MODEL AND DEVELOPMENT OF NON-PREDEFINED QUERY IN INFORMATION RETRIEVAL SYSTEM

A Thesis presented to the Department of
Computer Science
African University of Science and Technology
Abuja, Nigeria

In partial fulfilment of the requirements for the award of

Master of Science in Computer Science

By

JOHN ADEWALE ADEBISI

December 2014
MODEL AND DEVELOPMENT OF NON-PREDEFINED QUERY IN INFORMATION RETRIEVAL SYSTEM

BY

JOHN ADEWALE ADEBISI

A Thesis presented to the Department of Computer Science
African University of Science and Technology
Abuja, Nigeria

RECOMMENDED: .................................................................
Professor Amos DAVID
Supervisor

.................................................................
Professor M.K. TRAORE
Acting President and Head, Department of Computer Science

APPROVED: .................................................................
Chief Academic Officer

.................................................................
Date
ABSTRACT

The amount of information available today is extremely large. The increasing need for easier and faster information discovery demands optimal information retrieval techniques. A measure of performance of any information retrieval system is based on the effectiveness and efficiency of retrieval. While some techniques rely on algorithms that improve search, others aim at increasing user’s ability to formulate search queries. Here we present a non-predefined query model for information retrieval system based on a relational database.
DEDICATION

This thesis is dedicated to my friend, Oladapo Ogunsola, of blessed memory.
ACKNOWLEDGMENT

I have come this far by the special grace of God. Unto the Lord be the glory, great things He has done.

I would like to deeply thank my supervisor, Prof. Amos David, for his support all the way. This work has benefited from his deep insights.

A special thank you to Prof. Mamodou Traore, the entire staff and faculty of the African University of Science and Technology, Abuja. Their commitment to students is praiseworthy.

A big thank you to the African Capacity Building Foundation for an award of a postgraduate scholarship; and the Federal Capital Territory Administration for granting me a study leave with pay.

Many thanks to my family, especially Oluwaseun and Omoluabi, for giving me a home advantage.

Last but not least, my course mates, for their company. Without them I would be alone and lonely in the class.
# TABLE OF CONTENTS

Abstract.................................................................................................................................................... ii
Dedication.................................................................................................................................................. iii
Acknowledgement ..................................................................................................................................... iv
Table of contents ...................................................................................................................................... v
List of figures .......................................................................................................................................... vii
List of tables ............................................................................................................................................ vii

1. Introduction....................................................................................................................................... 1
   1.1 Research context ............................................................................................................................ 2
   1.2 Problem statement ........................................................................................................................ 2
   1.3 Research objectives ...................................................................................................................... 3
   1.4 Research methodology ................................................................................................................ 3
   1.5 Organization of work .................................................................................................................... 3

2. Literature review............................................................................................................................... 4
   2.1 Information Retrieval .................................................................................................................. 4
   2.2 Data or information retrieval? ..................................................................................................... 8
   2.3 IR Models ..................................................................................................................................... 10
   2.4 Boolean Retrieval ....................................................................................................................... 11
   2.5 Data warehousing ....................................................................................................................... 13
   2.6 Human-Computer Interface ...................................................................................................... 14
      2.6.1 User interfaces for search by Marti A. Hearst ................................................................. 14
      2.6.2 Designing the User Interface by Ben Shneiderman ....................................................... 15
      2.6.3 Models of Interaction ....................................................................................................... 16
      2.6.4 Design of Search Interfaces .............................................................................................. 17
   2.7 Related Works ............................................................................................................................. 18
      2.7.1 A. David, D. Bueno, P. Kislin, Case-Based Reasoning, User model and IRS 18
3. Research proposal .............................................................................................................. 21
   3.1 Information Retrieval System ......................................................................................... 21
   3.2 Non-predefined Query Model ......................................................................................... 22
   3.3 Interface for IR ............................................................................................................. 24
   3.4 Design Specifications ..................................................................................................... 26

4. Case study ......................................................................................................................... 32
   4.1 Employee Records ......................................................................................................... 32
   4.2 Data Modeling .............................................................................................................. 32
      4.2.1 Entity-Relation (ER) Model .................................................................................... 33
      4.2.2 Graph of Relations .............................................................................................. 34
      4.2.3 Dictionary of Attributes ....................................................................................... 35
      4.2.4 Dictionary for Query ............................................................................................ 35
   4.3 Cross Tabulation .......................................................................................................... 38

5. Conclusion ....................................................................................................................... 43
   5.1 Contributions .............................................................................................................. 43
   5.2 Perspectives ................................................................................................................ 43

References .......................................................................................................................... 44
List of figures

Figure 2.1 A simplified diagram of the standard model of the information access process.17
Figure 2.2 Using CBR with user model in IRS.20
Figure 3.1 The schema of an information retrieval system.22
Figure 3.2 The proposed query model for IR.23
Figure 3.3 Boolean search interface.24
Figure 3.4 Google search interface.25
Figure 3.5 Search query 1.27
Figure 3.6 Search query 2.28
Figure 3.7 Search query 3.29
Figure 3.8 Search query 4.30
Figure 3.9 Search query 5.31
Figure 4.1 Entity-Relation model.33
Figure 4.2 Graph of Relations.34
Figure 4.3 Advanced query 1.36
Figure 4.4 Advanced query 2.37
Figure 4.5 Frequency table.39
Figure 4.6 Cross Tab 1.40
Figure 4.7 Cross Tab 2.40
Figure 4.8 Cross Tab 3.41
Figure 4.9 Cross Tab 4.41
Figure 4.10 Cross Tab 5.42
Figure 4.11 Cross Tab 6.42

List of tables

Table 1.1 Data retrieval versus information retrieval.8
Table 4.1 Dictionary of attributes.35
Table 4.2 Dictionary for query.38
CHAPTER ONE

INTRODUCTION

Information retrieval is obtaining information by searching a repository for items that match user’s information need. According to Losee (1998), retrieval systems often order documents in a manner consistent with the assumptions of Boolean logic, by retrieving, for example, documents that have the terms dogs and cats, and by not retrieving documents without one of these terms. Systems consistent with the probabilistic model of retrieval locate documents based on a query list of terms, such as {dogs, cats}, or may accept as input a natural language query, such as I want information on dogs and cats. A probabilistic system then ranks documents for retrieval by assigning a numeric value to each document, based on the weights for query terms and the frequencies of term occurrences in documents.

We want to know how to “best” formulate a query, and our ultimate interest in measures of human utility: how satisfied is each user with the results the system gives for each information need that they pose? Manning et al (2009). Most everyday users of IR systems expect IR systems to do ranked retrieval, unfortunately relevance ranking is often not critical in Boolean systems. On the other hand, most IR systems rank documents by their estimation of the usefulness of a document for a user query, and there is little or nothing a user can do about it. However, many power users still use Boolean systems as they feel more in control of the retrieval process.

It is correct that the set of retrieved documents are not ranked in Boolean searches. However, the cost of a ranked set is a set that is not fully controlled or understood by the user. In Boolean searches, the user obtains well-defined search sets, which is a clear advantage if searching is considered a learning process. The well-defined set provides better feedback and therefore allows modified search profiles. Hjørland (2014).

Conventional IR systems are built on the Boolean model while most IR systems rely on sophisticated algorithms for better ranking. Most of the materials for ranking are usually documents of an unstructured nature (usually text). Today, research in the field IR are split
among various activities and between (a) optimizing algorithms for ranked systems and (b) extending the Boolean model to increase the selection power of users.

Researchers who are working on the storage side of the information retrieval system are engaged in designing sophisticated methods for identification and representation of the various bibliographic elements essential for documents, automatic content analysis, text processing and so on. On the other hand, researchers working on the retrieval side are attempting to develop sophisticated searching techniques, user interfaces, and various techniques for producing output for local as well as remote users. Chowdhury (2004).

1.1 Research context
The area of study is within the domain of Information Retrieval. IR is a broad area and an important aspect of life. The recognition of the important role information plays in our daily lives has led to an outburst of studies aimed at advancing the field of IR.

Although recent efforts in IR are largely towards the web search, Boolean search is arguably the best search strategy for exact-match information retrieval. One of the criticisms against the Boolean search is that most searchers do not have the skills required to formulate Boolean query. But what about simplifying the Boolean search through a better human-computer interaction? And what about expert searchers who would prefer having full control that only Boolean search provides?

This work is based on the Boolean search of structured information stored in a relational database. More emphasis is on the user interface that simplifies formulating Boolean query and access to relevant information.

1.2 Problem statement
Classical IRS have predefined query interface in which the user’s needs are anticipated and the associated query implemented the IRS. The issue here is on what happens to “ignored” needs that arise after the development of the IRS, based on a relational database.
1.3 Research objectives

The aim of this project is to propose a model that will allow an IRS to provide an open query interface, based on a relational database.

This work is user-centric and so much effort would go into developing the user interface suitable for building dynamic queries.

Lastly, it is expected that it would be possible to query for cross tabulation analysis using our query interface.

1.4 Research methodology

A review of existing literature in Information Retrieval provides a head start for this work. A related work in this area is that of Olubunmi AKINTADE on “Case Based Reasoning in Information Retrieval System: Principles, Evolutions and Applications”, 2007.

A model is proposed that gives searchers full control over the search process. This is followed by developing the query interface ideal for the model.

Microsoft .NET Framework is used to develop the user interface. The choice is influenced by the need to use Excel API to visualize crosstab results.

1.5 Organization of work

This research centres on ad-hoc Information Retrieval. It is organized as follows:

The next chapter gives a brief overview of the key advances in the field of Information Retrieval, and a description of where the state-of-the-art is at in the field. The third chapter introduces the proposed model and design specifications for the user interface. The forth chapter presents a case study for implementation using our model and query interface. The Conclusion is a summary of contributions and challenges to provoke future work in the field of Information Retrieval.
CHAPTER TWO

LITERATURE REVIEW

This chapter presents a brief overview of the requisite knowledge of Information Retrieval, and works related to this research. Materials presented here relate to the past, present and future outlook of Information Retrieval as a field.

2.1 Information Retrieval

The field of IR is vast and there are many aspects of it than can be reviewed in a lifetime. Any view about information retrieval is at best either half empty or half full, and there is no one generally accepted definition of information retrieval.

Information retrieval (IR) is a broad area of Computer Science focused primarily on providing the users with easy access to information of their interest, as follows: Information retrieval deals with the representation, storage, organization of, and access to information items such as documents, Web-pages, online catalogs, structured and semi-structured records, multimedia objects. The representation and organization of the information should be such as to provide the users with easy access to information of their interest. Baeza-Yates and Ribeiro-Neto (2011).

Early developments in IR date back to research efforts conducted in the 50’s by pioneers such as Hans Peter Luhn, Eugene Garfield, Philip Bagley, and Calvin Moores. Baeza-Yates and Ribeiro-Neto (2011). Mooers (1919–1994), who allegedly coined the term Information Retrieval, defined it thus: Information retrieval is the name for the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him. It is the finding or discovery process with respect to stored information. It is another, more general, name for the production of a demand bibliography. Information retrieval embraces the intellectual aspects of the description of information and its specification for search, and also whatever systems, technique, or machines are employed to carry out the operation.
Information retrieval is crucial to documentation and organization of knowledge." Mooers (1951).

According to Van Rijsbergen (1979), “Information retrieval is a wide, often loosely-defined term but in these pages I shall be concerned only with automatic information retrieval systems. Automatic as opposed to manual and information as opposed to data or fact. Unfortunately the word information can be very misleading. In the context of information retrieval (IR), information, in the technical meaning given in Shannon’s theory of communication, is not readily measured. Shannon and Weaver (1964). In fact in many cases, one can adequately describe the kind of retrieval by simply substituting “document” for “information.” Nevertheless, “information retrieval” has become accepted as a description of the kind of work published by Cleverdon, Salton, Spark Jones, Lancaster and others. A perfectly straightforward definition along this line is given by Lancaster (1968): “Information retrieval is the term conventionally, though somewhat inaccurately, applied to the type of activity discussed in this volume. An information retrieval system does not inform (i.e. change the knowledge of) the user on the subject of his inquiry. It merely informs on the existence (or non-existence) and whereabouts of documents relating to his request.” This specifically excludes Question- Answering systems as typified by Winograd (1972) and those described by Minsky (1968). It also excludes data retrieval systems such as used by, say, the stock exchange for on-line quotations.”

For thousands of years people have realized the importance of archiving and finding information. With the advent of computers, it became possible to store large amounts of information; and finding useful information from such collections became a necessity. The field of Information Retrieval (IR) was born in the 1950s out of this necessity. Singhal (2001). The important role information plays in almost every aspect of life explains the interdisciplinary approach to the organization and retrieval of information. Hjørland (2014) examined the role of library and information science and knowledge organization in the design and use of classical databases. He wrote:

Knowledge organization (KO) (also termed information organization) is a research field within library and information science (LIS) that is concerned with, among other things, indexing,
classification, metadata, and the optimization of bibliographical records (in physical bibliographies/catalogs as well as in electronic databases). Information retrieval (IR) is another research field located in both LIS and (today overwhelmingly) computer science. However, these two fields have the same basic objective: to make documents and “information” retrievable. They should therefore be seen as competing fields, and their relative strengths and weaknesses should be examined. Such an examination is difficult, however, because KO and IR are influenced by different ideas and techniques and have different research traditions. The difficulty of achieving coherence may be caused in part by the fragmented and conceptually not highly coherent nature of the research traditions described.

The view expressed by Hjørland (2014) is that KO and IR comprise different approaches and research traditions. He went further to say that the relation between KO and IR has not yet been thoroughly examined. Mooers (1951) considered KO to be part of IR: it included the “description of information,” “specification of information for search,” and is crucial to “organization of knowledge.” A related view was expressed by Kemp (1988), who, however, argued that “knowledge retrieval” should substitute for “information retrieval.”

The meaning of the term information retrieval can be very broad. However, as an academic field of study, Information Retrieval might be defined thus: Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). Online edition (c) 2009 Cambridge UP.

Jeremy Pickens (2000) confines IR to automatic search of collections of documents using computers. Introducing his paper, he wrote “The modern field of information retrieval (IR) began in the 1950s with the aim of using computers to automatically search collections of unstructured online text.” Pickens (2000).

Hjørland (2014) opined that IR should cover both manual and automatic retrieval and should not be limited to computer science. He argued, by limiting IR to automatic processes, the possibility of considering the role of human expertise seems lost; by limiting the field to
computer science, the contributions from other fields (such as linguistics, LIS, and science studies) seem to be neglected. It is also clear that the term information is not generally accepted in this field (and the difference between KO and IR is therefore not likely to be based on differences in meaning of information versus knowledge). Hjørland (2014).

Following from the many definitions of IR, Hjørland (2014) observed that the term IR is not used consistently. In his words:

“We cannot find univocity in the use of the terms IR and KO. On the other hand, people in information science have a fairly good impression of the meaning of these two expressions. The core traditions of IR include the works of researchers such as Salton (1991), Spärck Jones (1972), Van Rijsbergen (1979), and Robertson (1975), all of whom seem today to be more closely connected to computer science than to information science (as this field is understood here, equivalent to LIS). In KO, on the other hand, researchers such as Dahlberg (1974), Ranganathan (1967), Soergel (1999), Svenonius (2000), and Vickery (1986) are among the most well-known and cited. A suggestion could be that IR is about automatic methods in contrast to “manual” or “intellectual” methods in KO...IR as an input–output process, in which the knowledge represented is a “black box,” whereas KO is to a higher degree concerned with what is inside the box, that is, how the universe of recorded knowledge and culture has been represented in bibliographical records and databases. From this perspective, the users are not primarily interacting with computers, but with the universe of recorded knowledge, thus making studies of knowledge and culture important for information science.” Hjørland (2014).

Whereas Information Retrieval is concerned with all the activities related to the organization, storage and access to information; an Information Retrieval System allows searchers to find relevant information from a collection of documents.

Lancaster comments that an information retrieval system does not necessarily change the knowledge of the user on the subject of their enquiry; it merely informs them of the existence (or non-existence) and whereabouts of documents relating to their request. In fact, most information retrieval systems are, truly speaking, document retrieval systems. Chowdhury (2004).
Even though there has been some debate over the years, the two desired properties that have been accepted by the research community for measurement of search effectiveness are recall: the proportion of relevant documents retrieved by the system; and precision: the proportion of retrieved documents that are relevant. It is well accepted that a good IR system should retrieve as many relevant documents as possible (i.e., have a high recall), and it should retrieve very few non-relevant documents (i.e., have high precision). Unfortunately, these two goals have proven to be quite contradictory over the years. Techniques that tend to improve recall tend to hurt precision and vice-versa. Singhal (2001).

Researchers who are working on the storage side of the information retrieval system are engaged in designing sophisticated methods for identification and representation of the various bibliographic elements essential for documents, automatic content analysis, text processing and so on. On the other hand, researchers working on the retrieval side are attempting to develop sophisticated searching techniques, user interfaces, and various techniques for producing output for local as well as remote users. Chowdhury (2004).

2.2 Data or information retrieval?

Although the boundary between the Data Retrieval (DR) and Information Retrieval (IR) is a vague one, Rijsbergen (1979) argued that the dichotomy is useful to illustrate the range of complexity associated with each mode of retrieval. Rijsbergen (1979) offered some of the distinguishing properties of data and information retrieval followed by a brief explanation.

<table>
<thead>
<tr>
<th></th>
<th>Data Retrieval (DR)</th>
<th>Information Retrieval (IR)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Matching</strong></td>
<td>Exact match</td>
<td>Partial match, best match</td>
</tr>
<tr>
<td><strong>Inference</strong></td>
<td>Deduction</td>
<td>Induction</td>
</tr>
<tr>
<td><strong>Model</strong></td>
<td>Deterministic</td>
<td>Probabilistic</td>
</tr>
<tr>
<td><strong>Classification</strong></td>
<td>Monothetic</td>
<td>Polythetic</td>
</tr>
<tr>
<td><strong>Query language</strong></td>
<td>Artificial</td>
<td>Natural</td>
</tr>
<tr>
<td><strong>Query specification</strong></td>
<td>Complete</td>
<td>Incomplete</td>
</tr>
<tr>
<td><strong>Items wanted</strong></td>
<td>Matching</td>
<td>Relevant</td>
</tr>
<tr>
<td><strong>Error response</strong></td>
<td>Sensitive</td>
<td>Insensitive</td>
</tr>
</tbody>
</table>

Table 1.1 Data retrieval versus information retrieval
In data retrieval we are normally looking for an exact match, that is, we are checking to see whether an item is or is not present in the file. In information retrieval this may sometimes be of interest but more generally we want to find those items which partially match the request and then select from those a few of the best matching ones.

The inference used in data retrieval is of the simple deductive kind, that is, aRb and bRc then aRc. In information retrieval it is far more common to use inductive inference; relations are only specified with a degree of certainty or uncertainty and hence our confidence in the inference is variable. This distinction leads one to describe data retrieval as deterministic but information retrieval as probabilistic.

Another distinction can be made in terms of classifications that are likely to be useful. In DR we are most likely to be interested in a monothetic classification, that is, one with classes defined by objects possessing attributes both necessary and sufficient to belong to a class. In IR such a classification is on the whole not very useful, in fact more often a polythetic classification is what is wanted. In such a classification each individual in a class will possess only a proportion of all the attributes possessed by all the members of that class. Hence no attribute is necessary nor sufficient for membership to a class.

The query language for DR will generally be of the artificial kind, one with restricted syntax and vocabulary, in IR we prefer to use natural language although there are some notable exceptions. In DR the query is generally a complete specification of what is wanted, in IR it is invariably incomplete. This last difference arises partly from the fact that in IR we are searching for relevant documents as opposed to exactly matching items. The extent of the match in IR is assumed to indicate the likelihood of the relevance of that item. One simple consequence of this difference is that DR is more sensitive to error in the sense that, an error in matching will not retrieve the wanted item which implies a total failure of the system. In IR small errors in matching generally do not affect performance of the system significantly.

A data retrieval system deals with data that have a well-defined structure and semantics, Baeza-Yates and Ribeiro-Neto (1999) while an information retrieval system deals mostly with documents containing natural language text which is not well structured. Data retrieval
system is where a relational database rightly belong; web search belongs to information retrieval system. However, data retrieval and information retrieval are not mutually exclusive and there is no clear cut demarcation for that matter.

Manning et al (2009) explained: “The term ‘unstructured data’ refers to data which does not have clear, semantically overt, easy-for-a-computer structure. It is the opposite of structured data, the canonical example of which is a relational database, of the sort companies usually use to maintain product inventories and personnel records.” They explained further “In reality, almost no data are truly “unstructured”. This is definitely true of all text data if you count the latent linguistic structure of human languages. But even accepting that the intended notion of structure is overt structure, most text has structure, such as headings and paragraphs and footnotes, which is commonly represented in documents by explicit markup (such as the coding underlying web pages). IR is also used to facilitate “semi-structured” search such as finding a document where the title contains Java and the body contains threading.”

2.3 IR Models

Various IR models and techniques have been developed and the consideration for any search strategy depends largely on the information retrieval systems and the bias of the effectiveness of a particular model. Early IR systems were boolean systems which allowed users to specify their information need using a complex combination of boolean ANDs, ORs and NOTs. Singhal (2001). Even though it is a common knowledge in the research community that the Boolean systems deliver power and put searchers more in control of the retrieval process, Boolean model is challenged by modern search engines and information retrieval (IR) researchers, who often consider Boolean retrieval a less efficient approach. Hjørland (2014).

One of the criticisms against Boolean systems is that “there is no inherent notion of document ranking. Even though Boolean systems usually return matching documents in some order, relevance ranking is often not critical in a Boolean system. However, most everyday users of IR systems expect IR systems to do ranked retrieval. Most IR systems assign a numeric score to every document and rank documents by this score.” Singhal (2001). The score assigned to a document is based on some estimation of the relevance of a document to a user query.
Vector space model, probabilistic model, and inference network model are examples of ranked IR systems.

### 2.4 Boolean Retrieval

Boolean Retrieval is the earliest IR model and has been popular with the information retrieval research community. However, the amount of information available today and the need for easier and faster information discovery justify continuing efforts at developing more effective information retrieval techniques.

First let’s see what Boolean retrieval is before going into the case for its continued use. Boolean retrieval is based on Boolean algebra, named after the English mathematician Boole (1815–1864). This has had a great influence on computer science and plays an important role in online searching. Hjørland (2014). According to Manning et al (2009), the boolean retrieval model is a model for information retrieval in which we can pose any query which is in the form of a Boolean expression of terms, that is, in which terms are combined with the operators AND, OR, and NOT.

For example, a search for documents about both dogs and cats might be expressed as *dogs and cats*. An expression such as *dogs and rabies* or *cats and rabies* may be transformed into *dogs or cats* and rabies; another expression *not dogs and cats* may be transformed into *not dogs* or *not cats*; and so on.

Boolean searching is a kind of “exact-match retrieval” in that given documents or document representations either fulfil the demands or fail to fulfil the demands expressed in search logic. Hjørland (2014). Turtle and Croft defined exact-match retrieval as follows:

- The query specifies precise retrieval criteria.
- Every document either matches or fails to match the query.
- The result is a set of documents. (Croft, 1998)
The opposite of exact match is termed “best match” (or “partial match” or “scoring systems”). Turtle and Croft defined best-match retrieval in the following way:

• The query describes a good or “best” matching document.
• The result is a ranked list of documents.
• The result may include an estimate of quality. (Croft, 1998)

Whether using Boolean queries or ranking documents using document and term weights will result in better retrieval performance has been the subject of considerable discussion among document retrieval system users and researchers. Losee (1998).

Singhal (2001) expressed some reservations for Boolean systems: “Boolean systems have several shortcomings, e.g., there is no inherent notion of document ranking, and it is very hard for a user to form a good search request.” Hjørland (2014) argued that much of the criticism of Boolean retrieval seems to be based on problematic assumptions (which may be overcome by expert searchers in properly indexed databases).

The basic strength of Boolean retrieval is its well-defined selection criteria and the advanced user’s ability to fully control the search process. Singhal (2001) wrote “Even though it has been shown by the research community that boolean systems are less effective than ranked retrieval systems, many power users still use boolean systems as they feel more in control of the retrieval process.” Singhal’s criticism really did not diminish the Boolean systems. “The important thing is whether the user has full control over the search—if he or she wants to have it” opined Hjørland (2014).

Of all the arguments put forward above, only one (that Boolean search is difficult for novice users) seems valid. Hjørland (2014). These results do not in any way prove the superiority of partial match techniques or exact match techniques, but they do suggest that different queries demand different techniques. Further study and analysis are needed to determine which elements of a query make it best suited for partial match or exact match retrieval. Tibbo (1998).
However, despite its shortcomings, Boolean logic is powerful: when the technique has been learned, the searcher is empowered in ways with which no other technique can compete. Hjørland (2014).

Researchers who are working on the storage side of the information retrieval system are engaged in designing sophisticated methods for identification and representation of the various bibliographic elements essential for documents, automatic content analysis, text processing and so on. On the other hand, researchers working on the retrieval side are attempting to develop sophisticated searching techniques, user interfaces, and various techniques for producing output for local as well as remote users.

The Boolean search is useful for many search purposes today even though it is the earliest retrieval model developed for search. The ranked retrieval systems are more suited to matching and retrieving unstructured data. However, they are no match to the Boolean model when it comes to handling structured data and information.

### 2.5 Data warehousing

Immon (1990) refers to data warehouse as a central data repository where data from operational databases and other sources are integrated, cleaned, and archived to support decision making. A data warehouse provides management with convenient access to large volumes of internal and external data. The four characteristics of data warehouses are: subject oriented, integrated, time-variant and non-volatile

- **Subject oriented**: A data warehouse is organized around the major business subjects. That is subject oriented in nature.

- **Integrated**: In data warehouses, operational data from multiple databases and external data sources are integrated to provide a single unified database for decision support.

- **Time-variant**: Data warehouses use time stamps to represent historical data. The time dimension is critical for identifying trends, predicting future operations and setting operating
targets. Data warehouses essentially consist of a long series of snapshots, each of which represents operational data captured at a point in time.

- Non-volatile: New data in a data warehouse are appended, rather than replaced, so that historical data are preserved. The act of appending new data is known as refreshing the data warehouse. Lack of update or delete operations ensures that a data warehouse is free of update or deletion anomalies.

2.6 Human-Computer Interface

Users interact with information retrieval systems through an interface where they are usually expected to express their information needs in the form of a query. Chowdhury (2004). The user interface is the part of every computer system that determines how people control and operate that system.

This section discusses the interface for communication between the computer and human users of information systems. The human-computer interface is less well understood than other aspects of information retrieval, in part because humans are more complex than computer systems, and their motivations and behaviours are more difficult to measure and characterize.

Generally it is assumed that the more effective the system the more it will satisfy the user. A better understanding of human-computer interaction is required for creating systems that facilitate rapid learning and performance, yield low error rates, and generate high user satisfaction. But what makes an effective human-computer interface?

2.6.1 User interfaces for search by Marti A. Hearst

As steps towards achieving these goals, Shneiderman lists principles for design of user interfaces. Those which are particularly important for information access include (slightly related): provide informative feedback, permit easy reversal of actions, support an internal locus of control, reduce working memory load, and provide alternative interfaces for novice and expert users.
An important trade-off in all user interface design is that of simplicity versus power. Simple interfaces are easier to learn, at the expense of less flexibility and sometimes less efficient use. Powerful interfaces allow a knowledgeable user to do more and have more control over the operation of the interface, but can be time consuming to learn and impose a memory burden on people who use the system only intermittently. A common solution is to use a 'scaffolding' technique. The novice user is presented with a simple interface that can be learned quickly and that provides the basic functionality of the application, but is restricted in power and flexibility. Alternative interfaces are offered for more experienced users, giving more control, more options, and more features, or potentially even entirely different interaction models. Good user interface design provides intuitive bridges between the simple and the advanced interfaces.

2.6.2 Designing the User Interface by Ben Shneiderman

Ben Shneiderman, an expert in the field, gave eight golden rules of interface design. These principles can help you create a well-designed User Interface and thereby improve the usability of the system.

1. Strive for consistency
Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.

2. Enable frequent users to use shortcuts
As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.

3. Offer informative feedback.
For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.

4. Design dialog to yield closure.
Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of
accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.

5. **Offer simple error handling.**
   As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.

6. **Permit easy reversal of actions**
   This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.

7. **Support internal locus of control.**
   Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.

8. **Reduce short-term memory load.**
   The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

### 2.6.3 Models of Interaction

Most accounts of the information access process assume an interaction cycle consisting of query specification, receipt and examination of retrieval results, and then either stopping or reformulating the query and repeating the process until a perfect result is found. In more detail, the standard process can be described according to the following sequence of steps:

1. Start with an information need.
2. Select a system and collections to search on.
3. Formulate a query.
4. Send the query to the system.
(5) Receive the results in the form of information items.
(6) Scan, evaluate, and interpret the results.
(7) Either stop, or,
(8) Reformulate the query and go to step 4.

Figure 2.1  A simplified diagram of the standard model of the information access process.

2.6.4 Design of Search Interfaces
User interface design is a practice whose techniques are encompassed by the field of Human-Computer Interaction (HCI). This field studies how people think about, respond to, and use technology, and how best to design user interfaces to suit people’s needs and inclinations. Based on years of experience, a set of practices and guidelines have been developed to facilitate the design of successful interfaces. The practices are collectively referred to as user-centered design, which emphasizes building the design around people’s activities and thought processes, rather than the converse. The design process begins by determining what the intended users’ goals are, and then devising an interface that can help people achieve those
goals by completing a series of tasks. Goals in the domain of information access can range quite widely, from finding a plumber to keeping informed about a business competitor, from writing a publishable scholarly article to investigating an allegation of fraud. Information access tasks are used to achieve these goals. These tasks span the spectrum from asking specific questions to exhaustively researching a topic.

A user interface is designed via an iterative process, in which the goals and tasks are elucidated via user research, and then initial designs are created – often based on existing designs, but potentially including new ideas. These initial designs are tested with prospective users, and then are assessed and redesigned, and evaluated again, in a cycle that can repeat numerous times. This process is necessary to design sophisticated search interfaces including those that can provide some help to users for selection of appropriate search terms by using dictionary and thesauri, automatic spell checkers, a predefined set of search statements and so forth. Chowdhury (2004).

2.7 Related Works

2.7.1 A. David, D. BUENO, P. KISLIN, Case-Based Reasoning, User model and IRS

What best solution for the user’s query has been the major research interest in IRS. A. David et al identified major problem fields. Query formulation, query representation, information representation, query processing, solution evaluation were presented as examples of these fields. According to the authors, two related problem fields have not been well studied and integrated into IRS applications. “These fields are the field of problem definition and that of solution interpretation. These fields are important because, on the one part, the user’s query is just an expression of his information need that does not necessarily describe the user’s problem. This information need is generally a necessary component for solving the user’s problem. On the other part, the user’s evaluation of the solutions proposed by the IRS depends not only on their adequacy with the query but also on the general problem to be solved.”

The research objective was to study how case-based reasoning (CBR) techniques and user modelling can be combined in IRS in order to accelerate relevant solution finding. A. David et
al focused majorly on problem description and went ahead to propose a meta-model for describing problems in IRS.

The following model was proposed for representing a problem in the domain of information retrieval of bibliographic references. Instantiating the attributes of the model should help describe the problem as well as provide a means of comparing two information retrieval problems.

\[
\begin{aligned}
\text{problem} \\
\text{\{user-id\}} \\
\text{\{problem statement\}} \\
\text{\{constraints on document properties\}} \\
\quad \text{\{document type (book | article in act | article in journal)\}} \\
\quad \text{\{concept\}} \\
\quad \text{\{level of specialization (introductory | technical | specialized)\}} \\
\text{\{context\}} \\
\quad \text{\{for who (the user | someone else)\}} \\
\quad \text{\{user’s estimated level of knowledge on the concepts (no knowledge | beginner | expert)\}} \\
\quad \text{\{cognitive process (recall past solution | discover the domain | confirm a hypothesis)\}}
\end{aligned}
\]

The model was implement by considering each problem as a document. Each problem is represented by:
- User-id: the user’s identity
  - Problem statement: the expression of the user’s problem
  - Document type: the document type desired
  - Level of specialization: the required level of specialization of the document content
  - For who: who is concerned by the problem that is either the user or someone else.
  - User’s estimated level of knowledge on the concepts
  - Cognitive process: in which cognitive process the user is in for solving his problem

A user model was proposed and subsequently integrated into METIORE, an information retrieval system that can be applied to various types of domain. An implementation of a user
model allowed the system’s solution to be adapted to the user based on his objects. The system provides various levels of information access functions. The user can discover the content of the database, present query, and request for cross-analysis of the properties of the references.

Figure 2.2   Using CBR with user model in IRS
CHAPTER THREE

RESEARCH PROPOSAL

In terms of research, IR may be studied from two rather distinct and complementary points of view: a computer-centred one and a human-centred one. In the computer-centred view, IR consists mainly of building up efficient indexes, processing user queries with high performance, and developing ranking algorithms to improve the results. In the human centred view, IR consists mainly of studying the behaviour of the user, of understanding their main needs, and of determining how such understanding affects the organization and operation of the retrieval system. In this project, we propose a model that will allow an IRS to provide an open query from user point of view of IR.

3.1 Information Retrieval System

The objective of an information retrieval system is to enable users to search for information. An information retrieval process starts with the need for information that is translated to a search query. The query is matched with documents in the repository and the result of the search is returned as information to the user for further actions.
3.2 Non-predefined Query Model

For a non-predefined query model, we extend the above IRS schema to include a data dictionary. Information retrieval systems built on this model, which operate by means of Boolean search, allow the user to narrow or broaden the search by giving the user access to the data dictionary. The data dictionary exposes some attributes of the underlying databases to help users build extended queries.

It is possible to formulate such extended queries simply because users know enough about the schema of the collection to be able to specify it.

How to “best” formulate a query, given the searcher’s knowledge of the query and database characteristics?
Information systems should not only be made for “average” users but also for high achievers who want to examine things thoroughly and be well informed. To utilize such systems in optimal ways, the user has to know about the databases in which he or she is searching.

In advanced systems, there is a very wide range of possibilities to choose from [] and searches should never be performed without access to the structure and search fields in the single databases used. Discussion about the structure and contents of the data dictionary is deferred to the next chapter.

Figure 3.2  The proposed query model for IR
3.3 Interface for IR

The search interface is the window through which search systems are seen. The role of the search interface is to aid in the searcher’s understanding and expression of their information needs, and to help users formulate their queries, select among available information sources, understand search results, and keep track of the progress of their search. Hearst.

User happiness commonly depends very strongly on user interface design issues, including the layout, clarity, and responsiveness of the user interface, which are independent of the quality of the results returned. Manning et al. According to Shneiderman (1997), when the interface is well designed, it is comprehensible, predictable, and controllable; users feel competent, satisfied, and responsible for their actions.

Search interfaces built on the Boolean model are rich with AND, OR, or NOT increasing the selection power of users. Baeza-Yates and Ribeiro-Neto (1999).

Figure 3.3 Boolean search interface

However, the standard interface for a textual query is a search box entry form, in which the user types a query, activated by hitting the return key on the keyboard or selecting a button
associated with the form. The entry text box accepts a free text query as well as text query combined with search operators (such as Boolean operators). The free text query, which is extremely popular on the web, views the query as simply a set of words. Manning et al (2009).

Figure 3.4  Google search interface

Common search interfaces tend to mask query operators from the end user. A search engine may support such rich functionality, but most commonly, the capability is hidden behind an interface (say an “AdvancedQuery” interface). On the web, few people use advanced search interfaces and most would like to complete their search in a single interaction. Users may also be completely unfamiliar with structured search and advanced search interfaces or unwilling to use them. Manning et al (2009).

Early information systems were built for professional searchers but today’s modern search engines make information searching easy for everybody. One major contribution of web search engines is the simplification of the search interface and the interaction process. However studies seem to indicate that simplicity in searching may come at a high price and the ideological tendency to make things “user friendly” and easy for everybody tends to hurt the development of systems aimed at increasing the selection power of users and search experts.
The Boolean model is valuable in providing users with the power to make informed searches and have full control over what is found and what is not. The search interface for Boolean queries is usually more complex to design than the search interface of most search engines. However, both practitioners and experimenters have long known that some retrieval techniques work better on certain queries and documents than do other techniques. Losee (1998). One may have the best of both worlds by combining features from different search models.

Let’s not descend so low but focus to stay on top of our game, which is to develop an open query interface for Boolean information retrieval.

3.4 Design Specifications

The expression “FirstName = Johnson”, for example, can be combined with another expression, say “LastName = Oyemade” using the logical operator OR to form a Boolean expression:

$$\text{FirstName} = \text{Johnson} \text{ OR } \text{LastName} = \text{Oyemade}$$

FirstName and LastName are attributes of a database as seen by the user, Johnson and Oyemade are data values supplied by the user, = and < are arithmetic operators, and OR is a Boolean operator joining the two expressions. Boolean expressions typically use three operators: AND, OR, and NOT. For a non-predefined query, the interface receives as inputs attributes from a well-structured data dictionary.
Figure 3.5 Search query 1

An expression such as

\[(\text{FirstName} = \text{Johnson} \text{ OR} \text{ LastName} = \text{Oyemade}) \text{ AND Gender} = \text{M}\]

is not exactly the same as

\[\text{FirstName} = \text{Johnson} \text{ OR}\ (\text{LastName} = \text{Oyemade} \text{ AND Gender} = \text{M})\]
One problem we may have to contend with using the above query builder is that there is no explicit way to group expressions using brackets.

To remove this ambiguity, we may avoid mixing up the logical operators, or impose some form of groupings, or limit the query to not more than two expressions. Fig.

below is a simple but powerful query interface.
An alternative user interface allows the user to explicitly determine how expressions are grouped using brackets. This is an advanced query interface for building extended queries.
We can combine simple query and advanced query in a single interface and allow users to switch. This way we cater for both novice and expert users. The search interfaces presented here are open and are as transparent as possible, based on a relational database.
Figure 3.9  Search query 5
CHAPTER FOUR

CASE STUDY

In this chapter, we consider for study the case of a database containing employee records and related facts, such as *first name, last name, date of birth, state of origin*, and so on. The retrieval system is designed to allow the user to search for specific records that match one or more specific conditions or search criteria.

We want to examine how a well-structured dictionary can contribute to making information retrieval easy and transparent, using our model and query interface. We are going to be fair in our discussion of important topics of interest related to the case study.

4.1 Employee Records

We want to represent information about employees in a database. Bio data for each employee is recorded: *first name, last name, gender,* and *date of birth*. An employee is from a particular *local government area* of his/her *state of origin*. An employee can work in at most one *department* and a *department* can have several employees. Each employee belongs to a *job group*, and within each *job group* are *job titles*. An employee reports to his/her *immediate boss*. Each employee is placed on a *salary grade level*. Employees on the same *salary grade level* are entitled to equal *salary*. An employee’s monthly *pay* is a function of his/her *salary*. Changes in *salary structure* are kept for future reference.

4.2 Data Modeling

Briefly we present the entity-relation model and the graph of relations for the employee database. We will examine the dictionary of attributes and then present for consideration an extended dictionary for query.
4.2.1 Entity-Relation (ER) Model

![Entity-Relation Model Diagram]

**Figure 4.1** Entity-Relation model
4.2.2 Graph of Relations

- **LocalGovtArea**
  - LocalGovtAreaID
  - LocalGovtAreaName
  - StateOfOriginID

- **StateOfOrigin**
  - StateOfOriginID
  - StateOfOriginName

- **SalaryGradeLevel**
  - SGL

- **JobTitle**
  - JobTitleID
  - JobTitleName
  - JobGroupID

- **JobGroup**
  - JobGroupID
  - JobGroupName

- **Employee**
  - EmployeeID
  - FirstName
  - LastName
  - Gender
  - DateOfBirth
  - LocalGovtAreaID
  - DepartmentID
  - JobTitleID
  - ImmediateBossID
  - SGL

- **Rate**
  - RateID
  - DateModified
  - SGL
  - MonthlySalary

- **Salary**
  - MonthlySalary

- **Payment**
  - PaymentID
  - Netpay
  - PaymentDate
  - EmployeeID

Figure 4.2 Graph of Relations
4.2.3 Dictionary of Attributes

A dictionary of attributes contain definitions and representation of data elements. One benefit of a well-prepared dictionary of attributes is a consistency between data items across different tables. Below is the list of all the fields of the database indicating their domains.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Data Type</th>
<th>Sample Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmployeeID</td>
<td>Integer (9)</td>
<td>123456789</td>
</tr>
<tr>
<td>FirstName</td>
<td>Varchar (20)</td>
<td>Johnson</td>
</tr>
<tr>
<td>LastName</td>
<td>Varchar (20)</td>
<td>Oyemade</td>
</tr>
<tr>
<td>Gender</td>
<td>Char (1)</td>
<td>M</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>Date (8)</td>
<td>23/02/1980</td>
</tr>
<tr>
<td>LocalGovtAreaID</td>
<td>Integer (9)</td>
<td>356</td>
</tr>
<tr>
<td>LocalGovtAreaName</td>
<td>Varchar (50)</td>
<td>Ibarapa</td>
</tr>
<tr>
<td>StateOfOriginID</td>
<td>Integer (9)</td>
<td>23</td>
</tr>
<tr>
<td>StateOfOriginName</td>
<td>Varchar (50)</td>
<td>Oyo</td>
</tr>
<tr>
<td>DepartmentID</td>
<td>Integer (9)</td>
<td>67</td>
</tr>
<tr>
<td>DepartmentName</td>
<td>Varchar (50)</td>
<td>Information Technology</td>
</tr>
<tr>
<td>JobTitleID</td>
<td>Integer (9)</td>
<td>234</td>
</tr>
<tr>
<td>JobTitleName</td>
<td>Varchar (50)</td>
<td>Principal System Analyst</td>
</tr>
<tr>
<td>JobGroupID</td>
<td>Integer (9)</td>
<td>456</td>
</tr>
<tr>
<td>JobGroupName</td>
<td>Varchar (50)</td>
<td>System Analyst</td>
</tr>
<tr>
<td>ImmediateBossID</td>
<td>Integer (9)</td>
<td>123456</td>
</tr>
<tr>
<td>SGL</td>
<td>Integer (9)</td>
<td>10</td>
</tr>
<tr>
<td>MonthlySalary</td>
<td>Decimal (9,2)</td>
<td>450,910.00</td>
</tr>
<tr>
<td>RateID</td>
<td>Integer (9)</td>
<td>123</td>
</tr>
<tr>
<td>DateModified</td>
<td>Date (8)</td>
<td>29/11/2014</td>
</tr>
<tr>
<td>PaymentID</td>
<td>Integer (9)</td>
<td>123456</td>
</tr>
<tr>
<td>Netpay</td>
<td>Decimal (9,2)</td>
<td>450,910.00</td>
</tr>
<tr>
<td>PaymentDate</td>
<td>Date (8)</td>
<td>30/11/2014</td>
</tr>
</tbody>
</table>

Table 4.1 Dictionary of attributes

4.2.4 Dictionary for Query

The dictionary of attributes in Table 4.1 is prepared by and for the system designer and most times users don’t really profit from it. Discovering information from conventional databases often requires some knowledge of the attributes of the underlying database. However, a user needs an extended dictionary to be able to formulate dynamic queries.
Consider, for example, a query that retrieves all records where *LastName* or *FirstName* equals Johnson:

\[\text{FirstName} = \text{Johnson OR LastName} = \text{Johnson}\]

![Advanced Filter Window](image)

**Figure 4.3  Advanced query 1**

If there is a field, say *Name*, that combines *FirstName* and *LastName*, the query can now be reformulated as:

\[\text{Name contains Johnson}\]

*Name* is a derived field, not one of the attributes of our database, and should not be in the dictionary of attributes. Needless to say that the second query is easier to formulate than the first.
To accommodate search fields not defined in the dictionary of attributes, we propose an extended dictionary for query. Below are the guidelines to build a dictionary for query:

i. Any attribute can be included in query dictionary.

ii. Each field is defined only in terms of attributes.

iii. Each attribute is qualified by its table name.

iv. Any two fields can have the same expressions.

v. Data types are text, number, yes/no, date & time.

vi. Field names should be as descriptive as possible.
<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
<th>SQL Expression</th>
<th>Sample Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Name</td>
<td>Text</td>
<td>employee.firstname</td>
<td>Johnson</td>
</tr>
<tr>
<td>Last Name</td>
<td>Text</td>
<td>employee.lastname</td>
<td>Oyemade</td>
</tr>
<tr>
<td>Surname</td>
<td>Text</td>
<td>employee.lastname</td>
<td>Oyemade</td>
</tr>
<tr>
<td>Name</td>
<td>Text</td>
<td>employee.firstname &amp; &quot;&quot; &amp; employee.lastname</td>
<td>Johnson Oyemade</td>
</tr>
<tr>
<td>Gender</td>
<td>Text</td>
<td>employee.gender</td>
<td>M</td>
</tr>
<tr>
<td>Sex</td>
<td>Text</td>
<td>employee.gender</td>
<td>M</td>
</tr>
<tr>
<td>Date of Birth</td>
<td>Date</td>
<td>employee.dateofbirth</td>
<td>23/02/1980</td>
</tr>
<tr>
<td>Monthly Salary</td>
<td>Number</td>
<td>Salary.MonthlySalary</td>
<td>450,910.00</td>
</tr>
</tbody>
</table>

Table 4.2 Dictionary for query

The dictionary for query shown here is for illustration only. It usually has more entries than the dictionary of attributes.

4.3 Cross Tabulation

Cross tabulation is a tool that allows you compare the relationship between two variables. Cross-tabulation analysis, also known as contingency table analysis, is most often used to analyze categorical (nominal measurement scale) data. A cross-tabulation is a two (or more) dimensional table that records the number (frequency) of respondents that have the specific characteristics described in the cells of the table.

The axes of the table may be specified as being just one variable or formed from a number of variables. The resulting table will have as many rows and columns as there are codes in the corresponding axis specification. Displaying a distribution of cases by their values on two or more variables is one of the more commonly used analytic methods in the social sciences.
### Not Saved - Frequency of visit by Age

<table>
<thead>
<tr>
<th></th>
<th>Base</th>
<th>Under 13</th>
<th>18-24</th>
<th>25-34</th>
<th>35-44</th>
<th>45-54</th>
<th>55+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>204</td>
<td>59</td>
<td>43</td>
<td>38</td>
<td>36</td>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>29%</td>
<td>21%</td>
<td>19%</td>
<td>18%</td>
<td>10%</td>
<td>4%</td>
</tr>
<tr>
<td>Frequency of visit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Daily</td>
<td>18</td>
<td>9%</td>
<td>4%</td>
<td>2%</td>
<td>2%</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>11%</td>
<td>8%</td>
<td>4%</td>
<td>3%</td>
<td>-</td>
<td>0%</td>
</tr>
<tr>
<td>Twice a week</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weekly</td>
<td>64</td>
<td>16%</td>
<td>16%</td>
<td>16%</td>
<td>4%</td>
<td>4%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>31%</td>
<td>8%</td>
<td>4%</td>
<td>8%</td>
<td>2%</td>
<td>2%</td>
<td></td>
</tr>
<tr>
<td>Monthly</td>
<td>67</td>
<td>23%</td>
<td>22%</td>
<td>10%</td>
<td>13%</td>
<td>16%</td>
<td>3%</td>
</tr>
<tr>
<td></td>
<td>43%</td>
<td>11%</td>
<td>11%</td>
<td>5%</td>
<td>6%</td>
<td>8%</td>
<td>1%</td>
</tr>
</tbody>
</table>

**Figure 4.5** Frequency table

The following is age distribution by gender from the employee records, based on our query model. 

Figure 4.6  Cross Tab 1

Figure 4.7  Cross Tab 2
### Figure 4.8 Cross Tab 3

<table>
<thead>
<tr>
<th>Code</th>
<th>(Gender) = (Female)</th>
<th>(Gender) = (Male)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GL+</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GL++</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Figure 4.9 Cross Tab 4

<table>
<thead>
<tr>
<th>Code</th>
<th>(Gender) = (Female)</th>
<th>(Gender) = (Male)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GL+</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GL++</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The crosstab above is two-dimensional. Let’s have a frequency distribution of three dimensional variables – cadre, gender, and grade level. The following are screen shots from analysis to results.
CHAPTER FIVE
CONCLUSION

The field of information retrieval has come a long way, and has enabled easier and faster information discovery. One of the measures of the effectiveness of an information retrieval is how satisfied users are with the results they get from search for information. The quality of results obtained often times depend on the query posed to the system by the user. This work was an attempt to increase the ability of users to formulate dynamic queries through a transparent query interface.

5.1 Contributions
We brought up for discussion data retrieval versus information retrieval as a basis for comparing the different retrieval models. A case was made for the continued use of the Boolean model, and we presented an open dictionary for information retrieval system.

We spent time on simple query for ordinary users, and an advanced query for expert searchers of relational databases. Powered by an open dictionary and a good query template, we successfully performed three-dimensional cross-tabulation analysis.

5.2 Perspectives
With exponential growth in the amount of information available, information retrieval will play an increasingly important role in future. On one end are user interfaces for search; on the other end are optimized algorithms for search. The plain search box of most search engines is okay for free form query but most unsuitable for structured query. On the one hand, the AND, OR and NOT query model is being refined for simple query. On the other hand, we are beginning to witness search engines include search operators and tools for advanced query. The two approaches fit different situations differently, and research is ongoing to seek a common ground in a level working field of information retrieval.
References


Chowdhury, Basic concepts of information retrieval systems, 2004.

Amos David, David Bueno, Philippe Kislin, Case-Based Reasoning, User model and IRS


http://education-portal.com/academy/lesson/cross-tabulation-definition-examples-quiz.html#lesson
http://www.indiana.edu/~educy520/sec5982/week_12/chi_sq_summary011020.pdf
http://www.snapsurveys.com/blog/benefits-cross-tabulations-survey-analysis/