in one of several forms: free-text, boolean keywords, formal query language statements, etc.
- External knowledge may also be drawn upon in the form of controlled vocabularies, syndetic structures, subject headings, descriptions, etc.

There can be multiple outputs. The most obvious output is the expression of the retrieval results, in whatever form. More generally there can be feedback reporting the effects of any procedure.

With this in mind, we now outline the functional components that appear to be necessary and sufficient to represent information storage and retrieval systems. Not all components are present in all systems. Some components could be present more than once. Components may be implemented in more than one way, i.e. using different techniques. Note also that, as is usually the case in systems analysis, the granularity of the components is somewhat arbitrary. We propose that the following components, displayed in summary in Figure 2, are necessary and sufficient, between them, to represent the functionality of all operational information storage and retrieval systems. The intention is that the analysis will be technologically independent, one that would be as valid for paper-based as for computer-based retrieval systems.

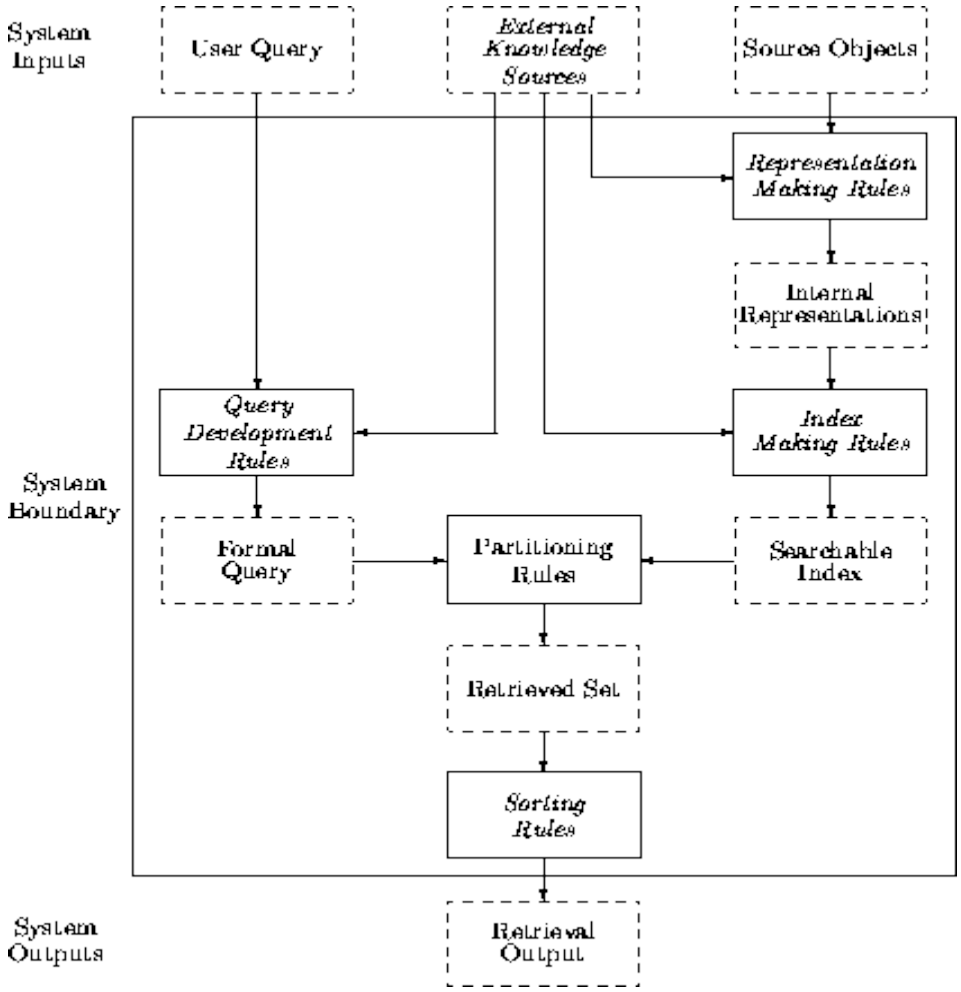


**Figure 2:** A minimally complete model of bibliographic oriented information retrieval (selection) systems. Solid boxes indicate processes (``transformers'' or ``partitioners''), dashed boxes data objects. *Italics* indicate optional components. Arrows show flows (or streams) of information objects. Note that the only required ``process'' is the central partitioning rule, and that subcomponents are formed by patterns of Objects ⇒ Process ⇒Objects.

### *2.2 Input Streams*

**User Query**

One form of input from the environment is the User's query, an expression of the user's information need, more or less compromised by the user's expectation of and experience with the information retrieval system. The ``user'' is ordinarily thought of as a human being, but the query could well be generated by a machine and only indirectly by a human being, such as in the case of relevance feedback or multi-stage retrieval.

**Source Objects**

A set of Source Objects of interest: documents, records, artifacts, images, signals, etc. These arise in the environment outside of the information selection system. A general theory of information storage and retrieval should be able to include bibliographic systems (searching records representing documents), ``full-text'' searching, copies or representations of museum artifacts, and, indeed, of any kind of definable phenomena, including imaginary ones. The set of source objects may well be a carefully selected set, as in a library or museum collection.

These objects and/or copies and/or transformations of them become ``resource input'' to the retrieval system. Through a variety of possible processes, they become the Representations in the system.

**External Knowledge Sources**

External Knowledge Sources are used in information selection Representation Making, Index Making and Query Development. In Representation Making, the external knowledge may be in the form of what people know or has been recorded concerning the Source Objects, their contexts (e.g. domain knowledge), their possible representations (e.g. linguistic

knowledge, thesauri, etc.), or their internal structures and interrelationships, are another resource which can be drawn on as a supplement to or as a substitute for the source objects.

Such knowledge can also be used in Query Development and Index Making in the form of thesauri, controlled vocabularies, subject heading lists, classification schemes, syndetic structures (use, see also, etc.), dictionaries, search intermediaries, etc. One of the major research areas in this field is to see how far this external knowledge can be formalized and moved *inside* the system and used in this way.

In an ideal world these three processes (Representation Making, Index Making, and Query Development) would all draw on the same, identical knowledge sources, but this is unlikely in practice. With the rise of client/server architectures, we can expect separation of the Query Knowledge Sources from the Representation Knowledge Sources and the Searchable Index Knowledge Sources. Knowledge Sources need to be continuously revised and updated and there is no assurance that the updating will be identical and synchronized. Further, the use of an External Knowledge Source in creating Representations is chronologically prior to the use of the External Knowledge Source for Query Development and may be several years prior, creating possible vocabulary problems even if the same External Knowledge Sources were used.

### *2.3 Internal Components*

**Representations**

Representations of the source objects are composed from the resource inputs in some combination of a copy or transformation according to the Representation Making Rules of part (or all) of the Source Objects and/or any (external) representations or descriptions (from External Knowledge Sources) of those resource objects. Representations can be derived from:

1. Source Objects

o Part or the entire object itself possibly copied and/or transformed. For textual objects these could include the text, title, original abstract, etc. For images, these could be scanned copies. (These are the ``brute facts'' of Descriptive features implicitly in or algorithmically derivable from the object: e.g. word occurrence, frequency, and co-occurrence; automatic abstractions from (or patterns recognized in) images; etc.

2. External Knowledge Sources

o Features derivable from other objects inside (e.g. relative word frequency in relation to a corpus) or outside (e.g. synonyms of topical terms) the retrieval system that are related to this object.

o Description or documentation of the object: description of the physical object and/or statements about the origins of the object and/or what the object signifies, e.g. subject headings, subject classification.

Depending on the nature and extent of the Representation Making Rules, the Representations, then, might be more or less transformed copies of the Source Objects: in a collection of unedited full-texts, each text (or copy of it) would constitute its own Representation. It might be a more of less transformed description of the object: in museum registration the representation might include an image of the object, but none of the original object itself (unless, presumably, it is a museum of electronic objects). In other cases the representation could be derived in part from the source object and in part from a description: in bibliographic systems, such as a library catalog, fragments derived from the object (e.g. title, publisher's name) would be combined with pieces of description (e.g. subject headings).

**Searchable Index**

Since the Representation is what is stored, the Representation is also that which could, in principle, be searched and, following selection, produced as output for display or other purposes. But this is not necessarily supported in practice. Current online library catalogs, for example, typically restrict searching to a few fields (notably author, title, and subject headings) within Representations that contain several other fields in which searching is not supported. This is sufficient reason why it is necessary to make a distinction between the Representation and the *Searchable Index*. The Searchable Index, in this technical sense, is the searchable part of the Representation. We use ``Searchable Index Rules'' to denote whatever determines what is to be searchable. Retrieval systems commonly have in addition, a syndetic structure for mapping permissible searches (see, see also, stop words, etc.), which we also treat as a second component of the Searchable Index. Again, in the case of unedited full-text, the Searchable Index will be co-extensive with the Representation and, therefore, with the Source Object (the original text). But, as

noted, in other cases, such as library catalogs, the Index Making Rules can restrict which parts of the Representation are available in the Searchable Index. The Searchable Index (like the Representation and the Source Object) might be partitioned into separate (sub) indexes, to allow more precise, targeted searching.

Procedurally, the Index Making process can be implemented in different ways: the Searchable Index might be derived by literally making parts of the Representations available for searching; it might be derived by copying parts of Representations; it may even be that part or all of the Representations exist physically only as fragments distributed via the Index Making Rules to the Searchable Index to be reassembled if and when needed. But we regard these alternatives as functionally equivalent and are not interested here in the technical details of implementation (storage costs, search effort, delay, etc.) that will make one technique preferable to another.

**Query Development Rules and Formal Queries**

Query development is a function that mediates between the User Query and the Formal Query. It transforms the user's query in order to harmonize it with the system's vocabulary of retrieval commands, index specification, and index vocabulary prior to retrieval. This role has traditionally been seen as an important function for skilled human intermediaries.

Computer-based query development that can match queries with the vocabulary in (or expected to be in) the system's Searchable Index is commonly called an ``entry vocabulary'' module. Examples include CITE, Paper Chase, and Grateful Med Automation of this function is promising and offers scope for expert and probabilistic techniques. ``Entry vocabulary'' modules parallel the syndetic structure, thesaurus and controlled vocabulary aspects of External Knowledge Sources used to create the Representation. It might ideally draw on the *same* thesaurus or other knowledge representation scheme, but it cannot be assumed that the same external sources will be used for these different components.

A query development system may be absent, present, or multiply present in any given retrieval system. (We will discuss query development in more detail below).

The Formal Query is the query as it is seen by the Matching Rule, after it has been transformed by the Query Development Rules. Examples of such formal transformations include truncation, weighting, substitution, normalization, vectorization, etc., many of which are conversions of ``external'' representations to ``internal'' representations. Such transformations apply both to computer and human based retrieval systems.

**Retrieved Sets**

A Retrieved set is logically a subset of the Representations as partitioned off by the outcome of the Matching Rule applied to the Formal Query and the Searchable Index. When displayed (or delivered as output) the retrieved set may be complete copies or very incomplete, transformed versions of members of the set of Representations. Note that this is not necessarily a simple binary outcome: Retrieve and Not Retrieved.

**Sorting Rules**

Commonly, but not necessarily, there is a separate process of sorting the retrieved set. Online library catalogs typically reorder retrieved sets alphabetically by author (strictly, by ``main entry'') prior to display. In card catalogs the order of the retrieved set is predetermined by the order in which the cards were filed. With retrieval systems that generate a strict rank-ordering, the ranked order preempts any postretrieval reordering. For a more detailed discussion see

***2.4 Output Streams***

Retrieval output, traditionally in the form of a display, but increasingly in the form of a stream of objects to be used elsewhere or for some other purpose, completes the basic retrieval cycle. Such streams can be directed to visualization tools, storage for later processing, or use as Input Streams to other selection systems, or as feedback within the system itself.

**References**


1. Nicholas J. Belkin and W. Bruce Croft. Retrieval Techniques. Annual Review of Science and Technology, 22:109--145, 1987.
2. David C. Blair. Language and Representation in Information Retrieval. Amsterdam: Elsevier, 1990.
3. Michael K. Buckland. The Potential of Extended Retrieval. United Nations University Second International Symposium on the Frontiers of Science and Technology: Expanding Access to Science and Technology -- The Role of Information Technologies, Kyoto,

12-14 May 1992., Proceedings. Tokyo: United Nations University Press, (forthcoming).

4. Michael K. Buckland, Barbara A. Norgard and Christian Plaunt. Filing, Filtering, and the First Few Found. Information Technology and Libraries, 12:3:311-319, 1993.
5. Michael K. Buckland, Mark H. Butler, Barbara A. Norgard and Christian Plaunt. OASIS: A Front-End for Prototyping Catalog EnhancementsLibrary Hi Tech, 4:10:7--22, 1993.
6. Michael K. Buckland and Fredric Gey. The Relationship between Recall and Precision. Journal of the American Society for Information Science, 45:1:12--19, 1994.
7. M. A. Cahan. GRATEFUL MED: A Tool for Studying Searching Behavior. Medical Reference Services Quarterly, 8:4:61--75, 1989.
8. T. E. Doszkocs. CITE NLM: natural-language searching in an online catalog. Information Technology and Libraries, 2:4:364--80, 1983.