Title Internet search techniques: Using word count, links and directory structure as internet search tools

Name Mehdi Minachi Moghaddam

INTERNET SEARCH TECHNIQUES:

USING WORD COUNT, LINKS AND DIRECTORY STRUCTURE AS INTERNET SEARCH TOOLS

Mehdi Minachi Moghaddam

A thesis submitted for the degree of Doctor of Philosophy
of the University of Luton

The University of Luton,
Park Square,
Luton,
Bedfordshire,
LU1 3JU.

January 2005

ABSTRACT

As the Web grows in size it becomes increasingly important that ways are developed to maximise the efficiency of the search process and index its contents with minimal human intervention. An evaluation is undertaken of current popular search engines which use a centralised index approach. Using a number of search terms and metrics that measure similarity between sets of results, it was found that there is very little commonality between the outcome of the same search performed using different search engines.

A semi-automated system for searching the web is presented, the Internet Search Agent (ISA), this employs a method for indexing based upon the idea of "fingerprint types". These fingerprint types are based upon the text and links contained in the web pages being indexed. Three examples of fingerprint type are developed, the first concentrating upon the textual content of the indexed files, the other two augment this with the use of links to and from these files.

By looking at the results returned as a search progresses in terms of numbers and measures of content of results for effort expended, comparisons can be made between the three fingerprint types. The ISA model allows the searcher to be presented with results in context and potentially allows for distributed searching to be implemented.

ACKNOWLEDGEMENTS

The research is dedicated to my parents who have given me their understanding support and the deepest motivation for the work. My dad past away on early morning of 6th Feb 2005 on my arms without any warning sign and this was a total shock for me. I love him so much and we look alike in so many ways and this is my dedication to my dad.

The writing of a thesis is a long and arduous task. Only with the support of others can anyone survive the ordeal. Fortunately, I have been blessed with an abundance of support from kind advisors.

I wish to express my deepest gratitude to Professor Angus Duncan for his most invaluable advice and fatherly personal encouragement. This thesis would not be possible were it not for support of Professor Angus Duncan. I must thank, a thousand times over, to him. His support has covered an overwhelming expertise in the methodological issues as well as in the philosophical underpinnings of the field without it, this work would not have been achieved.

I would not be where I am today without the sage advice of Dr. Andrew Williamson, I owe many thanks to Dr. Andrew Williamson who have supported, inspired, motivated, me throughout my graduate studies. I would like to thank him for his invaluable and constructive comments on the work and suggestions for improvements. This thesis owes its existence to Dr. Andrew Williamson to his kindness, and encouragement. I wish to thank him more for his assistance above-and-beyond the call of duty.

I would like to thank and express my deepest gratitude to Dr. Alfred Vella for continuing to supervise me to the end of my time at Luton...

Last but not least I would like to thank my younger sister Noushin for all her support these many years and everything she has done for me.

Mehdi Minachi Moghad

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Equations

GLOSSARY OF TERMS AND SYMBOLS

Symbol	Meaning	Page No
a	The search engine Alta Vista at www.altavista.com	page 68
e	A general search engine in formulae	page 85
F	A general HTML File	page 76
FP	A fingerprint	page 74
$FP_{+1}(F)$	The sum of the histograms of the immediate successors of file F: $\sum_{S \subset S(F)} H(s)$	page 76
$FP_0(F)$	The histogram of words contained in F	page 76
$FP_{-1}(F)$	The sum of the histograms of the immediate predecessors of F: $\sum_{p \subset P(F)} H(s)$	page 76
g	The Google search engine at www.google.com	page 68
$H(F)$	A general histogram of file	page 76
p	A general web page	page 77
$P(F)$	The set of predecessors of web page F, files that point to F	page 77
r_{pe}	The reciprocal rank of page p in the results of search engine e: $r_{pe} = 1/(\text{rank of page } p)$	page 84
r_p	The *Community Relevance* of page p is the sum of its reciprocal ranks: $\sum_{e \in \{a, g, i, w, y\}} r_{pe}$	page 85
s_{pe}	The contribution of page p to the strength of search engine e: $s_{py} = (r_{pa} + r_{pg} + r_{pi} + r_{pw}) * r_{py}$ in the case of yahoo for example	page 88
s_e	The *Strength* of search engine e: $s_e = \sum s_{pe}$	page 88
$S(F)$	The successors of web page F, those pages pointed to by F	page 76
σ_p	The score of a general page, for example the number of target words that it contains	page 84
u_e	The *Union Score* of a search engine e: $u_e = \sum \sigma_p$	page 88
v_e	The *Vigour* of search engine e: $v_e = \sum \sigma_p * r_{pe}$	page 89
w	The Webcrawler search engine at www.webcrawler.com	page 68
y	The Yahoo search engine at www.yahoo.com	page 68

CHAPTER 1

1 Introduction

This chapter presents an introduction to Internet searching system also sets out the problems and the principles of the research carried out. The chapter presents objectives of the work of this thesis and finally outlines the structure of the thesis.

1.1 Overview of Internet Searching

Naisbitt [1999] states:

"We are drowning in information but starved of knowledge"

Many changes have been taken place in the field of information supply and demand. As noted by the early 19th century paper was the most commonly used media for information distribution and is still very popular in the present day [Naughton, 1999]. However, currently people are turning towards other media to satisfy their information needs. Other aspects of information distribution that have changed over the years are the sources, use of access and its dynamic nature which are all driving further developments.

Hence this great supply of information is making relevant information gathering so difficult that it is increasingly hard to obtain a clear picture of the information suppliers as well as judging the quality of information. All the present changes in the market of information have influenced the accessibility and quality of up to date information.

1.1.1 The Importance of the Web and Web Search

The volume of information available on the web is increasing at a rapid pace. For example in 2002 it was estimated that the web contained nine billion of the surface web (fixed web pages such as HTML), this equates to 167 terabytes of information and about 39% terabyte of text after removing HTML tags, comments and extra white spaces [Michael, 2003].

As noted by Lawrence and Giles [1998], "the revolution that the Web has brought to information access is not so much due to the availability of information (huge amounts of information have long been available in libraries and elsewhere), but rather the increased efficiency of accessing information. The question is how can users find the information they are seeking in such an unorganised, unstructured and decentralised place? This can be viewed from two different viewpoints:

1) Meeting information demands has become easier due to the multiplication of information supply.

2) It has posed a new problem of retrieving the relevant information.

These two factors are caused by the fact that world wide computer network is available to a large portion of the world population which can retrieve and place information on the Internet.

As of December 1998, 85% of Web users used search services to locate Web pages and 60% used Web directories [Kehoe *et al*, 1999]. However 45% of the users stated that one of the biggest problems of using the Web was the inability to find the information they were looking for. There are many factors, which limit the efficiency of the web, and the significant factor is the dynamic nature of the web.

This nature of the web consists of many dead links and out of date pages that may have changed since they were indexed. Even excluding these factors, finding relevant information using Web search engines often fails. Because in an information retrieval system the information is typically presented in a ranked list ordered according to their estimated relevance to the particular query.

The relevance of every query is estimated on the basis of the similarity between the text of a document and the query. Such ranking systems work well when users can arrange a well-defined query for their searches. However, the users of Web search engines often use very short queries (70% are single word queries [Motro, 1998]) which often retrieve huge numbers of documents.

Thus based on such a compressed representation of particular users' search interests, it is impractical for the search engines to recognise the exact documents that are of interest to the end user. Also many web sites especially commercial web sites now are working to influence rankings. These problems are intensified when the users are working in a foreign language and when they are beginners at performing searches.

A web user is commonly faced with all these problems, so the user will retrieve a vast amount of documents, which are not relevant to their search. Hence these searches are termed low *precision searches*. The low precision of the web search engines, joined with the ranked list presentation, force the users to sort through a large number of documents making it harder for the end user to find the relevant information they want.

1.1.2 Research Background

The work described here was started in 1998 and was part of a much larger study of web search strategies. The project aimed to provide answers to the following questions:

Q1 How can users communicate their requirements for a search more effectively to a search engine?

Q2 How can the 'indexing' of a text based resource be automated effectively?

Q3 Can the user and search engine collaborate to make a more effective search using an iterative process?

This thesis reports upon the investigations of the author on the second question. The theme for this research is to investigate a procedure that may make searching the Internet more efficient. The process of information retrieval has been under study for many years. Despite this, new techniques are needed.

1.1.3 Current Search Strategies

Internet search engines use a centralised indexer architecture approach. Search engines have a program (called robots see chapter 4 section 4.3.3) that traverses the web sending new or updated web pages to their main server where they are indexed. The index is used in a traditional centralised old fashion to answer queries submitted from users. One can ask about current Internet search strategy:

'Why were some things not done differently?'

When undertaking a piece of research such as this many decisions need to be taken and some justification given for these decisions.

This is especially true for a subject such as Internet search where

i) Almost everyone agrees that there is a problem

ii) Almost everyone thinks that they understand the problem

iii) Almost everyone has ideas of what should be done.

Based on these observations one may ask the following questions about the strategy adopted in this work:

a) Why study Internet search in the first place?

b) Why restrict your system to single word search examples?

c) Why do we not have a more intelligent lexical analysis that uses the properties of language and, for example, note that car and cars have similar meaning?

However things are just not as simple as one may think. As recently as 2003, one of the early search engine gurus, Tim Bray writing an aptly named series 'On Search, the Series' [Bray, 2003a] sounded many pessimistic notes on the success of current search strategies.

Bray was co-author of the XML specification and deeply involved in the development of the *Open Text Index*, one of the first search engines.

Talking about the progress of Internet search he states:

"...the fact of the matter is that there really hasn't been much progress in the basic science of how to search since the seventies...." [Bray, 2003b].

This indicates that search is a hard problem and, because of its importance and the fact that many are often frustrated by the results of an Internet search, worth an attempt. This is part of a response to 'Why study Internet search in the first place?, question *a)* above (section 1.1.3 page 5). Later in the series Bray gives us this advice:

"There are two lessons that loom larger than all the others put together. Nobody Uses Advanced Search... Every search engine has an "advanced search" screen, and nobody (quantitatively, less than 0.5% of users) ever goes there. This drove us nuts back at Open Text, because our engine was very structurally savvy and could do compound/boolean queries that look like what today we'd call XPath. But nobody used it. What most people want is to have a nice simple field into which they will type on average 1.3 words and hit Enter, and have the result come back to them. So anyone who's building search needs to focus almost all their energy on doing an as-good-as-possible job given those 1.3 words and no other inputs " [Bray, 2003c].

Early on in this work it was decided to use single word searches as examples. In view of the '1.3 word average' reported above this does not pose a serious limitation on the work nor does it threaten its relevance unreasonably, answering question *b)* above ("Why restrict your system to single word search examples?" section 1.1.3). In response to question *c)* above ("Why do we not have a cleverer lexical analysis that uses the properties of language and, for example, note that car and cars have similar meaning?" section 1.1.3) again Bray may be quoted:

"And in my experience, the effort of wrestling with inflexions and synonyms and antonyms and homonyms and so on, in a search engine, is usually not particularly cost-effective"[Bray, 2003d].

Where this research parts from Bray's work, is that this research looks only at information that is available for all html pages on the web and does not take any special note of Meta data that may be included.

1.2 Project Aims

The brief overview of the web information retrieval suggests that unresolved issues exist in search engines such as web pages are indexed and for measures that might reasonably relate one set of results with another to look for consensus amongst search engines and searching effort. This work looks at a number of related issues in the problem of searching the text contained in the web. This report looks at ways of improving the search process that:

a) are automatic, requiring little or no human intervention,

b) use features contained only in the web and its structures,

It was not an aim of this report to produce a working search engine, after all there are many in existence already. This is a study of the information that may be freely available to a new search engine and to assess its potential use.

1.2.1 Main Focus of Study

Information about the contents of a web page is contained in:

a) The name of the page,

b) Its title and other tags,

c) Its 'content',

d) The directory in which it is stored,

e) The other files that are 'near by' on the web.

Other sources of information are available such as the many indices that have been produced by search engine providers. In this research only information that is 'freely available' is considered and comment restricted to the five items listed above. For this purpose, the name of the page is taken to be the complete URL. Because of many historic factors this does not always reflect the content of the page.

URLs have until relatively recently been restricted to the common 'www.xxx.yyy.zzz' style that does not lend itself well to encapsulating meaning. However with the newer ideas which allow more freedom in the choice of name there is hope that more meaningful names will dominate the Web in the future.

The title and other tags (a tag is a command in HTML that specifies how a document should be formatted) are embedded in the Web file itself. Tags in particular have been abused in the past in order to persuade search engines to recommend pages that have dubious relevance to the search. Some mechanism for penalising sites that regularly abuse this feature are increasingly being used but this may cause genuine tags to be less useful then they would otherwise be [Heydon and Najork, 2001].

The contents of the file in terms of the words it contains clearly is the actual determiner of relevance. As natural language processors become more accurate and less computationally costly, there is a hope that more of the web can be 'semantically' analysed [Strzalkowski *et al*, 2002]. This will certainly play its part in the future improvement of search engine development. However at present the development of a system that can be comprehensively implemented is some way off and other techniques need to be investigated in the meantime.

The directory in which an item is stored was the main way in which Internet resources were accessed prior to the web becoming popular. In the days before web browsers such as Mosaic made the current web structures possible, systems such as WAIS (Wide Area Information Servers), and Gopher were used to find information on the Internet.

Gopher was originally designed as a distributed document delivery system for the University of Minnesota. It has a very simple menu-based interface. Each menu entry leads to another menu, or to a file that is retrieved and viewed [Christopher, 2001]. Although directory structure is often a matter of personal preference of the web developer and thus may contain little real clue of the content of a page, the other files that are 'near by' on the web, in terms of links for example, may well give more clues.

One of the unique features of the Web is the links that exist between web pages. These links enable readers to navigate quickly between the pages and so the pages can be thought of as being 'near' each other on the web, even if they are physically on different computers many thousands of miles apart. Utilising this nearness is likely to form an important aspect of future search engine strategies and is one of the ideas investigated in this work.

1.2.2 Research Approach

This research addresses the above challenges by designing, implementing, an **I**nternet **S**earch **A**gent called ***ISA***. The analysis of the problem and the required specification of ISA are described later. The approach that this research presents is to view the web as a giant bank of documents, where structural relations (tree directory structure) are represented as document relations, which provide a uniform framework for accessing web pages, this research looks at the following:

- Specifying types of structural information to extract from the web and Hyperlink relational (see chapter 5, section 5.1.2),
- Developing a system that implements the abstraction of the web (see chapter 5 section 5.3.1),
- Evaluating current search engine technology (see chapter 5, section 5.2.4),
- Developing a search tool that will reside within an existing search engine (see chapter 6).

1.2.3 Basic Research Question

In this work the term '*Fingerprint*' (see section 5.1.1) is used to indicate a property of a file that, although not necessarily unique to that file, distinguishes a file from most others [Minaji and Vella, 1999].

Different fingerprints will be useful for different purposes. For example a fingerprint that is not likely to be useful for the purposes of search is the file length. That may however be a useful fingerprint if one were looking for a file containing, for example, the whole of the Encyclopaedia Britannica.

Again, not necessarily useful for the main aim of this report is the date stamp of a file, which would be useful for some purposes but not (yet) for content search. To illustrate an issue here it may turn out that some time in the future search engines might include a date facility (if file dates could be expected to be reliable).

So, for example, looking for information about the *Hutton Report* the search engine could be instructed to look only at files produced after 2002. This research investigates the following question:

"Can the use of '*Fingerprint*' type ideas improve the retrieval of information from large poorly structured databases such as the web?"

The fingerprint, of the file, used in this work is based up on histogram of the words appearing in the files or related files.

As part of the study the following sub-questions need to be considered:

i) Information about current search engines

   a) What do current search engines do?

   b) Do they do the same thing?

   c) Can a 'consensus' amongst search engines against which each others may be evaluated be formulated?

ii) Information about a sample of the web, its structure and some statistics

   a) What are typical values for the number of directories in a domain?

   b) What are typical values for the number of files in a directory?

iii) How well do current search engines do against the naïve, but relatively simple (to calculate), measure of number of the occurrences of the target word?

For the sample of the web considered in this work and the particular method of searching and the scoring system described in chapter 5 which uses the number of search terms in a document as a score, the following questions may be asked:

A How many files need to be searched to get x% of the total possible score? In this study the term '*score*' is a measurement for matching the search query to a search result (see section 5.2.1). Also in this work 10%, 25%, 50%, 75% and 100% values are given.

B How many search results are needed to get x% of the total possible score? In this work representative values of x% of 10%, 25%, 50%, 75% and 100% are used.

C How many files need to be searched to get the top x% of search terms in the results?

1.2.4 Current Problems of Internet Searching

Many problems face the search engines would be search engine provider. These include virtual pages, spam and the dynamic nature of even the 'static' pages.

Many pages that are viewed through browsers do not actually exist but are produced 'on the fly' as they are requested. This allows pages to be personalised, more dynamic and up to date.

As there are for all intents and purposes an 'infinite' number possible such of pages these are not easily addressed at present (although some ideas are available). For example many of these pages have a template from which they are built by substituting values. It may well be possible to account for these in an intelligent search engine.

Some web page providers want readers to be attracted to their pages even if they are not interested in the content of their web pages. Page hits are an artificial measure of a web page's success. Many web sites are funded mainly through their ability to attract readers and it seems of little interest to their producers if those so attracted benefit from the experience.

The web changes daily and the search engines are not able to keep up with this. A possible solution to this (that may have other problems of course) is to have providers inform the search engines about updates.

This could be done in a number of ways:

a) For example by providers sending new pages to the engines. This could be done to a central repository of new pages. It also has the added benefit of producing an historic record of web development as well as its potential high (total) coverage. One possible problem of this strategy may be the large volume of data/traffic that such a system would generate if implemented properly.

b) Providers could send addresses of new or changed pages to the engines. This distributes the load and allows search engines to be more selective in their trawl (although such selectivity could also be provided with a central system).

c) Providers might send the results of some process to the search engine's repository. This could work in a similar way to the popular SETI@Home and other screensavers in that spare processing capacity is used at the server to produce information and send it batch wise to a central site for collation.

Another limitation is using information that is in language other than English. This study will work only with information in English, that is in HTML and that is freely available. As large portions of pages distributed for free on the World Wide Web are in English, this is a realistic limitation. Future applications could convert dynamic, PDF, and other forms of electronic publication into text before analysing them with the program.

1.3 Structure of the Thesis

The remainder of this thesis is organised as follows:

Chapter 2 explores the literature and previous work in the field of information retrieval and places this research in the context of this developing field. The chapter describes the classic problem of searching.

Chapter 3 describes practical query operators for the manual formulation of structured queries in modern information retrieval systems. In this chapter the challenges of searching the Web are discussed.

In chapter 4 the Web Search Problem is formally defined and various automatic browsing techniques that can provide relevant information are discussed, also this chapter describes how traditional Information Retrieval engines can address the Web Search Problem.

Chapter 5 presents a searching strategy applied to this research and the scoring system that is used for ranking search results, as well as presenting measurements which are utilised in this study.

Chapter 6 discusses the ISA study done with the sample Web to retrieve relevant pages and the challenges faced in the design of the ISA program.

Chapter 7 describes experiments that evaluate four common search engines and compares three different methods for indexing web pages.

Finally, chapter 8 summarises the research achievements and focuses on the future work identified from the work of this thesis.

CHAPTER 2

2 Traditional Information Retrieval

This chapter studies Information Retrieval (IR) techniques. It also outlines the context behind the research question for using the text Information Retrieval (IR) system and suggesting Oard and Marchionini [1996] IR framework for using in this study.

This chapter summarises the history of IR. It introduces retrieval techniques and explains the Oard and Marchionini framework of information seeking tasks (Collection, Selection and Display). It presents each one of these tasks and highlights the need for data collection and the problems of searching. This chapter also provides a brief review of the role of the user interface in IR.

2.1 Traditional IR

Finding the desired information from a large information collection was a problem whose systematic solution was developed about four thousand years ago by librarians, who kept track of books by cataloguing them by author and title [Arlene, 2003].

Using a catalogue for finding books was an improved and easy way for searching of books physically, but it did not provide information, other than the title, on the content of the books. Keywords were the next improvement, introduced in the 16th century in the form of crude indexes. The keywords used were topic, subject and with pointers to the documents [Wheatly, 2002].

Using indexes was based on the idea of providing the selection of appropriate keywords, which would lead the user to the list of the related documents. The keyword index strategy was more or less an expansion of the catalogue method as it allowed searchers to find related items through a given concept (i.e. keyword); but along with the ease of use the problem of ambiguity in representation arrived. The choice of keywords for a given document depends on the subjective word choice for a specific interpretation of the document, as there are no rules for assigning keywords to documents.

In the 1950s with the advancement in computers, computerised information searching was introduced, in which the strategy of using the terms in the document collection to create the index was explored [Luhn, 1957]. In this method the searcher needed to guess just one word out of many used in the document to find a document. The problem with this method was that instead of not finding any relevant document due to improper choice of keywords, it often found too many documents, many being irrelevant [Witten *et al*, 1994].

Witten raised the question of how a system that can accommodate different expressions of information has the ability to pick the relevant items. An ideal IR system should have the ability to identify the ingredients (intent) of the information search along with the contents of the information source, in order to satisfy the information needs. In the late nineteen sixties came the first online catalogue containing bibliographic data.

With the online catalogue the access to information was more efficient, effective and speedy as it allowed a skilled librarian to rapidly search millions of records. The first online catalogues containing bibliographic data came into existence in the late 1960s [Leigh, 1998]. Online library catalogues have gradually become more user friendly with the use of menus and simple commands. Online library catalogues system usually forms an integral part of automated library systems, which includes circulation routines as well as acquisition processing and contain the details of books, conference publications, reports, and periodical titles.

With the advancements in technology, computing power and storage capacity have increased but the amount of information that the typical user must process has also massively increased. Real time data handling was not practised much in the 1960s because computers did not have the capacity to handle data in real time and so bibliographic records, controlled vocabularies and elaborated query languages were used. In the 1970s and 1980s, bibliographic data provided a path for full-text abstracts, that in many cases were replaced by the complete texts of documents.

In the 1990s people could access millions of documents from many locations with the help of the Internet. Fuelled by the expansion of computer capacity a variety of information exploration tasks have evolved. The limitation of users restricted to search through limited indices with restricted vocabularies is over and much texts are now available online. Users may now browse through these electronic repositories, as if they are in a comfortable bookstore rather than a mail order catalogue. Resource constraints and CPU usage time are no longer a significant restraint for users.

The technological innovations driving hardware design has outstripped the development of software user interfaces that mediate access to information. Thus many commercial systems still rely on complicated query languages as channels of communication between the user and machine. Few systems make useful recommendations to the user about possible passages of exploration.

Although the growth of computer networks and information services has already enabled users to use information sources, there is still a need for a new approach and strategy for information retrieval for the Internet. A new search strategy that can improve efficiency of retrieving information for a user request and also can have better coverage of searching the information which is available online.

2.1.1 Terminology of IR

Information Retrieval deals with enabling users to gain access to a large amount of predominantly textual information. Jones and Willet [1997] indicated that the basic technical issues handled by IR systems are:

- Representing the documents that consist of its data collection,
- Performing the calculations required for retrieving the relevant documents.

User interfaces to these systems range from command-line queries to more sophisticated browsing environments [Hearst, 1995].

The basic intention behind this review is to capture those aspects of information retrieval technology that impact on searching strategies. Defining what is meant by Information Retrieval is an important start point. Two different definitions of IR are:

- ***Salton*** [Salton, 1989]:"Information Retrieval systems process files of records and requests for information, and identify and retrieve from the files certain records in response to the information requests. The retrieval of particular records depends on the similarity between the records and the queries, which in turn is measured by comparing the values of certain attributes to records and information requests."

- ***Kowalski*** [Kowalski, 2000]:"An Information Retrieval System is a system that is capable of storage, retrieval, and maintenance of information. Information in this context can be composed of text (including numeric and date data), images, audio, video, and other multimedia objects."

The definitions of Salton and Kowalski imply the development of distributed networked information systems in general, and perhaps the Internet in particular more than , is reflected in the Kowalski definition. Salton's definition has something of a database viewpoint to it, whereas Kovaliski's definition is broader and applies much more to the contemporary environment. For the purpose of this project the Kowalski definition is chosen.

2.1.2 Theoretical Framework of Information Searching

A conceptual framework dealing with the related problem of information filtering, also known as selective dissemination of information in library and Information sciences was presented by Oard and Marchionini [1996]. The framework deals with large amounts of dynamically generated information in response to a user query. Filtering, i.e. selecting what should pass according to a relatively stable profile was discussed.

Oard and Marchionini introduced the term "information seeking" to describe any process by which users get information from automated information systems. The main purpose is to present users with direct information, or information sources, that are likely to satisfy their information needs. Information sources are the entities that contain information in a form that can be easily interpreted by a user. Documents are the information sources that contain text. These information sources may be in other contexts like audio, still or moving images, or even people.

Oard and Marchionini (see Figure 2.1) divided the process of information seeking into three subtasks:

- *Collecting* the information sources (see section 2.2).
- *Selecting* the information sources (see section 2.3).
- *Displaying* the information sources (see section 2.4).

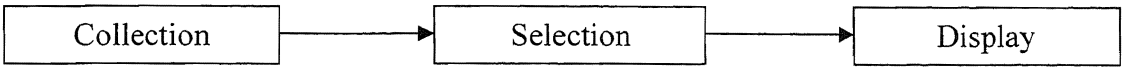


Figure 2.1 Information seeking task [Oard and Marchionini, 1996]

In this research it was decided to use Oard and Marchionini [1996] framework. The three tasks in this framework are fundamental to a process commonly referred to as "Internet Information Retrieval" in which the search system is presented with a query by the user and expected to produce information sources, which the user finds useful. "Text retrieval," the specialisation of information retrieval to retrieve text has an extensive research heritage

Three tasks in this framework are clarified in following sections. The first subtask is *Collecting* the information sources (see section 2.2) the second subtask is *Selecting* the information sources (see section 2.3) and the third subtask (last subtask) *Displaying* the information sources (explained in section 2.4).

Last but not least this study is an investigation of an IR for the Internet, thus the Oard and Marchionini framework needs to be adjusted with the Internet environment in mind. Hence, the issues concerned with this topic are reviewed and explained in chapter 3 and chapter 4, which discuss IR for the Internet, review the search engines system (which is common IR system for the Internet) and summarise the main points for further investigation.

2.2 Document Representations (Collection)

The *Collection* subtask of the Oard and Marchionini framework contains all the data needed for searching relevant information. The *Collection* subtask in a conventional IR system is used to co-ordinate the information using a variety of models; such as the Vector model [Salton, 1989] (see section 2.2.1) and Inference Networks [Haines and Croft, 1993] (see section 2.2.2). Each of these takes the requirements of each individual search and performs a specific query regarding every individual model.

Organising the *Collection* subtask is an important role for the searching process, thus *Collection* subtask should be organised in such a way that it is suitable for the searching of every individual model by preparing data to be entered in the respective model to its specific needs. Several ways have been found for storing the contents of a document (text) in the computer. These range from *Vector* model [Salton, 1989] to *Inference Networks* [Callan *et al*, 1992] as described in this chapter. These indexing methods are designed to store pre–computed partial search results that may be combined at run time to yield the documents matching a given query.

Not every word in the database is indexed: stop words and terms that occur frequently (in every, or almost every, document) are excluded because they are not effective in searching among documents. Since the Oard and Marchionini [1996] framework is utilised for this study, the *Collection* subtask of the retrieval system is used for the sample data required for this research. Data sampling is explained in chapter 5. In chapter 6 it is explained how this research implements a particular system to collect and generate sample data.

2.2.1 The Document Vector Model

The Vector model [Salton 1989] is a classic model of document retrieval based on representing documents and queries as vectors of index terms. The vector model was developed by Salton, to side step some of the information retrieval problems.

The document vector model transforms textual data into numeric vectors and matrices, then employs matrix analysis techniques to discern key features and connections in the document collection. The document vector model represents each document by a vector of numbers, each element of which represents the presence of a unique term. The numbers may be 1 or 0 (indicating presence or absence, respectively), or they may vary over (0,1) to indicate the relative importance of each term in the document.

Yates and Neto [1999] discus the vector model that relies on three sets of calculations. The calculations needed for the vector model are:

1. The weight[1] of each index word across the entire document set needs to be calculated. This answers the question of how important the word is in the entire collection.

2. The weight of every index word within a given document (in the context of that document only) needs to be calculated for the total number of documents in the system. This answers the question of how important the word is within a single document.

3. For any query, the query vector is compared to every one of the document vectors. The results can be ranked. This answers the question of which document comes closest to the query, and ranks the others as to the closeness of the fit.

Documents may be compared for similarity by taking the cosine product of their vectors; the larger the product, the more similar the two documents are considered to be [Lee *et al*, 1997].

For the vector model definition[2] see Yates and Neto [1999]. In addition, they explains the query vector and document vectors.

[1] Weigh quantifies the importance of the index term for describing the document semantic contents.
[2] Yates and Neto,[1999] describes the vector model definition in their book page 27.

2.2.2 Inference Networks

The inference network model associates random variables with index terms, the documents, and the user queries [Turtle and Crof 1992]. Inference networks are designed to combine multiple sources of evidence when estimating the relevance of a particular document to the user's query.

The inference network consists of two sub networks [Callan *et al*, 1992]:

- Document Network
- Query Network

The Document Network is produced during indexing and the Query Network produced from the query text during retrieval.

Haines and Croft [1993] explained that the Document Network represents the document collection and consists of nodes for each document (called document nodes) and nodes for each concept with the collection (document concept nodes).

The document nodes represent the retrievable units within the network, that is, those documents that user then wishes to see in the resultant ranking. A causal link between document node and the document concept node indicates that the document content is represented by the concept. Each link contains a conditional probability, or weight, to indicate the strength of the relationship. The evaluation of a node is done using the value of the parent nodes and the conditional probabilities.

2.3 Searching for Relevant Information (Selection)

In an Information Retrieval system each document is described by a set of representative keywords called *index terms* or *term* and an *index term* is a (document) word whose semantics helps in remembering the document's main topics [Lee *et al*, 1997].

Thus, *index terms* are used to index and summarise the document contents. It should be clear that distinct *index terms* have varying relevance when used to describe document contents. This effect is captured through the assignment of numerical *weights* to each index term of a document; a *weight* quantifies the importance of the index term for describing the document semantic contents [Bookstein *et al*, 2003].

Having represented the document collection and the information need in some manner, the next step of IR is to determine the "relevance" relationship between the two. The conventional approach is to compute for each document a quantitative measure that denotes the similarity, the probability of relevance, or the degree of nearness with respect to a given query. Such quantitative measures, otherwise known as the retrieval status value or simply document scores, are greatly influenced by the choice of term weighting [Salton and Buckley, 1990; Jones and Willet, 1997].

The determination of how the "relevant" documents are identified, as well as how terms are weighted, depend largely on the underlying assumptions of the IR models which not only formalise IR with sound theoretical underpinnings but also provide rationales for evaluating and analysing retrieval strategies. An example of an IR model is the *Boolean* model (see section 2.3.1). Boolean queries have been used traditionally to perform set based retrieval in full text databases. Set based retrieval partitions the database into two sets: the set of all documents matching the query, and the rest of the database.

Rijsbergen [1979] explained that the process of information searching could be illustrated with a black box system (see Figure 2.2.). The inputs to the system are the user's query and the existing documents. When the user gets an output result, he or she may apply feedback to the system in order to change the query to get a better result in the next search. However, this feedback component is not present in all information retrieval systems.

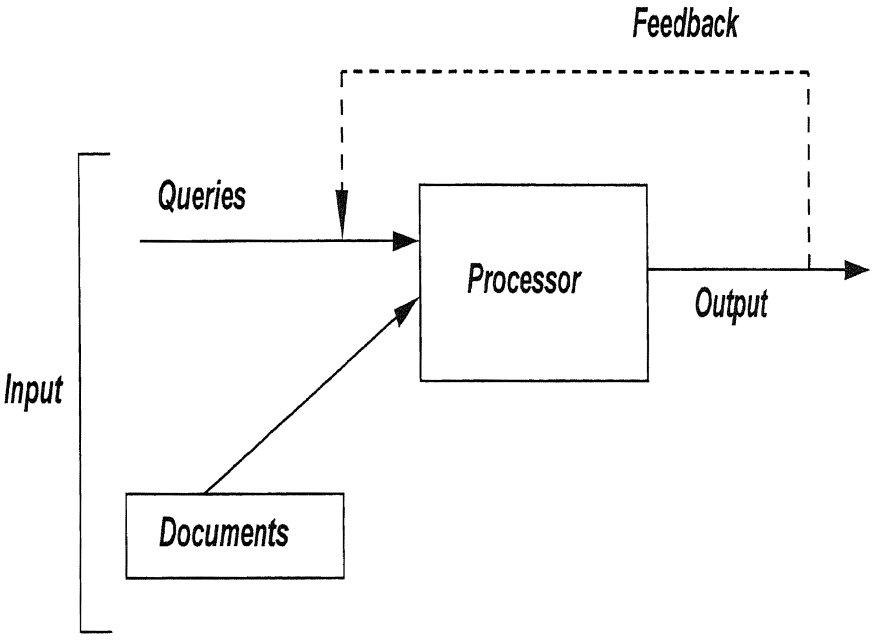


Figure 2.2 A typical Information Retrieval system [Rijsbergen, 1979]

There can be different subtasks involved in the information retrieval task that depend on the nature of the information need. They vary from locating a known item, finding an answer to a specific question and a general inquiry about a topic.

The biggest challenge of IR research is to effectively assess the user query along with the document contents and the relationship between them. Document representation is a crucial aspect of IR because it affects not only the correct assessment of information need and document contents but also the effective matching of the two.

2.3.1 The Boolean Approach

In IR systems, the Boolean model [Cooper, 1988] was the first to be implemented. In this approach the query terms represented with Boolean operators were compared against the *index* terms. The *index* is accessed by some search method such as Boolean approach. Each index entry gives the word and a list of texts, possibly with locations within the text, where the word occurs.

Boolean algebra is used for the Boolean model. In this approach words are logically combined with the Boolean operators AND, OR, and NOT. For example, using two logical statements x and y, the Boolean AND for them means that both x AND y must be satisfied, while the Boolean OR of these two statements means that at least one of these statements must be satisfied. Any number of logical statements can be combined using the three Boolean operators.

The advantage of these operators is that the response time in the *index* search becomes more reliable and fast [Charoenkitkarn *et al*, 1995]. The Boolean logic is very easy and has been adopted widely.

However, a problem is that the general public is not trained in its use, and does not know how to use the logic behind these operators to generate effective queries [Turtle, 1994]. An IR system could retrieve either too many or few documents due to a change in a single term in the query because of the sensitivity of the results to changes in the Boolean operators.

Hence, to overcome this difficulty some changes were made in the retrieval systems. New systems were introduced which ranked the output by assigning weights to each word of the query on their presumed importance [Turtle, 1994].

Other refinement strategies, such as controlling the query formulation process to ease the difficulty of constructing complex Boolean queries [Attardi *et al*, 1999] were investigated as well.

2.4 Retrieval Outcome (Display)

Display is the last subtask of the Oard and Marchionini [1996] framework. In this section information searching interfaces (Display subtask) and the role of the interface in facilitating the specification of information retrieval are briefly reviewed.

Typically, full text retrieval systems provide a list of retrieved titles and allow users to page through documents, one document at a time. Hypertext interfaces, on the other hand, have frequently contained graphical navigation aids such as overview maps, time line visualisations, and paths (see Nielsen [1990], for an overview).

Three dimensional visualisations of hierarchies [Robertson *et al*, 1994] and graphs [Plaisant *et al*, 2002] have also been proposed for overview diagrams [Kules, 2003]. The intent of such displays was to provide sufficient global context to the reader to prevent disorientation and to allow backtracking.

2.4.1 Information Searching Interfaces

In the present day, with the number of databases accessible on line and the subsequent need for enhanced techniques to access this information, there has been a strong rise in interest in the research carried out in the field of information retrieval (IR).

In the past a small community has carried out IR research. Applications of text retrieval focused on bibliographic databases, based on basic Boolean logic approaches to text matching, and paid little attention to research on subjects such as user interfaces, retrieval models, query processing, term weighting and relevance feedback [Chignell *et al*, 2001]. However, in the present day the circumstances have changed. Retrieval approaches based on IR research have influenced major information services such as the World Wide Web (for example, Google and Yahoo).

User interfaces which were developed for the information retrieval (IR) systems have been increasingly supportive in progressive interactive search formulation and refinement. The main aim of the developer is to make IR simpler and easier (user friendly) for the end user. One of the examples for achieving a user friendly interface has been to design "intelligent interfaces" that manifest some of the knowledge and functions of human intervention (e.g. [Belkin *et al*, 1993]).

Another example of a user interface for the IR system is to build an interface that actively supports communication with an intermediary or other user; utilising this approach amplifies the interactive nature of the IR system to incorporate other human resources [Modjeska and Waterworth, 2000].

2.4.2 Main Aspects of IR Interface

Users who use an information retrieval system often have little understanding of how they can make best use of it. A user interface for an information retrieval system should help users formulate [Drabenstott and Weller, 1996]:

- Users queries
- Select among available information sources
- Understand search results
- Keep track of the progress of users search

The human computer interface is less well understood than other aspects of information retrieval, because humans are more complex than computer systems and user's motivations and behaviours are more difficult to measure and characterise.

Effective interfaces for information retrieval systems are a high priority for users of these systems. The interface is a major part of how a system is evaluated, and as the retrieval and routing algorithms become more complex to improve accuracy, more stress is placed on the design of interfaces that make the system easy to use and understandable.

There are two main aspects of the user interface of information retrieval systems:

- The query interface
- The result (answer to a query) interface

The basic query interface is a box where one or more words can be typed. Internet search engines also provide a query interface for complex queries as well as a command language using Boolean operators and other features, such as phrase search, proximity search, and wild cards.

The result usually consists of a list of relevant documents for example, for the Internet search engines the search result is a list of the top ranked Web pages, and typically the information includes the URL and few lines of the Web page content. The order of the list is typically by relevance, but there is not much public information about the specific ranking algorithms used by Internet search engines.

However, the user interface for an information retrieval system needs some new ideas about how to display large, abstract information spaces clearly. Until this happens, the role of user interface in information retrieval will probably be primarily confined to providing overviews of topic collections and displaying large category hierarchies.

2.5 Summary

This chapter focused on traditional information retrieval and interactive information exploration interfaces. Aside from this, information retrieval system has been described. It also suggests that in this research Oard and Marchionini's [1996] framework is used. It discussed the scope for user interface innovation to support interactive information exploitation activities.

In addition, methods usually used with Oard and Marchionini's framework such as *Vector* model and *Inference Networks* are briefly introduced. This allows this study to present a new model that is used for the *Collection* task of the framework in chapter 5. Details regarding the particular approach this research takes for the *Selection* task of the Oard and Marchionini framework will be discussed on chapter 6.

Thus this chapter reviews information retrieval systems, from the past 50 year e.g. library system before the Internet's arrival, after gathering all the information one can move on to the next stage of this research. Chapter 3 deals with the cutting edge electronic information retrieval techniques, thus the electronic age is elaborated by viewing variety of techniques to gather information for example like the Internet.

CHAPTER 3

3 Internet Information Retrieval

In this chapter the issues of retrieving information from the Web are discussed, along with a description of Web statistics and structures. The major mechanisms used at present to search the Web are summarised. This chapter is divided into the three main sections; technical aspects relating to information retrieval using the Web and in particular, two of the major retrieval architectures are presented in the first two main sections. The last section identifies the main issues in relation to Internet search problems which require addressing.

3.1 Information Retrieval from the Web

In order to search for a particular piece of information, some sort of data (documents, Web page, etc) has to be stored in a particular computerised medium, which is used to retrieve the information that a user has searched.

A starting point for a text examination process may be the complete document body, an abstract, the title or even just a list of words. Luhn [1957] indicated that the incidence of a word occurring in a document could give a useful indication of the document's subject matter. Vechtomova [2005] proposed that words occurring in the title, summary, in key positions within a paragraph or appearing several times within a paragraph are a useful indication of scope and nature of predictability effects, because there is a higher probability that terms in these positions indicate significant material on the topic. Vechtomova (op cit.) indicated that the word frequency measurement could be used for document scoring systems.

For example, if f symbolises the frequency of occurrences of a variety of words in a particular position of text and r is their rank order (the order of their frequency of occurrence) than a design relating f and r yields an arc analogous to a hyperbolic arc (see in figure 3.1). If the words, w, in a collection are ranked, r, by their frequency, f, they roughly fit the relation: $r * f = c$ (Zipf's Law [Bookstein *et al*, 2003]). Different collections still have a different constant c. In English text, c tends to be about $n / 10$, where n is the number of distinct words in the collection (see Zipf's Law).

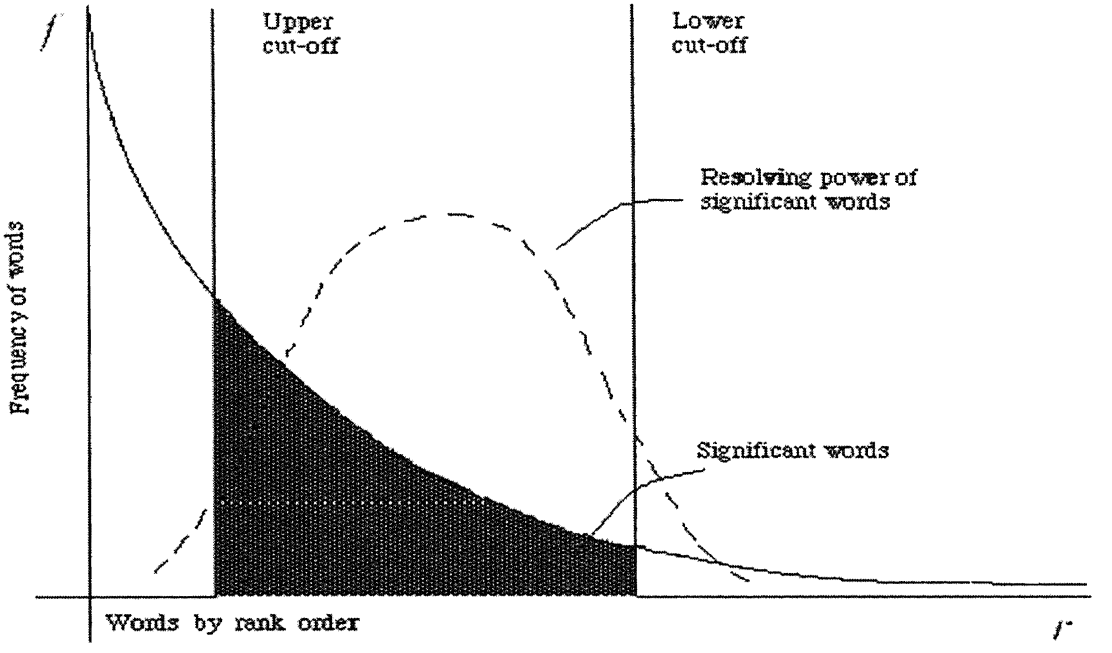


Figure 3.1 A curve plot relating the frequency f and the rank order of words r [Cameron, 2001, derived from C.J. Van Rijsbergan, *Information Retrieval*].

This describes Zipf's Law, which states that the product of the regularity of use of words and the rank order is roughly constant. Hence the arc is used to identify two cut-offs, the words above the upper cut-off are considered to be common to all documents (i.e. and, in, the, etc.). Those which are below the lower cut-off are expected to be uncommon, therefore do not contribute significantly to the substance of a particular document [Leroy *et al*, 2003].

The resolving power of major words, i.e. words that distinguish the particular contents of the genuine text, reach a height at a rank order position half way between the two cut-offs and from the peak fall off in either direction dropping to almost zero at the cut-off points. The cut-off points are estimated using a trial and error method, there are no given values.

Several previous studies have been undertaken on information retrieval techniques (for example Salton [1989]; Belkin and Croft [1992]; Belkin *et al* [2003]). However, these experiments used structured and reasonably harmonised collections such as sets of scientific papers or news stories on a related topic.

These collections are similar to the document sources described in the Text Retrieval Conference (**TREC**) run by NIST (National Institute of Standards & Technology) [Harman, 1993], which uses moderately small, well structured collections for preparing benchmarks. The TREC subset contains 10,000 Wall Street Journal articles extracted from the test collection used in the first Text Retrieval Conference, sponsored by the National Institute of Standards and Technology and the Advanced Research Projects Agency to develop a comprehensive test bed.

However, approaches that work well on TREC may not work well on the Web. The vector space model, described in section 2.2.1, which is one common model on TREC, calculates a vector for each document and a vector for the user's query, with the vector calculated from the word occurrence in the document and the query. The angle between the query and a document indicates the match between the two.

The vector space model tends to find very short documents based on the query document clustering, however the organisation of documents according to some criteria (such as semantic resemblance) may resolve some of these difficulties [Salton 1989; Liere and Tadepalli, 1996]. Another criticism of this previous research is that it has used document collections that are static (or nearly so) [Belkin *et al*, 2003]. The Web is rather different; and techniques that work well in a semi-static document collection may not be effective for the Web.

3.1.1 Dealing with the Information Overload on the Internet

With the growth of the Internet and other networked information, there is no problem in finding information, however, the problem still persists of finding the relevant information. The user's probability of finding the relevant information is getting more difficult as the data available on the Web is increasing with extraordinary speed [Tekla, 2001]. This fact can be recognised as the *information overload* problem, first explored by Bush [1945].

Bush [1945] found that an individual who was presented with a huge amount of information to process, which exceeds his/her physical or mental capabilities, could be seen as suffering from an information overload problem.

A similar concept, information anxiety, is the primary defining characteristic or result (symptom) of the information overload problem. Information anxiety results from user inability to access and extract meaning from the wide accumulation of information available to the user [Nelson, 1994].

Ljungberg and Sørensen [1998] point out that this information overload is a concept branching out from a database viewpoint of information technology. It focuses on situations where the quantity of information overwhelms the mental processing capability of the beneficiary. However it does not focus on communication methods.

Hence information overload is frequently represented by the problems associated with information retrieval in large databases. Thus, in order to decrease the risk of confronting information overload, the volume of information must be decreased, either by discovering very effective tools for information processing, e.g., information retrieval or filtering, or by increasing user cognitive capacity, by analysing the information more efficiently.

3.1.2 Difficulties Caused by the Nature of the Internet

There are two major sets of issues in information retrieval caused by the architecture and growth of the Internet. The first is caused by the data itself; the second concerns the means by which the Internet interacts with the user.

The first issue regarding the data available on the Internet has several problems. To summarise the more important concerns:

- **Diverse formats:** The information content of a Web page may be contained on a variety of different forms (i.e., PDF files, WAV files, JPG files etc.) and in different written languages. An effective information retrieval process would need to be able to extract the content reliably from all these formats.

- **Transient information:** Web pages can be added and removed effectively at will. An information retrieval process needs to track and be able to bring to the attention of a user relevant information that has been added and hide information that has been removed.

- **Reliability of information:** There is no control over what gets published on the Internet, and a proportion of the information available may be false or misleading, and the information retrieval process should provide some guidance to the user on the veracity of the information it presents.

- **Quantity of data:** The growth in the number of pages makes the information retrieval task harder, as more information needs to processed in response to a query.

- **Physical location of information:** The Internet has a distributed and dynamic architecture and the information retrieval process should be flexible enough to readily access disparate sources of data.

This research is investigating the second and last issue that is how to present the information to the user, so that they can identify what is relevant to them without being overwhelmed by overflowing information. And how could make best use of the Internet distributing architecture characteristic for retrieving relevant information and have decentralise Internet searching system, instead of using existing search engines that use a traditional old fashion centralised indexer architecture approach.

There are several dimensions to these issues; how to define the query so that the documents retrieved most closely matches the user's needs and how to present the retrieved information to the users so that they can extract the information they require. This is especially important when there is (potentially) a large number of Web pages that could be presented to the user after a single query.

3.1.3 Measuring the Web

"When you can determine what you are wont to express, and explain it in a numeric form then you have some knowledge about it. But when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of science." William Thomson (Lord Kelvin, 1907)[1]

This study presents some difficult questions concerning the Web's structure (see section 1.2.3), and attempts to provide some quantitative answers to them (see section 7.3). Therefore this section reviews some quantitative answers given in previous work on measuring the Web (such as: How big is the Web? What data formats are being used? What is the "average page" like?). It uses the numbers in these answers to drive some 3-D visualisations of growth.

[1] (Lord Kelvin, derived from Tim Bray, *Measuring the Web)* http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Thomson.html.

Measuring the size of the Internet is a difficult task, due to its inherent size and rapid growth. Most studies of the size of the Web have emphasised the number of hosts connected to the Internet as well as estimates of the amount of data on the Web (see www.netcraft.com, www.whois.net).

Netsizer (www.netsizer.com) estimates that there were approximately three million Web servers in 2001 (see Figure 3.2)[2]. However, according to Lyman et al [2003] the total amount of data available on the Web (static pages, dynamic page, e-mail, and instant messaging) was 532,897 Terabytes in 2002.

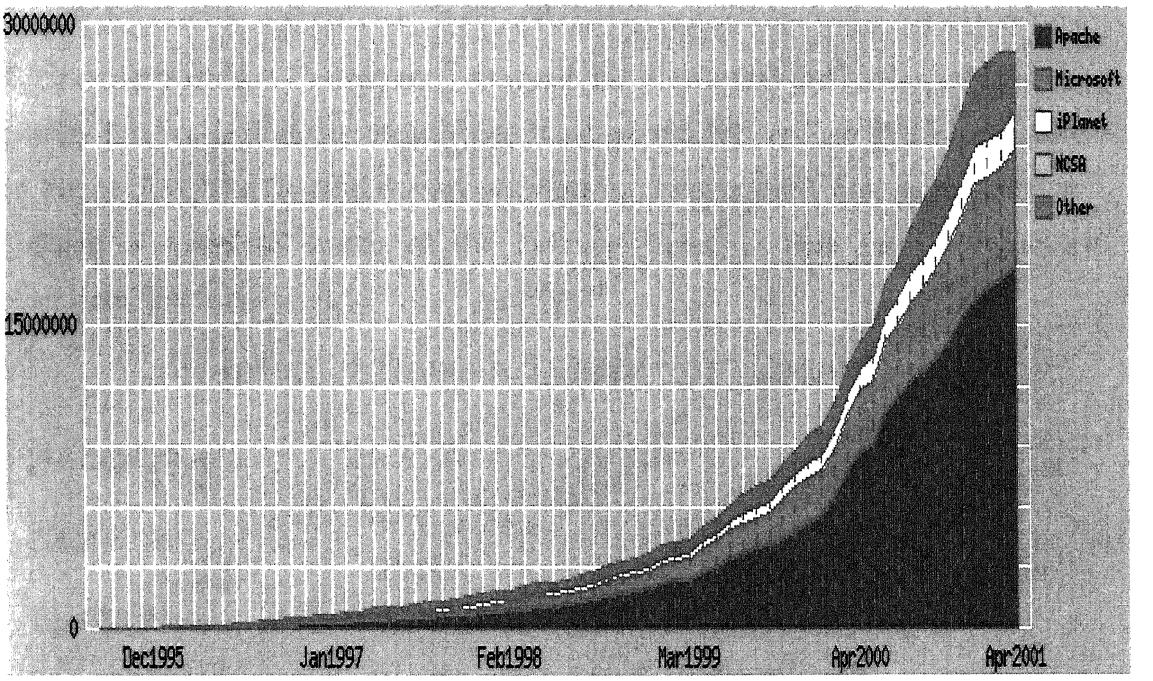


Figure 3.2 Web growth (www.netsizer.com).

The most popular format for Web documents is **HTML**, based on the Open Text (www.opentext.com) heuristic the analysis of Lyman et al [2003] states that over 87% of Web sites are written mainly in HTML.

[2] After 2002 the Netsizer (www.netsizer.com) was not a free Web site therefore the latest estimate is not available for this thesis

Shapiro and Varian [1998] estimated that the static HTML text on the Web was equivalent to about 1.5 million books. They compared this figure to the number of volumes in the University of California at Berkeley Library (8 million), and, noting that some of the Web's information can be considered "useful", concluded that "*the Web isn't all that impressive as an information resource*". Shapiro and Varian's assessment is incomplete since the Web incorporates digital resources of many varieties beyond plain text, often combined and re-combined into complex multi-media information objects. However, the Web is equivalent to large library collections and a significant portion of the information on the Web is text and text is the most popular resource.

ISC (Internet Software Consortium) research has shown that the number of Internet Domains had reached approximately two hundred and fifty million by January 2004 (see Figure 3.3). The Domain Survey attempts to discover every host on the Internet by doing a complete search of the Domain Name System. Internet Systems Consortium, Inc sponsors it.

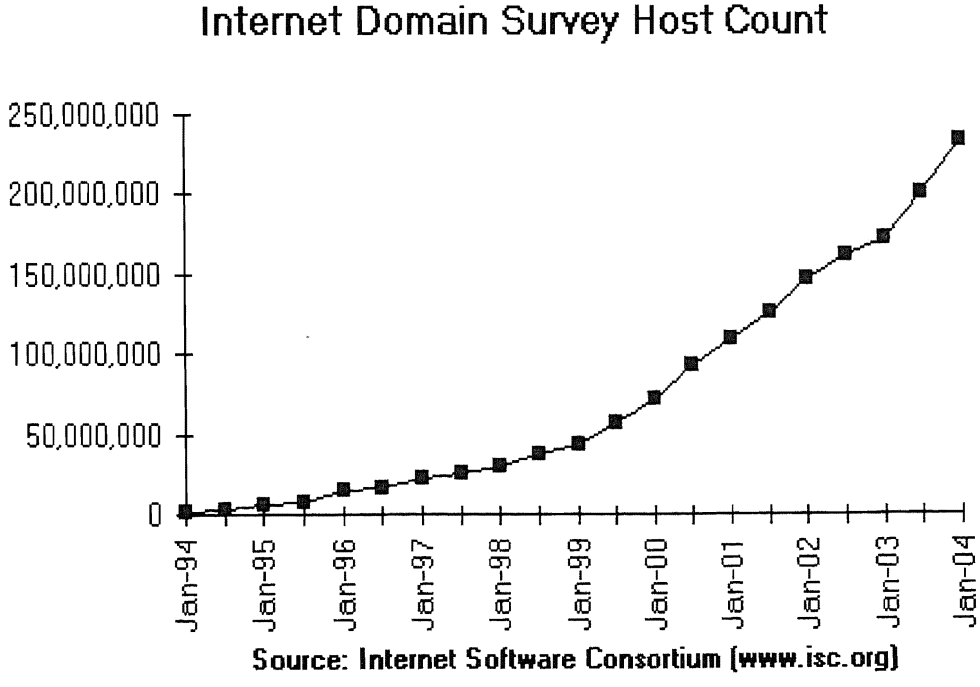


Figure 3.3 Internet Domain survey from ISC

O'Neill et al [2003] examined some key trends in the development of the Web and its size and growth:

- **Number of Web Sites**

A Web site is defined as a distinct location on the Internet identified by an IP address, and a Web page in response to an HTTP request for the root page. The Web site consists of all interlinked Web pages residing at the IP address.

Figure 3.4 shows the growth of Web sites based on data from the OCLC Office of Research Web Characterisation Project (wcp.oclc.org), an initiative that explores fundamental questions about the Web and its content through a series of Web samples conducted annually since 1998.

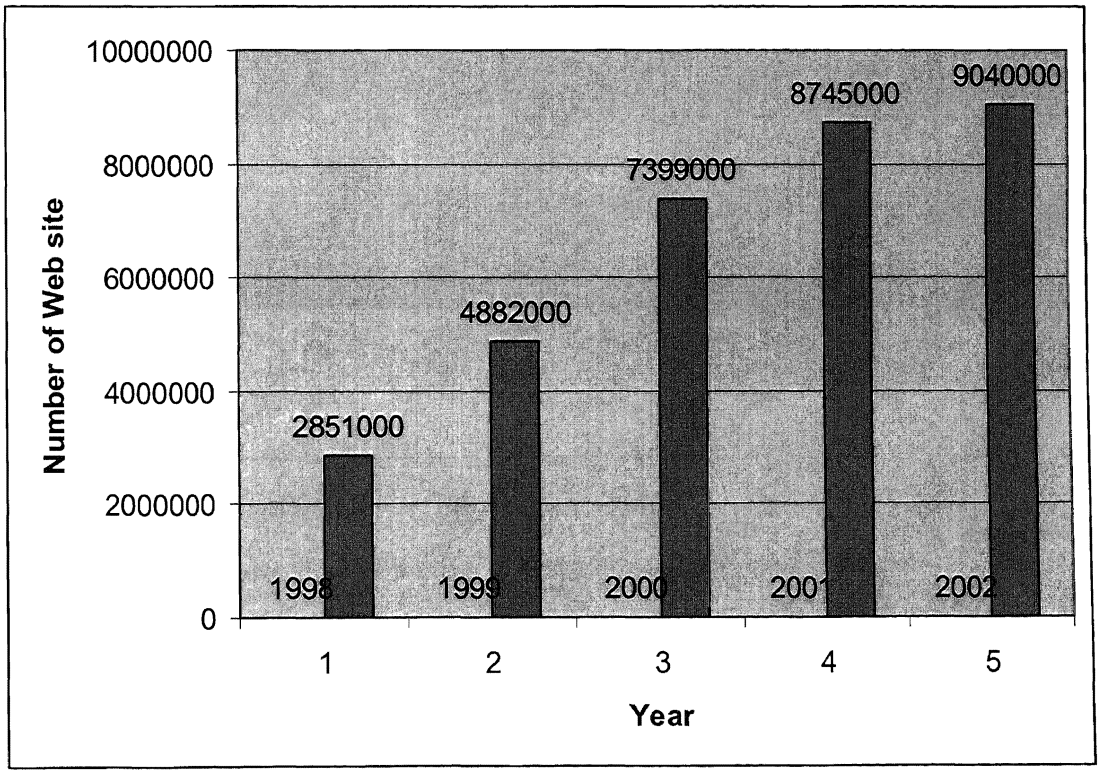


Figure 3.4 Growth of Web sites between 1998 to 2002 (data from wcp.oclc.org).

- **Number of Unique Web Sites**

Number of unique Web sites is the number of Web sites; adjusted to account for sites duplicated at multiple IP addresses. Figure 3.5 shows the growth of Unique Web sites based on data from the OCLC (according to the results of the Web Characterisation Project's survey) during 1998 to 2002.

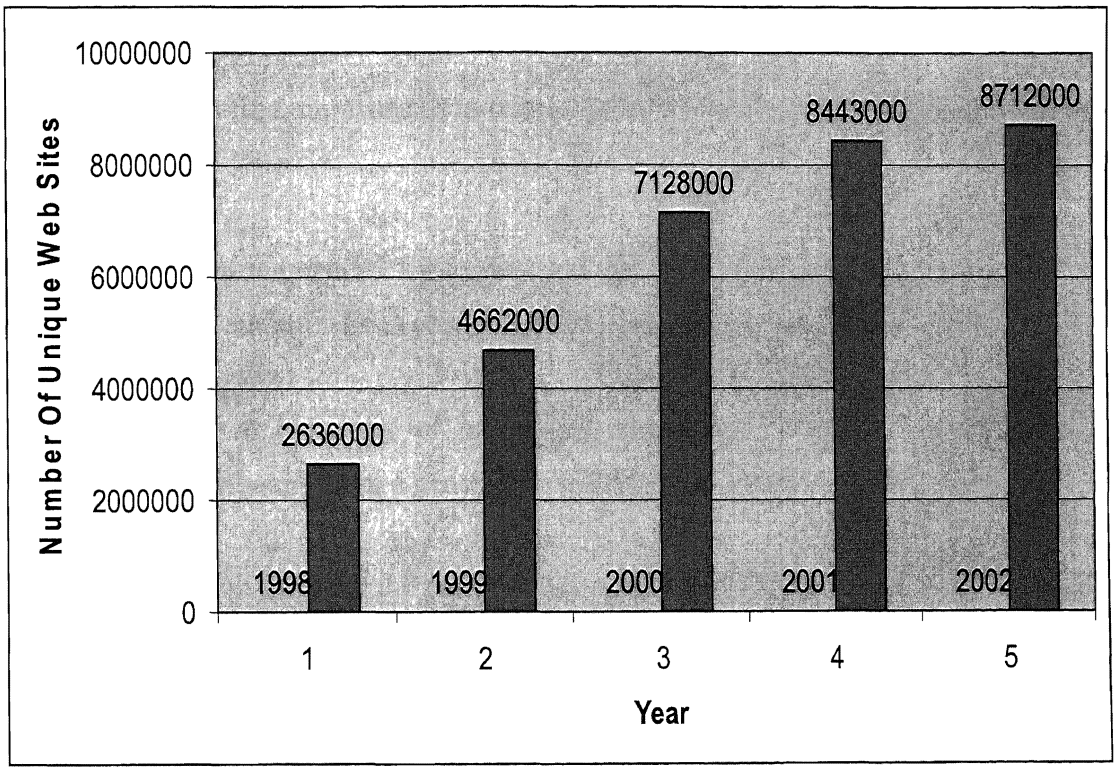


Figure 3.5 Growth of Unique Web site between 1998 to 2002 (data from wcp.oclc.org).

- **Web Site Types**

Unique Web sites are assigned to one of three categories:

- Public
- Private
- Provisional

Public: site provides free, unrestricted access to all or at least a significant portion of its content

Private: site's content is intended for a restricted audience; restriction can be explicit (e.g., fee payment or authorisation) or implicit (obvious from nature of content)

Provisional: site is in transitory or unfinished state (e.g., "under construction"), and/or offers content that is, from a general perspective, meaningless or trivial

Examination of growth rates for the period 1998 - 2002 reveals (measured in terms of the number of Web sites, see Figure 3.6, 3.7 and 3.8):

- between 1998 and 1999, the public Web expanded by more than 50 percent
- between 2000 and 2001, the growth rate had dropped to only 6 percent
- between 2001 and 2002, the public Web actually shrank slightly in size

O'Neill et al [2003] indicated during the five years observation that covered by the surveys, majority of the growth in the public Web occurred in the first three years of the survey (1998 - 2000). In 1998, the public Web was a little less than half its size in 2002; by 2000, however, it was about 96 percent of its size in 2002.

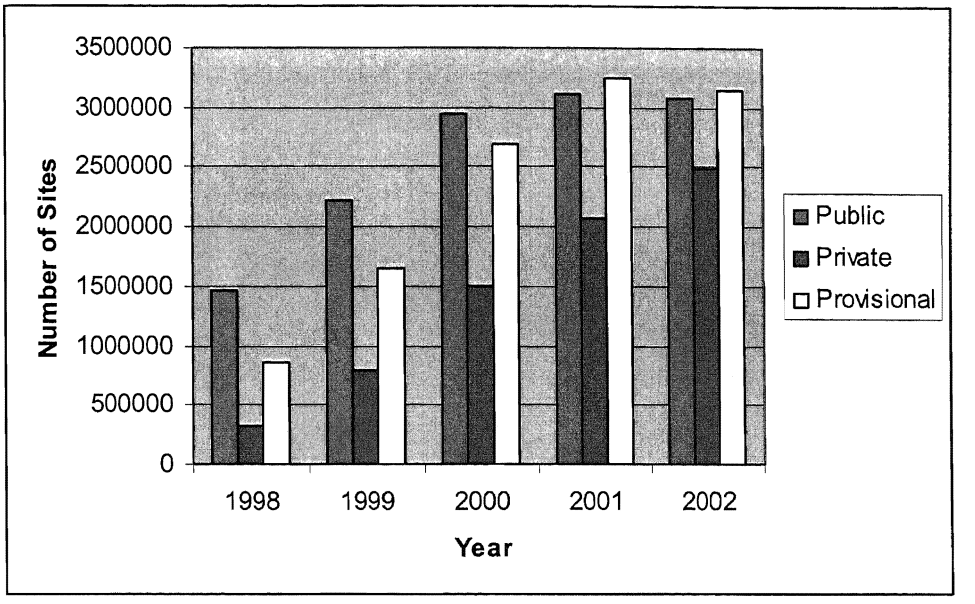


Figure 3.6 Size of different types of Web site between 1998 to 2002 (data from wcp.oclc.org).

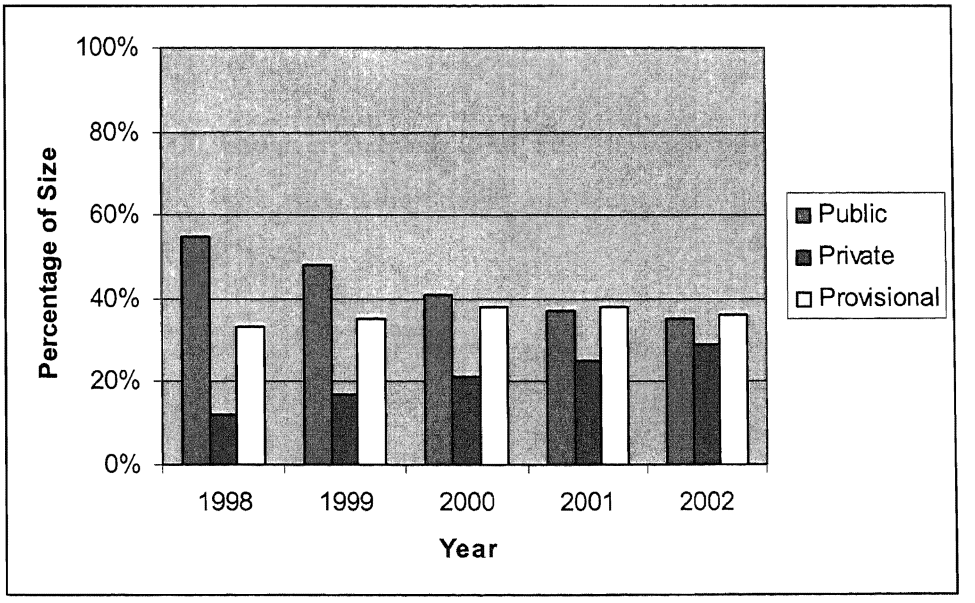


Figure 3.7 Percentage of Size of different types of Web site between 1998 to 2002 (data from wcp.oclc.org).

The results of the Web Characterisation Project's survey illustrate that the public Web is slowing down in the growth dramatically. The public Web demonstrated a net growth of 772,000 sites between 1998 and 1999; a similar number (713,000) were added between 1999 and 2000.

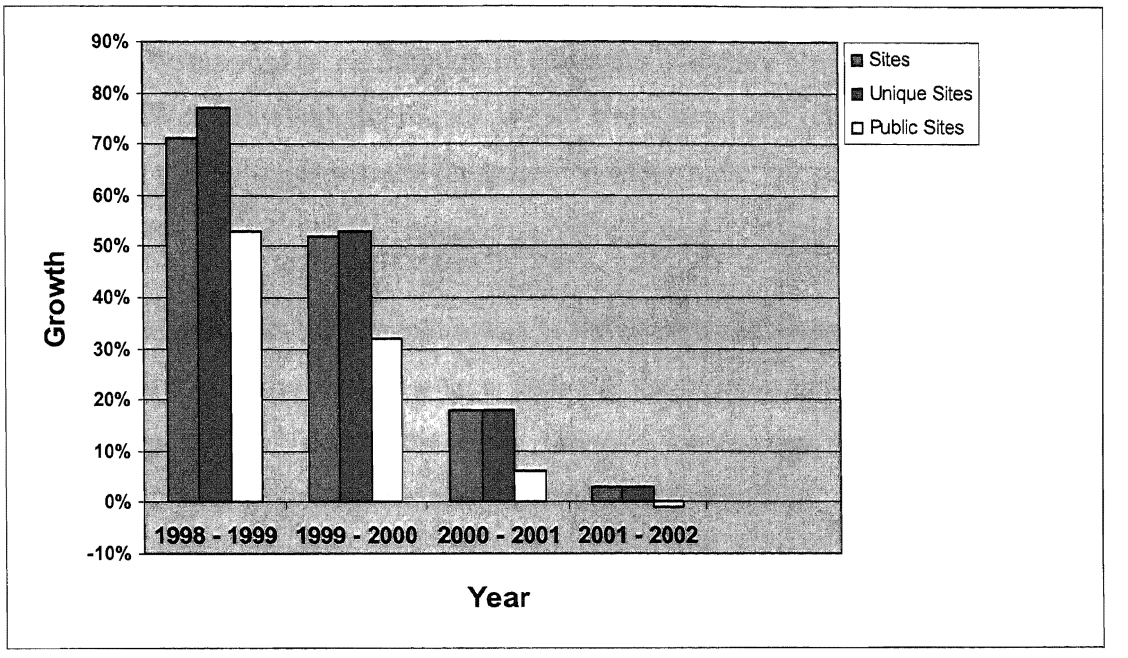


Figure 3.8 Change in Growth rates of Web sites between 1998 to 2002 (data from wcp.oclc.org).

After 1999 and 2000, however, between 2000 and 2001 the growth dropped off, only 177,000 new public Web sites were added, and, the public Web shrank by 39,000 sites between 2001 and 2002.

Bent *et al* [2004] concludes using evidence from the Web Characterisation Project's (WCP) survey that the public Web's growth has stagnated, if not ceased altogether. But the apparent shrinking of the public Web does not necessarily mean that less content is available. Results from the WCP survey suggest that while the number of sites may have plateaued, the size of Web sites is increasing. For example, the average number of Web pages contained within a public site in 2002 was 441, compared to 413 in 2001. The amount of information in databases and other formats not accessible by traditional Web-crawling techniques is said to be large and growing. The use of virtual hosting technologies permits the grouping of multiple "virtual sites" at a single Internet location.

One possible reason that could explain this is simply that the Web is not a new thing to users. Those users who wish to own a Web site more likely already have one. The demand to "get online" during the early years of the Internet has possibly been substituted with an urge to improve and expand existing Web sites.

The conclusion of these surveys indicate the following trends:

- The Web is an information collection of significant proportions, exhibiting a remarkable pattern of growth in its short history (and a significant portion of the information on the Web is text).

- Evidence suggests that growth in the public Web, measured by the number of Web sites, is reaching a slow level.

- The Web has been positioned as a global information resource, but analysis indicates that the Web is dominated by content supplied by entities originating in the US and the vast majority of the textual portion of this content is in English.

3.2 Search Systems on the Web

The Web consists of the surface Web (fixed Web pages) and the dynamic Web (the database driven Web sites that create Web pages on demand). The amount of textual data available on the Web were estimated to be in order of 167 Terabytes for the surface Web and 91,850 Terabytes for the dynamic Web [Lyman *et al*, 2003]. Thus, the Web can be seen as a very large, unstructured but extensive database. This triggers the need for efficient tools to manage, retrieve, and filter information from this database. In this section discussion includes searching and retrieving for the Web search system.

3.2.1 Architectures of Retrieval Systems for the Web

The major difference between standard IR systems and the Web is that, in the Web, all queries must be responded to without accessing the Web pages. Otherwise this would involve either storing a copy of the Web pages near by (which ends taking over too much space), or accessing remote pages through the network (which may be a very slow process). This difference has an effect on the indexing and searching algorithms, as well as the query languages.

Search engines use a centralised architecture involving Crawler software agents. These agents negotiate the Web sending new Web pages to the main server where they are indexed.

The crawler architecture has several variants the most important of which are:

- Crawler-indexer architecture
- Distributed Architecture (Harvest)

3.2.2 Crawler-indexer Architecture

The majority of the search engines use central crawler-indexer architecture. Crawlers are programs (software agents) that traverse the Web sending new or updated pages to the main server where they are indexed.

The aim of a general-purpose web search engine like Alta Vista or Google is to index huge numbers of pages from the Web. Figure 3.9 shows the structural design of a classic crawler-indexer, using the Alta Vista crawler as an example [Schneider, 2004].

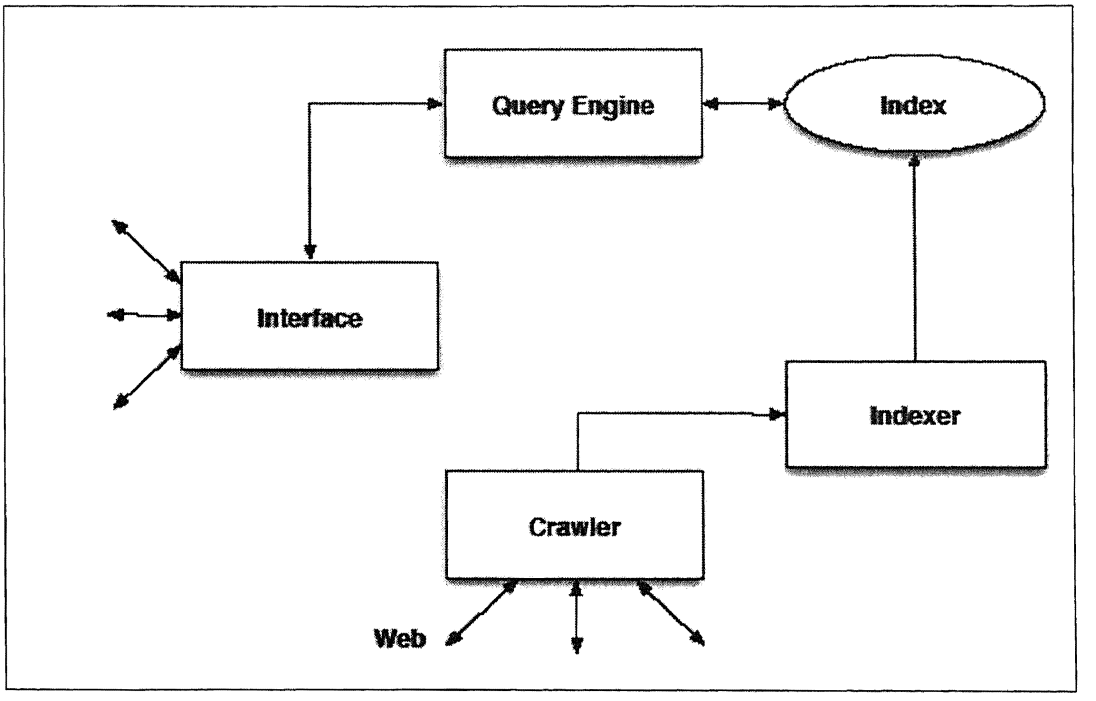


Figure 3.9 Typical crawler indexer architecture copied from (Schneider (*op cit.*)).

The classic crawler-indexer has two parts, one that deals with the end users, consisting of the user interface and the query engine and another that consists of the crawler and indexer modules.

One major problem faced by the software is the assembly of the data, this problem is caused by the dynamic nature of the Web, the configuration of the communication links and the high load on Web servers. In addition the volume of the data presents problems.

The largest Web search engines available for Web coverage, according to 31 December 2002 statistics, were Google, Alta Vista, Hotbot, and MSN, in that order [Greg, 2003]. Thus, according to him these search engines claim up to 3 billion of Web pages have been indexed.

Most search engines are based in the United States and focus on documents in English. Nevertheless, there are search engines which specialise in different countries and other languages, for instance, to query and retrieve document written in Chinese, French, and Dutch. There are also search engines aimed at specific topics, for example the Search Broker (debussy.cs.arizona.edu) which allows searches in many specific topics, another one is DejaNews, which searches the News Group archives.

According to Greg [2004a] by November. 2002, Google moved their claim up to 3 billion, and estimate the size of the Web pages Google indexed by February. 2004 to be 4 billion.

Greg [2004b] claims that Alta Vista indexed over 2 billion Web pages by 2003 and after that Alta Vista Switched to Yahoo database in March 2004. He showed that the Excite (WebCrawler) indexed less than 1 billion of Web pages before becoming defunct as a separate database on 2002. It now uses InfoSpace. Greg also indicates that the Yahoo directory had over 1.7 million records in October 2000. The last official word from a Yahoo representative was in November 1997 when they claimed Yahoo! contained over 730,000. Considering this, he claims that the Yahoo directory probably has over 3 million records as of 2003.

3.2.3 Harvest Architecture

The second type of crawler architecture is the Distributed Architecture. Liu et al [2002] stated that Harvest (an example of this architecture) is a system to collect information and make it searchable using a Web interface. Harvest can collect information from the internet using HTTP, FTP, and local files, and supported formats include HTML, PS, full-text, mail, news, WordPerfect, and many more. Adding support for a new format is easy due to Harvest's modular design. Harvest is the most common example of Distributed Architecture; typical Harvest architecture is illustrated in Figure 3.10.

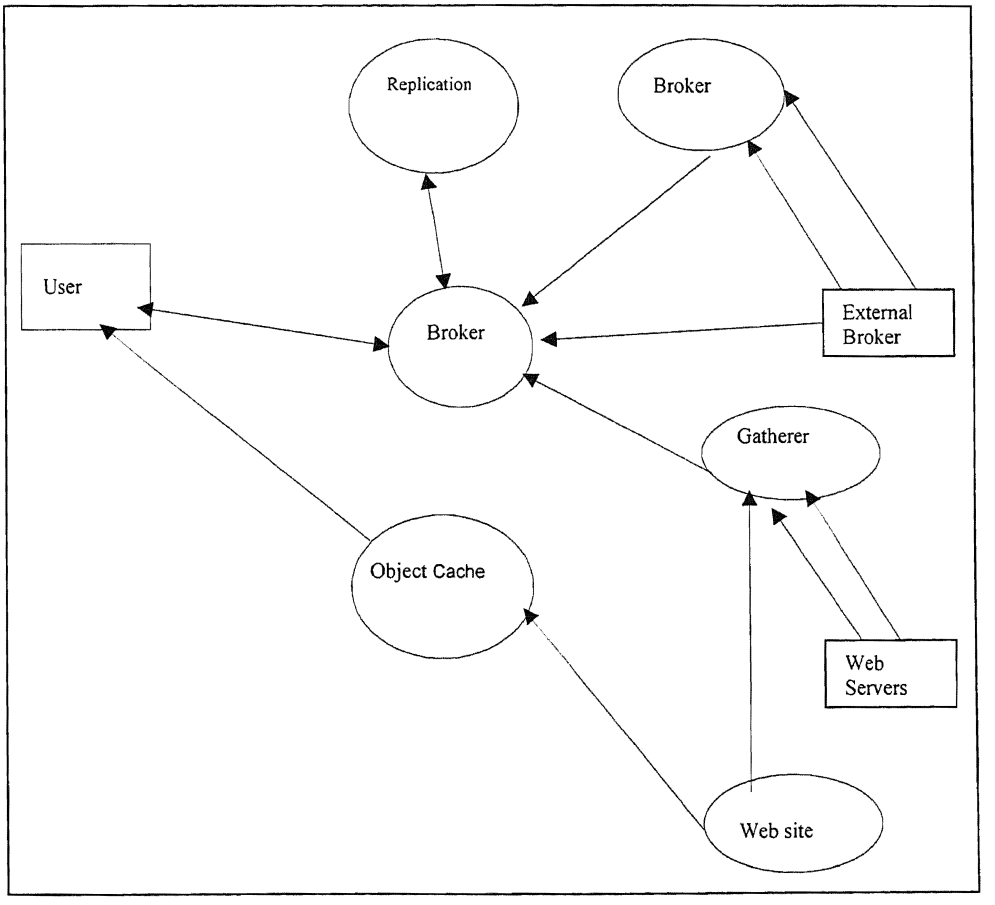


Figure 3.10 Harvest architecture, the second type of crawler architecture

Harvest is more efficient than the crawler architecture, but because Harvest requires the co-ordination of several Web servers that gives it a disadvantage in terms of complexity compared to the crawler.

Harvest consists of two main elements:

- **Gatherers**: a gatherer collects and pulls out indexing information from one or numerous Web servers. Gathering is periodic and is defined by the system.

- **Brokers**: a broker provides the indexing apparatus and a query interface to the data collected. Brokers recover information from one or more gatherers or other brokers, while adding incrementally to their indices.

The Harvest distributed approach addresses some of the problems of the crawler indexer, such as:

- Web servers receive requests from different crawlers, increasing their load.

- Web traffic increases because crawlers retrieve entire objects, but most of their content is discarded.

- Information is gathered independently by each crawler, without co-ordination between all the search engines.

Depending on the organisation of the *gatherers* and *brokers*, various improvements on server load as well as the network traffic can be accomplished.

The data collection process used by popular search engines such as Google, Yahoo, Alta Vista and Webcrawler is a centralised *crawler-indexer* architecture. Because this study uses the results of these search engines (see section 5.3.1) as a basis for a sample of the web, fetching the web sites that they return, the foundation for data collection is *crawler-indexer* architecture in this study.

3.3 Overview of Internet Search Problems

According to Bhowmick et al [2003], it has become clear that search engines are less than ideal for retrieving an ever-growing body of information on the Web; the major problems with the query interfaces currently provided by search engines are discussed below:

- One of the main reasons why search engines generally return results containing a lot of irrelevant documents is the ambiguity of terms used in the user query, partly because they may not be aware of the logical view of the text used by the system. For example, a search for a word like 'Bank' will lose part of its semantic information in the search for 'bank', when the search engine used is not case sensitive.

- The usage of advanced Web-query operators such as 'and' and 'or' differ between Boolean logic and natural language, with the natural language often used in queries being the less sophisticated. For example, when choosing between two things in natural language an 'or' is being used which does not match the Boolean interpretation.

- On the Web, the difference between a single-word query and a two-word exact phrase match can mean the difference between an unmanageable mess of retrieved documents and a short list with mainly relevant documents.

- Users often view only the first page of results of a Web search engine. This may indicate that the answer is often too large and may not help the user discriminate against irrelevant and out of date results.

A search engine is usually selected by the user on the basis of its user-friendly features like simplicity, coverage and relevance of the answer.

3.3.1 Some Issues in Web Searching Systems

At the present moment there are a wide range of techniques which can open up new and unique research areas in Web searching. The major issues are listed below:

- Querying: more work on merging the structure and the content is needed to improve queries [Chang and Ding, 2004].
- Indexing: the implementation issues for indexing algorithms for decreasing storage overheads and I/O times must be improved. How to achieve fast searching applications of indexing systems that involve real-time and multi-user constraints requires more work on concurrency control, update, and recovery strategies [Park et al, 2002].
- Ranking: achieving a better ranking is essential; for example, by using a method that can automatically generate ranking strategies for different contexts and tailors it to different users' needs [Kazai et al, 2004].

The observations of Jansen and Spink's [2005] point to the need for further research to characterise search engine performance, analyse its sensitivity to subject areas and to determine the significance of bias under specified search conditions. However, other techniques such as following the hyperlink structure have also been utilised, as this structure can contain useful information [Ng et al, 2001].

3.3.2 Hyperlinks

A Hyperlink is an image or portion of text on a Web page that is linked to another Web page, either on the same site or in another Web site. Clicking on the link will take the user to another Web page or to another place on the same page. Words or phrases, which serve as links, are underlined, or appear in a different colour, or both. Images that serve as links have a border around them, or they change the cursor to a little hand as it passes over them. Although traditional information retrieval has informed the approach for Internet searching Chakrabarti [2003] indicated that searching techniques from traditional IR are not sufficient in themselves for Web retrieval.

Traditional IR approaches do not take advantage of other Web features such as hyperlinks. Today, documents on the Web make much more discriminating use of hyperlink connections; an author of a Web page usually generates hyperlinks therefore these links tend to be labelled in a meaningful manner by their surrounding context.

As the name of the World Wide Web implies, the Web pages are heavily interconnected. According to Fisher and Everson [2003], hyperlinks can be viewed as implicit annotations that encode human judgments about relevance. Confronted with the thousands of documents that might be returned by a general query, the link information can be used to extract a much smaller number of "authoritative" sources on the topic. These pages are likely to be of greater utility to the user than a "ranked" list of hundreds of nearly indistinguishable documents. Hyperlink potential is explored in this study (see chapter 5 section 5.1.2).

Some researchers [Fieldsend *et al*, 2004], as well as commercial search services (e.g. DirectHit, Alexa) have also looked at ways to leverage Web user statistics (i.e. counting link clicks). Using hyperlinks can provide additional information that may be used for enhancing Web search systems and lead to an efficient retrieval system. Accordingly, in IR the link-based retrieval strategies (also known as hypertext) investigate different methods of enriching local document content with the content of connected documents.

Balling [2003] indicated that the context of hypertext in IR is a model of information retrieval based on representing document relationships as edges of a generic graph in which the documents are the nodes, and that hypertext structure based methods have proved to be a powerful way of exploring the relationships between documents. However, in the Web where hyperlinks connect Web pages with differing content, Web page connections via hyperlinks can sometimes lead to poor retrieval performance; for example, when hyperlinks may be used for commercial advertising.

One way to address such problems would be to categorise hyperlinks according to their purpose and usefulness so that they may be utilised appropriately. But considering the inhomogeneous characteristics of the Web, categorising hyperlinks may be unrealistic. One difference between traditional information retrieval and Web information retrieval is the Web's hyperlink structure, which can be based on any hyperlink feature, such as path keywords, protocols and host names [Mostafa, 2005].

Google's founders formulated a search algorithm based on web page overall link popularity called *PageRank* [Wang et al, 2003]. Essentially, the more incoming links a web page has, the better. However, it is more complicated than that. Langville and Meyer [2005] indicated that *PageRank* actually gives an importance score for each webpage, not a relevancy score. Another implementation issue concerns the accuracy of *PageRank* computations: the accuracy with which Google works is not known. Langville and Meyer quote that "much work, thought, and heuristics must be applied by Google engineers to determine the relevancy score, otherwise, no matter how good *PageRank* is, the ranked list returned to the user is of little value if the pages are off-topic". It seems that there are various patterns in hyperlinks, but whether these patterns can be successfully identified without human judgement remains to be seen; this is one of the research questions of this thesis (see section 1.2.2).

3.4 Summary

This chapter focuses on information retrieval on the Web and the major mechanisms used to search the Web. Also some quantitative answers given in previous work in measuring the Web are reviewed. A significant portion of the information on the Web is text and the vast majority of the textual content is in English. This chapter identified Internet search problems and techniques, which can open up additional study areas in Web searching. The incorporation of hyperlinks can provide additional information, which may be applied, to improve IR systems on the Web. The next chapter (chapter 4) focuses on the most common information retrieval system used for Internet searching, the search engine.

CHAPTER 4

4 Contemporary Web Search Techniques

Web searching is the task of finding information present on the Internet relevant to a particular query. This chapter highlights issues with regard to the Internet searching technique. This chapter addresses the approach for retrieving information from the Internet. It focuses on Internet search engines and how Internet search engines are operated.

The contents of this chapter can be summarised in three main sections, in the first section attention is given to a definition of Internet searching and an introduction of two main approaches for retrieving information from the Internet.

This work looks at a number of related issues in the problem of searching the text contained in the web. For this reason, the second section of this chapter looks at how Internet search engines are used to tackle the searching processes and the last section is concerned with how all the web pages can be located in an efficient manner.

4.1 The Internet Search Definition

Internet information retrieval is quite similar to general information retrieval, which can be defined as: given a set of documents and a query, determine the subset of documents relevant to the query. Hence the Internet search definition is the issue of finding the set of documents on the Internet relevant to a given user query.

An Internet searching system accepts as input a set of Web pages and a query. Note that this set of Web pages is not necessarily the entire Web. The Internet searching system output is a subset of all Web pages such that every page is relevant to the query. Like Information Retrieval, Internet Searching is difficult to resolve because it is difficult to define a searching formula or approach and thus must be approximated. The Internet Searching problem is further compounded because there is no direct way to obtain all Web pages. Therefore, only subsets of available Web pages are given as input. Thus, relevant documents must be located from the given set of Web pages.

There are many techniques that have been developed to address Internet Searching issues, and the two most popular are: Web Browser Agent and Search Engine

4.1.1 Web Browsing Agents

In order to tackle Internet searching demand a wide variety of procedures and techniques have been developed. The simplest approach is manual browsing. But this is impractical as the web is enormous and is unstructured and unorganised. Hence automatic browsing is a better solution.

For dynamic web browsing, programs which browse to find appropriate information are similar to sequential text searching. In these programs the web is treated as a graph, in which the documents symbolize nodes and the hyperlink symbolize the edges. The task is to discover relevant information by following the links. Hence, with this technique the user is searching the current structure of the web, rather than what is stored in the index of the search engines, thus making this approach slower.

De Bra and Post's [1994] *Fish-Search* was the first heuristic devised which explored automatic Web browsing by enhancing the Mosaic Web browser to automatically browse starting from the current search page and continuing to a certain depth. When a user requested a search, *Fish-Search* would conduct an exhaustive depth-first search from the current page. This search can be restricted to either a certain period of time or until an assured number of applicable pages have been retrieved.

Even though a number of heuristics have been provided by *Fish-Search* in order to maximise the search [Houben *et al*, 1994], users observed that *Fish-Search* was not very different from an extensive search. So if all users were to utilise a *Fish-Search* for every exploration, the overall stress load would impose a significant load on Web servers.

An analogous program to *Fish-Search* was designed by Lieberman [1995], called *Letizia*. *Letizia* is a user interface program that monitors user behaviour and attempts to locate useful objects. It does not use any speculative criteria to direct its search, but rather uses resource constraints, such as exploring and retrieving only a few numbers of pages per minute, in order to restrict its search. Even though *Letizia* adheres to more resource constraints than *Fish-search*, it has the potential of using more resources.

In a similar manner to *Fish-search*, *Letizia* conducts a depth-first search through pages that the user is currently browsing. Hence *Letizia* continuously performs and refines its search while consuming all the resources possible and only returns the end result according to the demands of the end user.

Another common technique is LaMacchia's *Internet Fish* (or *IFish*) Construction Kit [LaMacchia, 1997]. Users can generally construct *Internet Fish* using the tools available form the construction kit. *IFish* is a kind of agent that automatically browses the web searching for appropriate information.

As compared to *Fish-Search* [Liberman, 1995], *IFish* uses a relaxed model where appropriate information is shown to the end user when it is found. As for the *Fish-Search* it attempts to find all the appropriate information at one go. Hence in the *IFish* model there are no time constraints on how fast the information is retrieved for the end user, as well as no constraints on how long any search process can take.

The advantage of the *IFish* model over *Fish-Search* is that it does not impose a huge load on the web servers. Users of the *IFish* can be encouraged to use multiple *IFish* searches for their required information needs. However if *IFish* was to be used very frequently with multiple searches at the same time, this leads to an enormous load on the network resources.

How Web agents deal with relevancy is a useful feature for them. Users may not be able to efficiently define the criteria of a document, which is relevant to a given query; hence agents usually provide a very expressive language for users to define what they mean by relevant.

For example, on a given query, one user may find a particular document relevant, and another may find it irrelevant. If both users used their own agents, they could each define their relevancy criteria for their own agent, and thus only the user who found the document relevant would see it.

The main disadvantage of using web agents is that they consume enormous amounts of resources. A few well- designed agents do not pose a large problem for the web. However in great numbers even well designed agents can overwork a server. Even if overloading a server with requests was not a problem, the entire families of automatic browsing agents also suffer from one of two problems relating to the scale of the Web.

One of the problems associated with automatic browsing agents is that they require a relatively predictable path of web pages to retrieve information that may be relevant; i.e. if there is no path leading towards the relevant information, then the agents will not be able to locate that information.

4.1.2 Information Retrieval Using Search Engines

One of the common approaches for Internet searching is to make use of the generic search engines e.g. Alta Vista, Google, and Yahoo. Information on the Web for these search engines is located in a body or set of documents, via a two step process:

- The first phase is indexing, which converts the Web page into a type of an index that maps words to documents.

- The second phase is the retrieval phase, in which the query is used to search for documents.

The program that prepares the index is called the *Indexer* and the program, which facilitates the retrieval via the index is the *Search Engine*. Using a set of Web documents as the corpus search engines can locate information on the Internet. So after this process the documents can be indexed and searched like any other body of documentation.

In order to explain how the indexing and retrieval process works, the following sections give a summary of how web documents are indexed using a straightforward inverted index, as well as how those documents are then retrieved. In the field of information retrieval, indexing and retrieval are two basic research areas [Deerwester *et al*, 1990].

4.2 Internet Search Engine

In the popular Internet culture, "*Search Engine*" is the term generally used to describe a tool for retrieving information from the Internet. However, "*Search Engine*" means something different to most information retrieval (IR) researchers as it has traditionally referred to the programs that do the actual matching of query terms to a database in an IR system. Therefore, some people are uncomfortable applying "*Search Engine*" to Internet search tools such as Yahoo.

However, in this study it is proposed that the use of the term "*Search Engine*" is applied to all Internet search tools since it is a favourable name among most Internet users and is clear and recognisable. Most search engine are made up of three main components:

- Web pages, which are mostly HTML format
- A Database and Indexing of HTML documents
- A Retrieval system

After discussing these three components the issue of the collection of the Web pages is addressed.

4.2.1 Web Pages and HTML

The Web largely consists of text files formatted by the Hypertext Mark-up Language (HTML.). HTML uses mark up tags within in which text is enclosed like <a,b,c>. These tags usually appear in pairs around a region of text, with the opening tag of the form'<word'arg>' and the ending tag have a sign of a greater then sign '</word>'(but not all mark up tags require a closing tag).

Using a tag called an anchor tag can create a hyperlink. A user can select a hyperlink by clicking on the particular hyperlink, which combines a Uniform Resource Locator (URL) with a particular piece of text. Hence when the user selects the text containing the hyperlink, the web browser loads the URL which is embedded in to the hyperlink. An example of an HTML document is represented in Figure 4.1.

It is a standard that every HTML document contains a title. The title is embedded in the tags <title> and </title>. For example this can be seen in the Figure 4.1 "Sample Document". One of the other functions of HTML is the use of Meta tags. There are two important types of Meta tag:

- The key word tag
- The description tag.

A comma separates key words and phrases of the document from the main word Meta tag; aside from this the description tag is embedded with a short description of the page. So by the help of these two tags indexing of the pages is made possible.

```
<html>
<head>
<title>A Sample Document</title>
<meta name="keywords" content="sample document, html">
<meta name="description" content="Just a sample document">
</head>
<body>
<h1>This is a large header</h1>
This is the text of a sample HTML document.
<a href="http://www.cnn.com">This is a hyperlink to CNN</a>
</body>
</html>
```

Figure 4.1 A sample HTML document.

4.2.2 Indexing of Web Pages

There are many options available for today's Internet searcher. Search engines offer the ultimate in indexing by completely crawling through Web sites and compiling full-text databases.

Before the indexing process starts the HTML documents must be cleaned of any irrelevant data. The cleaning of irrelevant data refers to the removal of the document's header tags and mark-up tags in the body of the text. However in some cases the data contained within the tags is used in the indexing process, the data used can be a title or keywords as well as the description existing in the body of the text.

Chowdhury [2004] explains that indexing is the intellectual analysis of the subject matter of a document to identify the concepts represented in the document and the allocation of descriptors to allow these concepts to be retrieved; it can be used to speed up the search. Indexing a large number of documents can be done semi-automatically using software applications. IR systems represent a document collection with what is called an inverted index, which is usually composed of two elements:

- The vocabulary
- The occurrences

The vocabulary is the set of all different words in the text. For each such word a list of all the positions where the word appears is stored. The set of all those lists is called the "occurrences". These positions can refer to words or characters. Word positions simplify phrase and nearness queries, while character positions facilitate direct access to the matching text positions. By the use of mapping between each term in a document's body to the documents that contain that word an inverted index can be created.

4.2.3 Retrieval

In order to retrieve information the search engine can accept a query consisting of a list of keywords. The retrieval process is carried out by locating keywords with the help of an inverted index, which retrieves a set of documents for each keyword typed in. After the retrieval process the sets of documents are then fused together. The fusing of these documents depends on the query semantics of each particular engine. For example a Boolean engine requires the query to be formulated using Boolean logic.

The search engine will sort through the millions of pages it has indexed and present the Web page addresses that match the user's topic. The matches will be ranked, so that the most relevant ones come first. One can ask how do search engines go about determining relevancy, when confronted with millions of web pages to sort through? They follow a (search engine specific) algorithm. Exactly how a particular search engine's algorithm works is a closely kept trade secret. However, all major search engines are believed to follow some general rules. Boydell et al [2005] indicated that IR techniques based on keyword density (word frequency) are used to provide a set of complementary options that summarise a Web page and enable a rapid decision about its usefulness.

Search engines also check to see if the search keywords appear near the top of a web page, such as in the headline or in the first few paragraphs of text. They assume that any page relevant to the topic will mention those words right from the beginning.

Frequency is the other major factor in how search engines determine relevancy. A search engine will analyse how often keywords appear in relation to other words in a web page. Those with a higher frequency are often deemed more relevant than other web pages [Sullivan, 2003].

4.3 The Web Document Collection Issue

Using a traditional Information Retrieval approach, for an Internet Information Retrieval system has its own difficulty. The indexer (see section 4.1.2) in an Internet IR system is a program that prepares the index for search engines. An indexer needs to have access to the Web pages in order to index them.

However, there is no direct way to obtain all of the available documents on the Internet. Hence the web document collection process has the issue of getting all the available information from the Internet.

To find Web pages, a system takes as input a starting set of Web documents and a method of obtaining Web pages through hyperlinks, such as:

- Collective user histories
- Distributed system
- Spiders or Web Crawler

These return, as output, the set of all available Web pages. Assuming that all the Web pages have been found, a traditional Information Retrieval approach can then be used.

4.3.1 Collective User Histories

This approach relies on collecting the browsing history of users. An early example of this is Lim's Coollist system by Jong-Guyn [1994], where a Coollist Repository is used to store the histories of users' browsing at the end of each browsing session. A session and in the Coollist system is defined by a fixed amount of idle time. The histories from the Coollist Repository subsequently are merged into one index by the Coollist Library.

The Coollist Library is capable of creating other indices, which includes users of certain groups, which in turn can possibly create a comprehensive index of the Web. It is reasonable to assume that a Web page will be revisited by its owner at least once to check that there are not errors in the HTML. Thus, all pages would be part of some user's Coollist, and thus would be added to a global index.

However, with all these advantages the approach is still not practical due to several non-technical issues. One problem is privacy, where some users are not willing to declare their browsing histories, or in some cases they're legally bound from declaring this history (for example, browsers from a corporate site). The second problem is related to the client program needing to send histories to a repository. It is not possible for all users to have this kind of program unless it is set in the browser or the operating system.

4.3.2 Distributed System

An alternative solution to the Web document collection issue is the local indexing of pages, known as *Distributed System* [Yates and Neto, 1999]. The Web server creates an index of local pages. Two possibilities exist here. The index can be transferred to a central repository to create a global index.

Alternatively, the Web server transfers routing data to a central repository, and then the relevant queries are forwarded to the local server. According to the Harvest model (see section 3.2.3), a global index is created through merged indices in a central repository [Bowman et al, 1994]. This implies that all Web servers that create local indices use the same indexer, and the comprehensive index would then be the means to search the whole Web (see Figure 3.10). The convenience of this model is that resources for retrieval can be integrated in one central location, which makes the search service more convenient to operate.

While a *Distributed System* appears to be a straightforward method to obtain a comprehensive Web index, it is a technology that is difficult to use. A *Distributed System* requires that Web server operators index their site using a particular indexer, this puts a substantial load on the operator.

Losee and Church [2004] indicated that many challenges remain in the area of distributed information retrieval. One challenge is focusing on new methods and algorithms to efficiently and effectively access data distributed over large heterogeneous distributed systems. Other challenges are to measure retrieval effectiveness on a large text collections (such as the Internet) and building distributed IR systems from heterogeneous components.

The advantage of a centralised search system is that it avoids many of the problems to be overcome of a decentralised search system. However it doesn't make logical sense to structure all information in a single hierarchy because the nature of information is very interconnected with no center. Therefore investigating a decentralised search system has considerable potential worthwhile studying the problem.

the advantages of that make it worthwhile to overcome the problems, involve scaling.

Using a decentralised approach should gain much greater processing power with each remote server performing part of the search, and it avoids a centralised controller managing the entire process.

Centralised searching seems to be necessary at present. But can the searching be even more distributed to the point of having no center? Techniques from the research could be used in a decentralised search system, instead of using existing search engines that use a traditional old fashioned centralised indexer architecture approach.

4.3.3 Spiders

One problematic issue related to locating all relevant documents is the undefined time a Web browsing agent may take to perform a search. Although a search engine can quickly retrieve Web pages, it cannot directly locate them. This is due to the Internet's dynamic nature and that to access a web page, it is necessary to know either its current URL or a hyperlink.

One solution is to use a traditional indexer and search engine to search all available documents gathered through a Web browsing agent, or a spider. A spider, also known as a "crawler" or "robot", is a program that browses the World Wide Web in a methodical, automated manner and searches for information on the Web. It is used to locate HTML pages by content or by following hypertext links from page to page. Search engines use spiders to find new Web pages that are summarised and added to their indexes [Ye et al, 2004].

To do this, a spider first takes a starting pool of URLs and a document inverted index. Until the pool of URLs is empty, the spider removes a URL from the pool and downloads the document to which it refers. The next step involves the spider extracting all URLs from the document it has downloaded and inserting the URLs into the pool and the documents into the index. This is summarised by the pseudo code in Figure 4.2 [Yates and Neto, 1999]. This is the technique that is used by ISA.

Although a spider provides a sound theoretical model by which to locate all available Web pages, there are technical limitations. Baeza-Yates et al [2005] indicated that generally, the web spider could not download all the pages on the web due to the limitations of its resources compared to the size of the web. Additionally, if a spider requests more than one page from a low-powered server, this server may not positively respond. Plus, network speed has improved less than current processing speeds and storage capacities; while one can always buy more, bigger, and faster computers, and design smarter software to coordinate them, in the end network bandwidth will be the limitation.

```
Spider(URL pool url_pool , Document index index )
    while ( url_pool not empty)
        url←pick URL from url_pool
        doc←download url
        new_urls←extract urls from doc
        insert doc into index
        insert url into indexed_urls
        for each u ∈ new_urls
            if u ∉ indexed_urls
            append u to url_pool
```

Figure 4.2 A simple spider algorithm.

As Edwards et al [2001] noted, "Given that the bandwidth for conducting crawls is neither infinite nor free it is becoming essential to crawl the Web in a not only scalable, but efficient way if some reasonable measure of quality or freshness is to be maintained." Internet search engines are attempting to index everything on the web because of their commercial interest, but Henzinger et al [2002] estimated that only a moderate percentage of the actual documents have been indexed, while growth continues at an exponential rate.

4.3.4 Internet Search Engines Constraints

In an ideal world, a spider-based search engine that updates its index using polling could maintain a comprehensive index of the entire Web. However, there are many real-world constraints on Internet Search services. These are the availability of storage and the amount of CPU cycles available. For instance, fast and reliable storage is a priority but at the same time the number of CPU cycles used in retrievals depends on the size of the index as well.

This in turn affects the time a search engine takes to return results to the user. Furthermore, search servers can only do so much work. As CPU utilisation for a single query increases, the number of queries that can be processed by a single server decreases. Thus, in order to satisfy the same user base, additional servers may need to be installed. This section reviews and discusses features of four common search engines (Google = ***g***, Alta Vista = ***a***, Yahoo =***y*** and WebCrawler = ***w***).

Google

Google is the popular Web Search Engine for many Internet users. Google was officially launched on September 1999 [Greg, 2004a]. In June 2000 Google announced a database of over 560 million pages, which grew to over 600 million by the end of 2000 and then 1.5 billion by December. 2001. The 2+ billion reported on their home page as of April 2002 includes indexed pages, unindexed URLs, and other file formats. By November 2002, they moved their claim up to 3 billion, and in Feb. 2004 it went to 4 billion

Google is now the largest in term sof size and scope, and includes PDF, DOC, PS, and many other file types also Google has additional databases (such as: Google Groups, News, and Directory).

But as Greg [2004a] indicates Google has limited search features (no nesting, no truncation, does not support full Boolean) and it only indexes the first 101 KB of a Web page and about 120 KB of PDFs. Google Link Searches must be exact and are incomplete. Google has language, domain, date, and file-type content limits. Greg [2003a] points out that Google does not always behave as advertised nor deliver the results expected. He notes that Google, like other search engines, has always given inaccurate numbers and also Google may report zero results for some terms even when it has indexed pages that contain those terms.

Alta Vista

Alta Vista is one of the large and common search engines, but it is no longer as popular as it used to be. As of March 25, 2004, Alta Vista no longer uses its own database. Instead, it uses Yahoo!'s database which has indexed over 2 billion web pages by 2003 [Greg, 2004b].

Alta Vista has considerable assets such as robust search features, international coverage, interfaces, and foreign language handling, and indexes PDF files. But Alta Vista only indexes first 110K of a Web page and 750K of PDFs and the database is not as large as it used to be.

Yahoo!

Yahoo is one of the well-known searching tools for the Internet. Originally just a subject directory, it now is a search engine, and directory Yahoo!'s own database was introduced in February 2004 [Greg, 2004b]. Greg also indicated that the last official word from a Yahoo representative about database size was in November 1997 but he estimated that probably Yahoo has over 3 million records as of 2003.

Yahoo has a large, new (as of February 2004) search engine database and has many services and products for popular and general information. But Yahoo has a lack of some of the advanced search features that other search engines have and also like other search engines only indexes the first 500 KB of a Web page and it has a very commercial emphasis.

WebCrawler (Excite)

WebCrawler (it is known also as Excite), provides sophisticated personalization, offers excellent relevant results for very popular queries, and its News Search provides important access to Web versions of newspapers, magazines.

WebCrawler (Excite) is no longer a separate search engine. As of December 17, 2001, Excite.com ceased searching its own database. It now gives Overture paid positioning results and then Inktomi results from Overture. The directory is now the Open Directory. The news search uses Dogpile's meta news search. WebCrawler (Excite) has personalization features and high relevance on popular topics but has a smaller database

These reviews further provide evidence supporting the hypothesis that the growth of indices is independent from the creation of new pages, and that it is related to the addition of substantial resources to the service [Henzinger et al, 2002]. Henzinger et al indicate that search service companies are focusing on improved finding of quality matches within their index rather than expanding the size of their index.

There have been changes in the search engine industry over the past few years, with surprising announcements almost every month. While most of the search engines domains remain active, the actual state of the search engines is quite different, like so many aspects of the Internet; the death of a search engine is no simple matter. It can come in a variety of versions. Indeed, most of the original search engine URLs remain, and with some kind of a search box on the page.

Google, Alta Vista, and Yahoo all still survive with unique databases and features. Lycos also survives, using the Fast Search database that is also available at www.alltheweb.com. So despite all the changes, the remaining search engines continue to fill a major role in information retrieval on the Internet.

4.4 Summary

This chapter highlighted Internet searching issues by looking into the methods currently used for retrieving information from the Internet. This chapter touched on topics like automatic browsing techniques, discussed the role of internet search engines in addressing how they are working and it presented web document collection issues. Further the method of using a spider based index to provide a comprehensive search is discussed. This highlights the facts why the Internet search engines are most commonly used in the modern world for extracting information from the Internet.

CHAPTER 5

5 Methodology and Searching Measurement

This chapter addresses the methods of the search strategy used in this study. It involves combining three retrieval methods (text, hyperlink and directory structure). The chapter also explains the importance for adapting these methods to the process of collecting (see Oard and Marchionini framework section 2.1.2 *Collection* Figure 2.1) sample data for this study and the issues involved are highlighted.

Moreover the chapter is present searching process (see Oard and Marchionini framework section 2.1.2 *Selection* Figure 2.1) and emphasises four measures needed for investigating search technique that is used in this research.

5.1 Research Contribution Overview

Previous chapters discussed text retrieval research and various Information Retrieval (IR) approaches were considered, including Internet search techniques.

As Chakrabarti [2003] pointed out traditional information retrieval has inspired Internet searching systems. However, searching techniques from traditional IR were not good enough for Internet retrieval. Therefore this research investigates other Internet features in particular Web pages are rich in sources of information (e.g. text, hyperlinks, and directory structure) and thus offer an opportunity to utilise various sets of retrieval approaches.

The nature of the Internet search environment is such that retrieval approaches based on single sources of evidence can suffer from failings that can limit the retrieval performance in certain situations [Chakrabarti, 2003].

For example, Chakrabarti [2003] verified that text-based Information Retrieval approaches have difficulty in dealing with the diversity of vocabulary and quality of web pages, while link-based approaches can suffer from incomplete or noisy link topology. A third approach as is to integrate the text and link-based approaches by using directory tree structure that reflects the structure of a Web site in a way that has the potential for an enhanced retrieval strategy.

Having considered three approaches (text, hyperlinks, and directory structure) for designing the search system to carry out this study the system needs to identify:

- **Content of Web pages:** The Web page contents include Text, HTML tags, and Fields (i.e. title, METATAG, URL).

- **Hyperlinks of Web pages:** Both incoming and outgoing.

- **Characteristics of Web pages:** The characteristics of a Web page include the directory structure, file name (e.g. index.html), File size, File type (i.e. *PDF* file, postscript file).

This research studies not only the effects of combining the retrieval methods, but also goes a step further by using the directory structure of each web site, reflecting as far as possible the structure of that page on the Internet itself. The ISA system (Internet Search Agent) developed in this research exploits this combined approach consisting of:

- Fingerprint (Text-Based approach)
- FP_{-1}, FP_0, FP_{+1}, (Link-Based approach)
- Directory structure

Each of these three explained in the following sections.

5.1.1 Fingerprint Definition (Text-Based Approach)

In the Text-Based approach, given some text, say ***T***, a 'fingerprint' $FP_0(T)$, or FP(T) for short, can be formed from the histogram of the words. This histogram is simply the alphabetic list of the words that the text contains together with the number of times that the word occurs in the text i.e. the word's frequency. Thus the histogram of the text:

> *Although the Web will mean a rapid convergence in the functionality available across cultures and machines it could also lead to the divergence of the Human Computer Interface into a much more personal and more effective forms. While this concept would pose some difficulties, it also holds out considerable benefits for users.*

can be represented as:

> *{a:2, across:1, also:2, although:1, and:2, available:1, benefits:1, computer:1, concept:1, considerable:1, convergence:1, could:1, cultures:1, difficulties:1, divergence:1, effective:1, for:1, forms.:1, functionality:1, holds:1, human:1, in:1, interface:1, into:1, it:2, lead:1, machines:1, mean:1, more:2, much:1, of:1, out:1, personal:1, pose:1, rapid:1, some:1, the:4, this:1, to:1, users.*:1, *web:1, while:1, will:1, would:1}*

A simple fingerprint might consist of the histogram itself. Sometimes a histogram may have words that are thought not worth consideration in identifying the meaning of the text. These words are called '***Noise Words***'. For example (depending upon the application), it may be wished to ignore parts of a histogram containing common words such as 'a', 'the' and 'to'. The context will determine when this is done.

A histogram can likewise be formed from a file. When a file contains more than just text, an HTML file for example, there is a choice of including or ignoring other non text features (such as mark up tag) in the fingerprint, depending upon the context.

5.1.2 Interpretation of FP_0, FP_{+1} and FP_{-1} (Link-Based Approach)

Using the Hyperlinks, relationships can be defined between files containing HTML. If one file points to another, through an HTML link for example, the first file is called an '**immediate predecessor**' (written as '1-predecessor') of the second and the second an '**immediate successor**' (written as '1-successor') of the first. These ideas can be generalised to include longer chains of links. For example the Web site of W3 (World Wide Web Consortium), the following shows three files on the web with → showing a link(see Figure 5.1):

www.w3.org/ → www.w3.org/Consortium/Activities → www.w3.org/RDF/

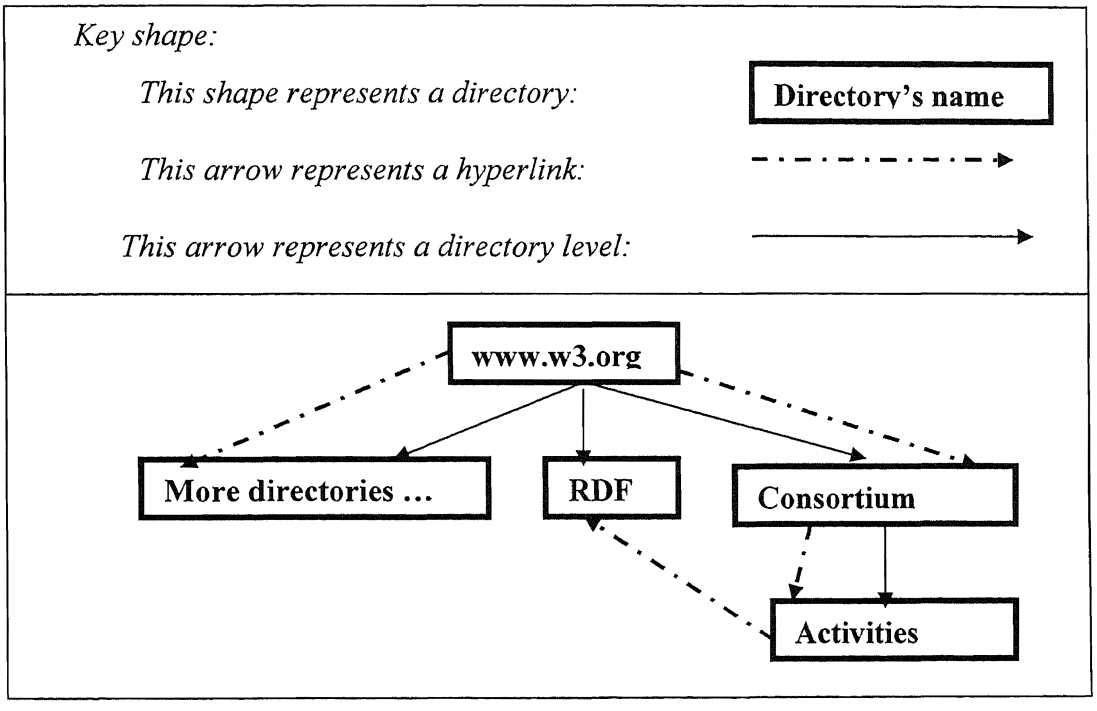


Figure 5.1 The tree structure of w3.org web site.

In this example www.w3.org/ is said to be a 1-predecessor of www.w3.org/RDF/. Notice that as often happens on the Web, the actual file name is not given but a path is. The file actually referred to will depend upon the server's operating system or set up for example some server use "index.htm" or "default.htm".

The Fingerprint of a file indicates something about what it contains. In a more indirect way, the Fingerprints of its predecessors and successors also point to its contents.

Suppose that a search for information about 'Klingons' is being undertaken. Then there may be files mentioning the word Klingon many times. If a file contains Klingon in its fingerprint then it would, on the face of it, be a good place to look for information about Klingons.

What about files that were pointed to by many files containing Klingon in their fingerprint? Experience leads one to believe that pages pointed to by 'interesting' pages, have a chance of themselves being 'interesting'. Clearly the connection may be more tenuous but there is a good chance that, having 'stumbled' upon a good source of information, that source may point to an authority on the subject at hand. This is the basis of what is called FP_{-1}. The FP_{-1} fingerprint of a file F is the sum of all of the fingerprints of those files pointing to F. FP_{+1} fingerprints are defined through a file's successors in a similar way. This can then be formulated using the following expressions (in symbols):

If ***F*** is a file and ***H (F)*** is its 'histogram', i.e. tuples of words in ***F*** and their frequencies then these can be added to form histograms in the natural way.

*The successors **S (F)** are a set of files given by **S (F)** = set of files pointed to by **F**.*

*The predecessors **P (F)** are a set of files given by **P (F)** = set of files pointing to **F**.*

For both of these the original file is removed from its predecessors or successors if it links to itself and any duplicate files (see appendix A Figure A.5 to Figure A.7).

FP_0 (*F*) = histogram of words contained in *F*

FP_{-1} (*F*) = sum of the histograms of the immediate predecessors of *F*

FP_{+1}(*F*) = sum of the histograms of the immediate successors of *F*

The mathematical notation of this is represented as follows (see Equation 5.1 and 5.2):

$$FP_{+1}(F) = \sum_{s \subset S(F)} H(s) \qquad \text{Equation 5.1}$$

$$FP_{-1}(F) = \sum_{p \subset P(F)} H(p) \qquad \text{Equation 5.2}$$

These definitions can be generalised to $FP_{+k}(F)$ and $FP_{-k}(F)$ in the natural way although there are some technical issues to be considered in the precise definition. These technical issues include accounting for duplicate files and occasions when the URL does not specify the filename. The system then looks for "default.htm", "index.htm" or as other system specific file. Subsequent to the initial design of the fingerprint approach presented here, it apparent that others were thinking along the lines of what is here called FP_{-1}:

"the well-known PageRank, essentially a measure of how many other pages point to it." [Bray, 2003e]

Google seems to have been developed around the same time (1997/98) as this research was being designed. Whilst quoting from the series here is one of Bray's conclusions:

"Google seized search leadership from Yahoo; can we conclude that it's more important to know how popular something is than to know what it's about? If you'd told me that ten years ago I would have had a hard time believing it, but the evidence seems pretty compelling." [Bray, 2003e]

Where this work parts with Bray is that it looks only at information that is available for all html pages on the web and does not take any special note of meta data that may be included.

5.1.3 Directory Structure of Pages (Hierarchical Structure Approach)

The directory structure used by a web site designer represents one way of showing its contents and this information can be used by a search engine. Figure 5.2 shows an example of such hierarchical structure of a web site on a web server.

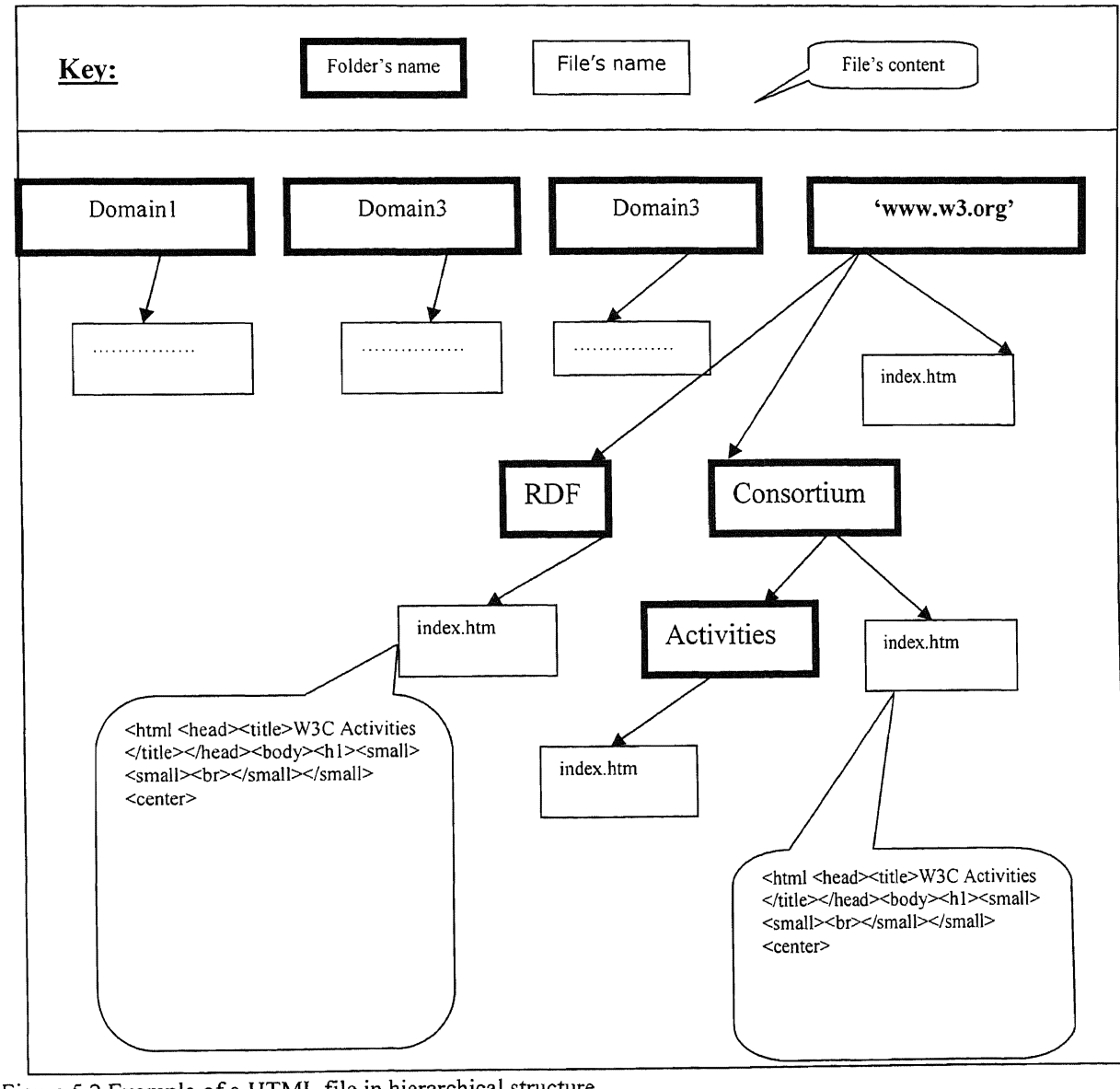


Figure 5.2 Example of a HTML file in hierarchical structure

In order to use hierarchical structure for the fingerprint approach, the fingerprint (FP) of a directory is defined as the sum of the FP of all the documents and the sub directories contained in the directory. Generating the FP of a directory involves the following steps (see Figure 5.3):

1. Get the main URL of a web site plus its files (e.g. www.aaai.org)

2. Obtain sub directories of the site (e.g. Magazine, Advertising, and Articles)

3. Generate directories and sub directories for new FP files.

4. Generate a FP for each Web page.

5. Save all of them under the main directory, for example a directory called WWW in sample of the web (see section 5.3.1 for *Sample Web*).

The main URL (www.aaai.org) is characterised by a single FP. This cumulative FP indicates the content of the site, enabling an easier search of this node.

Starting with directories containing no sub directory add the FP of the files in the directory to form the directory FP (StdFP=fileFP1+fileFP2, see Figure 5.3).

For a directory containing sub directories the sum of the sub directories FP generates the higher-level directory FP for example the Fingerprint for the folder "Articles" in Figure 5.3 (FPArticles=folderFP1999+folderFP1998+file FP1).

From the Fingerprints one can now calculate the Fingerprints of all the branches from that directory plus all the files in the directory. For each sub Fingerprint (branch), a number of characteristics are collected when calculating the Fingerprint: the highest frequency words, the URL, and the number of included documents.

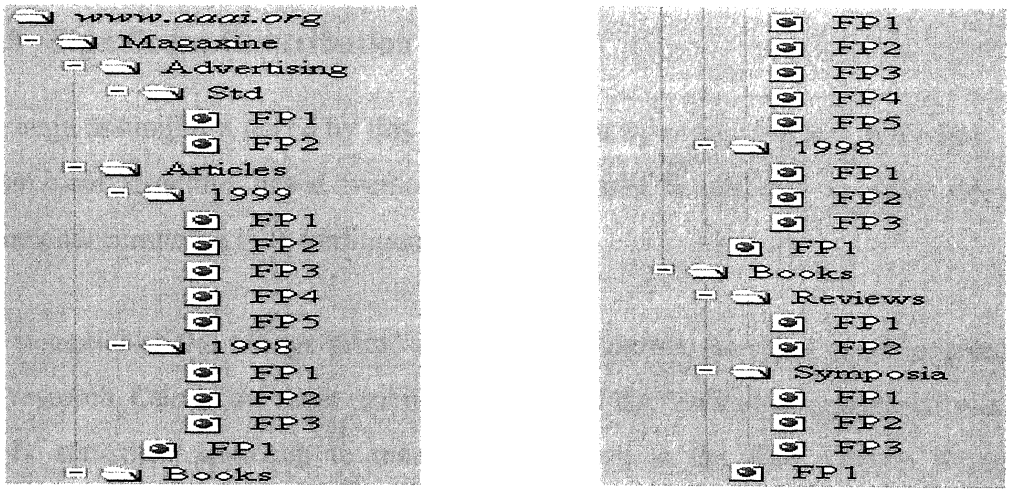


Figure 5.3 Hierarchical Finger Printing.

5.1.4 ISA: an Aid to the Investigation

The Internet Search Agent (ISA) helps the understanding of the possible use of Fingerprints as an aid to searching the Internet. ISA was developed to test the ideas of fingerprinting, the link-based approach and the directory structure. Its architecture is explained in the following chapter (Chapter 6). Given a local web (in this study called the *Sample Web* see section 5.3.1), ISA can be instructed to produce fingerprints of various types, such as FP_0, FP_{-1} and FP_{+1} (see section 5.1.2). ISA is able to calculate and produce the FP_0, FP_{-1} and FP_{+1} of a document and its directory structures. The results of a search using a conventional search engine can be downloaded by ISA and stored on disk. The results and analysis of the downloaded pages are stored in an Access database for further processing.

ISA is also able to search fingerprint files for target words, ranking the results according to user specified templates which allow the user to give weights to the components of the target fingerprints. ISA is able, for example, to score (see section 5.2.1) pages against target patterns according to user defined criteria.

5.1.5 Significant Contribution of ISA

The main assumption tested by this research is that by combining text, hyperlink, and hierarchical structure based methods, the user will be directed to more relevant documents compared with traditional searches.

The potential of the "Fingerprint" approach for Internet searching is investigated in this research. One of the most striking examples of the "statistical hypothesis" was the purely statistical approach to machine translation at the IBM Watson Research Laboratories which made use of the very large Canadian English /French parliamentary corpus [Brown et al, 1988]. The results were striking: with virtually none of the conventional sources of linguistic knowledge (lexicons, syntax, semantics, etc.), the system produced figures of between 50 and 65% of sentences correctly translated, depending on the relationship of the training to the experimental corpus. Wilks [1998], however, has pointed out following more detailed analysis that there is evidence that some linguistic phenomena are resistant to this approach.

Chakrabarti [2003] explained that Text-based approaches have difficulties in dealing with the vocabulary problem (i.e. different expressions of the same concept), the diversity of document quality and content, fragmented documents, and documents with little textual content (e.g. index pages, bookmarks). He also indicated that Link-based approaches do not fare well when faced with a variety of link types (e.g. citation links, navigational links, and commercial links).

The most obvious of the complementary strengths is found in the combination of text- and link-based approaches. Ranking text-based retrieval results by a measure of "link importance" may help differentiate Web pages with similar textual content but varying degrees of importance, popularity, or quality [Chakrabarti, 2003].

Combining text, hyperlink, and hierarchical structure based methods can be viewed as attempting to maximise the combined strengths of accessing the content and structure of the Web page and its context while minimising their individual weaknesses with regard to internet searching.

In this project, therefore, the combination of directory structure, text and hyperlinks are examined in order to learn more about different approaches of Internet searching. The aim is to investigate the potential of combining text, link and tree structure methods, and develop experiments to evaluate the effectiveness of this approach.

5.2 Measures Used in this Research

The aim of a search is to produce a list of results containing the 'best' URLs for the search topic. It should do this with a minimum 'effort' and in the quickest time.

This study considers the search process after the search has been formulated. It looks at a particular technique that does not use indices.

In order to investigate any particular search technique measures of search effort and search progress are needed as well as some measure of relevance of a particular URL to a given search criteria. This section address four measurements that are used in this research:

1. Measuring the relevance of search results
2. Measuring search effort
3. Measuring search progress
4. Comparing search engines

5.2.1 Measuring the Relevance of Search Results

Commercial search engines evaluate pages against the search criteria using complex and largely unpublished formulae. Because of the secrecy associated with these formulae and the complexity of their calculations, they are not available for use in this research.

In general search engines use large databases which contain information about the contents of web pages and use database technology to speed up their searches. Besides the time it takes to return the search results no measures of search effort are available. Also, in general the results are returned at the end of the search process and no intermediate states of search are available. In short concepts such as search progress and search effort are not easily applicable during the use of commercial systems.

The approach used in this research is very different. The web pages are abstracted into fingerprints with one fingerprint file produced for each web page and other type of fingerprint. These are stored in a structure similar to the web itself. Searching then consists of traversing this structure and returning the URL in order of decreasing score.

The relevance of a page to a query is not easy to estimate. One method that has been tried is to ask 'experts' or 'searchers' to give a subjective view of the relevance of a page to a particular search. This is a very expensive and slow process [Greisdorf and Spink, 2001].

In this work measures of relevance are sought that are more objective even if at the expense of accuracy. Measures are selected that are easy to calculate and robust in that the effects of adding the results of new search engines or adding more search results does not lead to a need for complete recalculation.

The term ***score*** has been used above (see section 5.1.4) to indicate a function associated with a URL that can easily be calculated from the search expression and the fingerprint file. Any such function could be used in such searches. One possibility is to use a count of the number of occurrences of the search term in the HTML file at the particular URL.

In commercial systems and major IR systems this ***score*** (i.e. number of occurrences of the search term) forms part of the calculation that produces the relevance of a result to a given search; different search engines, however, treat this score in different ways [Stata et al, 2000]. This is the measure of relevance used in the examples given in this thesis and will henceforth be referred to as ***score***. This is used because:

- Experience shows that it is correlated with the content of a page
- It is easy to calculate
- Higher scores are, in practice, correlated with more relevant pages
- It is robust
- It is used in successful commercial systems

The score of the page is referred to as σ_p . When asked to perform a search, a typical search engine returns a list of 'results'. These results are usually returned in decreasing order of importance.

The '*reciprocal rank*' (see Equation 5.3) of a result is a way of measuring the importance that the search engine places on the result, which in turn is a measure of its perceived relevance. The expression 'Google relevance' is used for the reciprocal rank of a result in Google's list, so that for page ***p*** its Google rank is given by:

$$r_{pg} = 1/(\text{Google rank of page } p) \qquad \text{Equation 5.3}$$

where the first subscript (p) indexes the page and the second subscript (g) indexes the search engine. Using the reciprocal rank the results of a number of searches using different engines may be combined.

The sum of relevancies of a page given by the 'community' of search engines that are being considered will be referred to simply as its *Community Relevance* (r_p) and as defined by Equation 5.4.

$$r_p = \sum r_{pe} \qquad e \in \{a, g, i, w, y\} \qquad \text{Equation 5.4}$$

where the sum is taken over all search engines used in this research.

Community Relevance is not an absolute measure but a high (or a low) *Community Relevance* does give an indication of the value (or respectively otherwise) of the page to the search request.

Page *A* has a higher *Community Relevance* than page *B* if its sum is larger. Five search engines are used in the calculation of *Community Relevance* in this work. Thus, for any given search the maximum *Community Relevance* that a page can have is 5 (if a page were to be at the top of all the search engine's results) and the minimum 0 (for pages not appearing in the results of any search engine). Intermediate values of *Community Relevance* would indicate either that the engines do not rate a page highly or that they do not agree, with some giving high whilst others giving a low relevance.

5.2.2 Measuring Search Effort

The search process for a current search engine involves database look up and calculations based upon the content of the search query and the indices used by the engine. The amount of effort expended by an engine in performing a search depends crucially upon the details of the algorithms and data structures used.

In the experimental system described, the process is more complex and slow. No attempt has been made to speed up the system by using indices to aid search as one of the main interests of this work is the structure of the *Sample Web* (see section 5.3.1) itself. In particular the 'Fingerprint' files behave in some ways as crude indices, they are not inverted and so it is necessary to search for terms in the many files that model the sample.

For this work an indication of such effort is useful, and based upon the algorithm that fingerprint searches use, this was chosen to be easily available as the search progresses and clearly corresponds to the computational resources used by the search.

5.2.3 Measuring Search Progress

There are a number of measures of search progress that can be used. The literature contains little information on this aspect of search.

One parameter measuring search progress is the number of results reported. This is done either by specifying the actual number of results, the number of results as a percentage of the total possible or as a percentage of the final number of results. Thus if in a particular query there are potentially 1000 results to return from the *Sample Web*, but 500 have been asked for, then once 100 results have been returned this may be reported as 100 or 10% or 20% respectively

The percentages will be used when the comparison of raw figures is not as clear as a comparison of these 'normalised' numbers. Thus, comparing a search for a term, which only occurs 100 times in the *Sample Web*, against a search for one that occurs 10000 times, would be unhelpful without this normalisation. Presentation of the results using this normalisation can only be done once the search has been completed, for it is only then that the final numbers of results are known.

If a numerical measure of relevancy of a result page is available that increases with relevancy, then another measure of search progress can be formed from this by taking its sum.

For example, using the number of occurrences of the search term in a result as a measure of relevance, the total of this found so far can be used as a measure of search progress. If the total possible sum (in the example the total number of target terms) in the *Sample Web* is known then progress can be expressed as a percentage of this. In what follows the term '*score*' is used to describe any such measure of how closely a search result matches the search query, target word count is used as a concrete example although any such measures could be used as alternatives.

In this research number of search results and total score are used as indicators of search progress.

5.2.4 Comparing Search Engines

Using a number of search engines to answer a query produces, in general, quite different sets of results (see Chapter 7 section 7.5.3). A means of comparing these results is needed. Given two ordered lists, a correlation between these lists can be calculated using the standard methods of statistics. Two lists that differ from one another in that one includes an item that the other omits, should be given higher correlation in the current context if the item omitted from second list occurs late in the list but lower correlation if it occurs early in the list of results.

In order to compare the results of several search engines, a number of measures have been developed (these measures are given names, which are meant to evoke some feeling of their meaning in the reader). The names are written in italic to remind the reader that their meaning is to be found in their definition and not solely in the name that is given to them. Thus in this work terms such as *strength*, *vigour*, and *relevance,* which are introduced below, are to be read with their definitions in mind.

A measure of the commonality between a search engine and its peers is its '*strength*' (see Equation 5.5). For a particular set of results and search engines, a search engine is given a '*strength*' which is the sum of contributions from each of its results. For a particular search engine the contribution of a result is the sum of relevancies given by all of the other search engines multiplied by its relevance for that search engine being measured. So for page p with relevancies r_{pa}, r_{pg}, r_{pi}, r_{pw} and r_{py} its contribution to the *strength* s_y of Yahoo is given by:

$$s_{py} = (r_{pa} + r_{pg} + r_{pi} + r_{pw}) * r_{py}$$

The *strength* of Yahoo is given by: $s_y = \sum s_{py}$ Equation 5.5

where the sum is taken overall of the results that Yahoo returns in this search.

It is important to note that as some search engines share information they are likely to agree more with each other because they use the same database or otherwise share resources. This may give them an unfair *strength.*

The *union score* (see Equation 5.6) of a search engine is the sum of the scores of its results (up to some cut off number of scores) is also a measure of its success. Thus the *union score,* u_y of Yahoo is:

$$u_y = \sum \sigma_p \qquad \text{Equation 5.6}$$

where σ_p is the score of Web page p and the sum is taken on the overall results that Yahoo returns in this search. The higher the union score the more search terms appear in the results pages. However the number of results used for this measure needs to be considered for it to be used in a fair comparison of search engines. This is done by using the first fifty results.

It could be argued that the position of the result ought to be taken into account when evaluating the search engine as has been done for its *strength*. If the formula for *union score* is modified to take this into account by multiplying by the relevance of the page the '*vigour*' (see Equation 5.7) of the search engine is obtained. Thus the *vigour* of Yahoo is:

$$v_y = \sum \sigma_p * r_{py} \qquad \text{Equation 5.7}$$

where the sum is taken on the overall results that Yahoo returned in this search.

5.2.5 Measurements to be Taken

Because of the wish to monitor both progress and effort the following are stored as the search progresses:

- The *search rank* of a file which is just its place in the search (not the results)
- Total files ('html' and 'folder') opened {TF} to get to this stage of the search
- Folder files opened {FF} and the html files opened {HF} to get to this stage of the search
- The *score*, that is the score against the matching criteria
- The *result rank* of a result, its final position in the list of search results.

From these the cumulative score for the search so far can be calculated. This is labelled the *Cumulative Score* in the tables and graphs.

5.3 The Experiments

In order to understand the potential value of the fingerprint idea a set of experiments were designed. The following section describes the strategy and rationale for the decisions that were made in designing the experiments.

5.3.1 Building the Sample Web

To produce results that are at least to some extent reproducible it was decided that a manageable number of web sites be downloaded to form a 'universe of discourse' that does not change from day to day. Given this decision there is the problem, when trying to evaluate fingerprints against other search techniques, of bias caused by the choice of sample. There are a number of alternate strategies that present themselves.

The first and possibly most obvious one is to see if there is any overlap between results returned by current search engines and the particular sites that were downloaded. Initial trials of this idea showed that unless a good proportion of the web were to be downloaded it would be unlikely that there would be any significant overlap between what was downloaded and what the search engines returned as results.

A more promising possibility would be to use the results of conventional search engines as a basis for a sample web, fetching the web sites that they return. From everyday use of search engines it is expected that that not all of the results would be worth downloading, not just in terms of their content but more because of their form.

It was decided to exclude not only 'virtual' pages described previously but also those pages in languages other than English and character sets that did not display correctly on the facilities available. The strategy thus became:

a) A number of search terms and a sample of search engines were chosen.

b) For each search term and search engine a search for that term using that engine was performed followed by the removal of any 'inappropriate' results in terms of language or character set.

c) As much as possible of the domains that resulted from the searches in b) above were then downloaded.

This strategy resulted in a concentration of pages that are 'likely' to be relevant to the searches performed. However this is not seen as a particular problem as the aim was not to produce a stand alone search engine but rather to obtain an understanding of fingerprints and their relationship with search.

In order to alleviate some of the expected complexity of the process it was decided to store the example web pages and other results in structures that mimicked, as far as was possible, the domain structures that they come from on the Web.

Thus a page with URL: www.microsoft.com\help\outlook\index.htm, would be stored as a file with the same name in a directory structure www.microsoft.com\help\outlook built from the URL and named index.htm.

A final important point is that the *Sample Web* was to be kept small enough to work within the limits of resources available but large enough to characterise the web. In this research the *Sample Web* consist of 1.220 Gigabit of HTML files (see section 7.3 table 7.2).

5.3.2 The Search Terms

In order to generate the web sample a number of search terms were chosen. Except for *Duran* and 'liquid marbles', these were chosen from the area of lean manufacturing with a view to the inclusion of terms that would generate a mixture of important issues. Because of the widespread use of the Web across most of the developed world it would be very hard to predict 'the meaning' of any particular terms that may be searched for. Rather many different meanings are likely depending upon the context. The intention was to concentrate most of the effort on search terms, which are single words from a technical area not related to computing.

At least one word for which there is likely to be two or more 'strong' contenders for meaning are required. That is a word that is, in Web terms ambiguous. Another choice was for a person's or music group's name should be included. This would likely result in many pages that may be described as part of the 'grey' Web. These are web pages that are poorly indexed because they are personal pages or pages of 'fan clubs'.

The inclusion of a term that was foreign but widely used amongst a group of English speakers was also desirable with the expectation that many of the results returned by the traditional search engines would be discarded.

The ambiguity of acronyms was also thought of as important. Thus a word that has a meaning as an ordinary but possibly technical term at the same time as being one which could be thrown up as an acronym.

Another wish was to see what would happen with a two word search phrase where the individual terms taken separately would not give a hint at the meaning of the phrase table 5.1 gives the main search terms used.

Search word	Features
Duran	The most common meaning is of course the group Duran Duran. It also turns up often as a name and as the Spanish for last.
Gemba	The meaning of Gemba derives from Japanese waste reduction philosophy. However it refers also to Global Executive MBA courses, people's names etc.
Lean	This is a collection of activities in manufacturing, design and business processes through continuous improvement. Of course it also has its 'original' meaning(s) related to lack of excess.
Liquid Marbles	Taken separately these two words will appear in an enormous number of sites. Together they have a special meaning used in the study of surface tension effects between two liquids.
TRIZ	This is a method of discovery and innovation used by lean manufacturing Gurus. One feature is that would be expected some foreign language (Russian) sites to appear. According to Google, there are over 25000 sites that contain the word!

Table 5.1 search terms.

(Other search terms were also used for some of the experiments but those listed in Table 5.1 were used to build the sample web)

Table 5.2 lists some of the other search terms used in this study. They were chosen because of their frequency *within* the *Sample Web*. This was of course not known when the sample was being downloaded but resulted from an analysis of this sample. The words were chosen to represent classes of words with similar frequencies so than any major differences in their behaviour might be studied.

Words that occur infrequently in the sample may be quite common words but ones that do not occur too often in the sample.

For some of the test search terms were divided into groups depending upon the number of times the terms appeared in the *Sample Web*.

The groups were listed in Table 5.2:

Word frequency	Words
1000	algorithm, broadband, disc, encyclopaedia gospel, lavender
100	allegiance, berman, best-sellers, compassionate, convictions, correlates, dehydration, Humane impatient, intellectuals, Masculine, migrated, plagiarism restrictive, stockholders, subconscious, superiority transistor, wellington, whiteboard
10	acrobatic, addressable, adriatic, allergenic, bodyguards bodywork, circumspect, cubicles, disharmony experimentalist, festering, flutist, gymnastic, hollowed incisors, industriously, instinctual, interrogative knaves, lavishly, licentious, manicure, mapmakers nuclide, onerous, pamphleteer, radiologist, resettlement smugglers, stabilisation, typologies, uncircumcised unforgivable, vaccinate, Waives

Table 5.2 search terms, which were chosen because of their frequency.

This was done to see if this 'word frequency' factor influenced the results of the experiments. In the results and graphs that follow the groups are referred to by their frequencies, the terms: 10 (for the average of the words with frequencies 10), 100 and 1000 are used in the legends.

5.4 Summary

The focus of this chapter has been on the relevance of results and how search engines compare. Suitable measures for the measurement of effort, progress and the relevance of a web page to a query have been defined and measures for comparing search engines proposed.

This chapter also explained how sample data was assembled for this study and what kind of search terms was selected. The next stage of this research is implementing the ISA tool. Chapter 6 deals with how ISA is designed, its structure, AND demonstrates how it works, its limitations, and some of the challenges faced in developing it.

CHAPTER 6

6 Designing the Internet Search Agent (ISA)

This chapter discusses the methodology used for the Internet Search Agent (ISA). It begins by describing the implementation of the ISA. Later it explains how ISA creates a fingerprint and the processes needed for producing FP_0, FP_{+1} and FP_{-1} and a number of issues related to the search of retrieval results. The challenges faced in the design of the ISA are elaborated and the search strategies used are discussed.

6.1 Purpose of ISA

ISA was developed to achieve three main objectives. The first one is to create the *Sample Web* necessary for this research, second one to implement the search strategy (using fingerprint and directory structure) described in chapter five. The third objective was to analyse the results generated by experiments.

ISA was developed to examine the ideas of integrating fingerprinting, hyperlink and the directory structure. ISA produces all three fingerprints (FP_0, FP_{-1} and FP_{+1}) which are examined in this research. It helps to investigate the potential use of fingerprints as assistance to retrieving information from the Internet.

6.2 Design of ISA Structure

In Figure 6.1 the general structure of the ISA is shown and the main inputs and outputs of the system are illustrated:

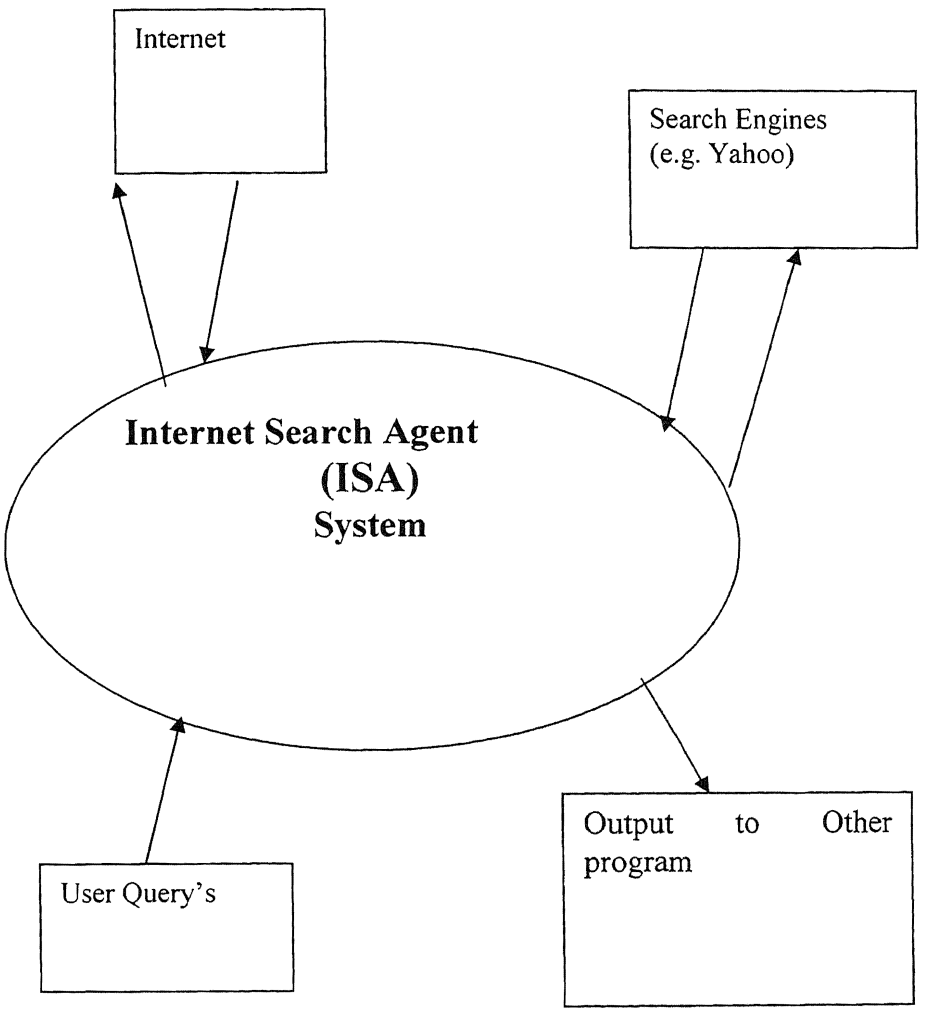


Figure 6.1 Content diagram for ISA showing main components.

- **Internet**

This part is used for fetching selected URLs, which are used for the *Sample Web* processes of the system. The Internet is the source for gathering information from the Web to maintain a repository, the *Sample Web*.

- **Search Engine**

This part is used for selecting URL's using search terms chosen for this process (as explained in section 5.3.2). Aside from this, search engine's results are used for comparing and contrasting the search results acquired by the experiments of this research using ISA.

- **User Query**

This part is used for selecting search terms in the research experiments, as well as for searching *Sample Web* preparation queries.

- **Output to other program**

This part is used for collecting and categorising research experiments result, in order to carry out more analysis and study using other program such as Microsoft Excel.

6.2.1 Architecture of the ISA

Figure 6.2 shows the ISA structural design. This diagram shows three modules for manipulating the main inputs and outputs of the system. These modules are:

- **Input System:** The input system has two objects, the WWW object and the User Object. The WWW object is used for providing Web pages for the local repository and the user object is used while dealing with search query terms.

- **Output System:** This has the storage object, which provides the output facility for the ISA system in the research experiments.

- **Operation System:** This is used to deal with automation processes, *Sample Web* building processes, experiment's results collection, searching and query processes.

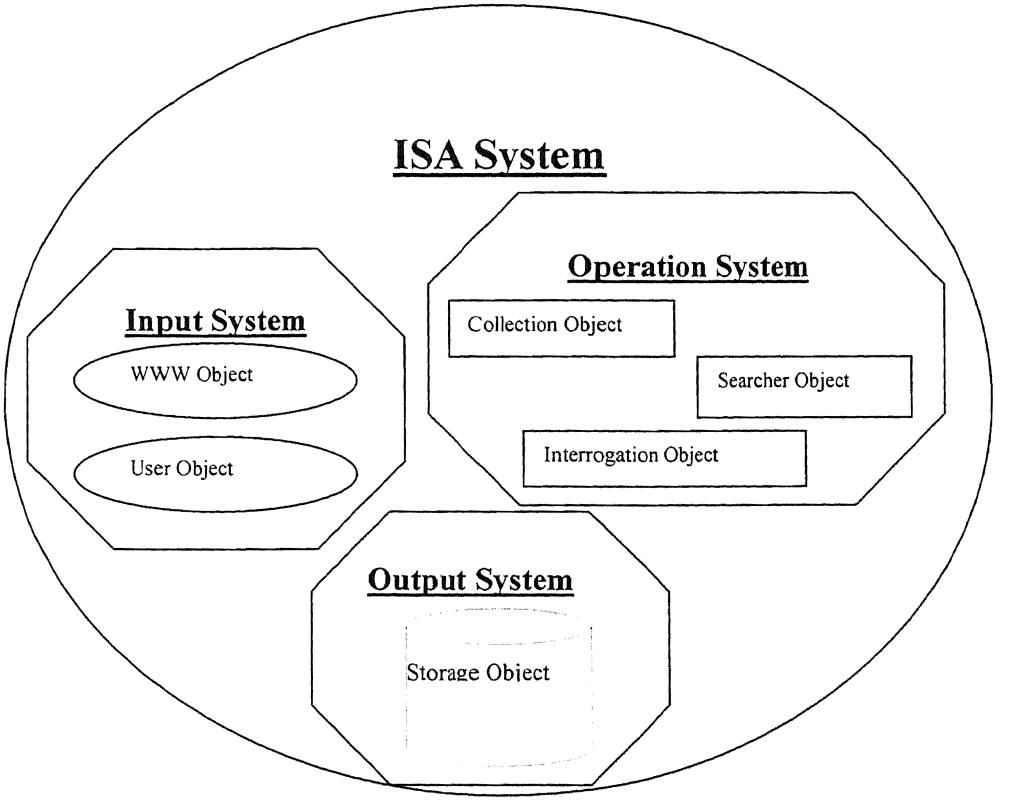


Figure 6.2 ISA architecture, this shows the three main elements of the ISA concept.

6.2.2 Relationships Between Objects in the ISA

In Figure 6.3 the object relations of the ISA are illustrated. ISA consists of the following six main objects (also see Figure 6.2):

1. ***WWW*** object: is an agent that gathers information from the Internet, to maintain a repository of *Sample Web*.

2. ***Collection*** object: consists of filters to convert, summarise and substitute contents according to fingerprint's definition of FP_{-1}, FP_0, FP_{+1}. The pages are fetched from the WWW object, stored locally and the parsing the html page extracts the URLs from them and stores the links.

3. ***Searcher*** object: retrieves the documents matching the search query content, then filters them by comparing their score against those documents which are in the list.

4. ***Interrogation*** object: translates user queries and options to the appropriate query file and then will be delivered to Searcher object.

5. ***User*** object*:* interact with user and generates a search term to be delivered to the Interrogation object.

6. ***Storage*** object: stores the search results coming from the Searcher object. It parses and aggregates the results, formats them appropriately for different tests and makes then accessible for further investigation.

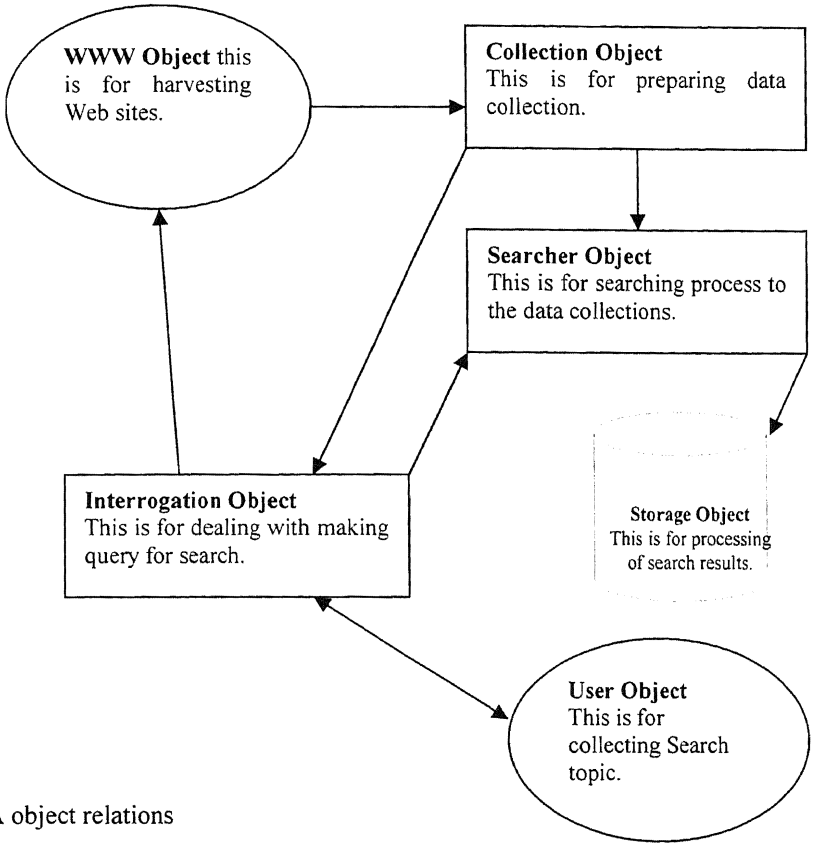


Figure 6.3 ISA object relations

ISA operates using of combining text, links and tree structure methods that can be evaluated against the current methods.

Component processes were also kept distinct from one another in an attempt to maximise the unique contributions to the combined solution space. As for the development of ISA, an Object Oriented Design (OOD) approach was taken in order to benefit from the reusability of the code. Therefore it would be easier to interface different elements of the system which will enable future upgrades following further research.

6.3 Decomposition of ISA Architecture

In this section the architecture of the ISA is decomposed and analysed to explain the functionality of the system. The ISA system has been implemented in Visual Studio which exploits the benefits of object-oriented design, boundary checking and memory management, and is designed for distributed networked environments.

6.3.1 ISA Input System

ISA Input System has two main objects, one is the WWW object and the other one is User object (see Figure 6.2).

6.3.1.1 WWW Object

The main function of this object is to provide web pages to the system for the *Sample Web* page and the indexing process. The WWW object can be subdivided into three elements; the Web site, the PageSucker[1] and the *Refinery* process (see Figure 6.4). The PageSucker process fetches the information, and then the *Refinery* process carries out the filtering action whenever it is necessary. The *PageSucker* carries out the downloading of web pages for the ISA, and is based on Java bean crawlers technology [der Linden, 1997]. The operation of the *PageSucker* can be described as follow:

- The *PageSucker* grabs a web page and stores it in an original hierarchical tree structure.
- This structure is saved on a *Sample Web* drive that is used for creating the Fingerprinting process.
- Lists of selected URLs are fetched from the Web site; all pages and the sub directory of each selected URL is fetched. Moreover one extra level of URL links is also retrieved, which is pointing out to other web pages, the **'immediate predecessor'** (written as '1-predesessors').

[1] See *PageSucker* web site at www.pagesucker.com

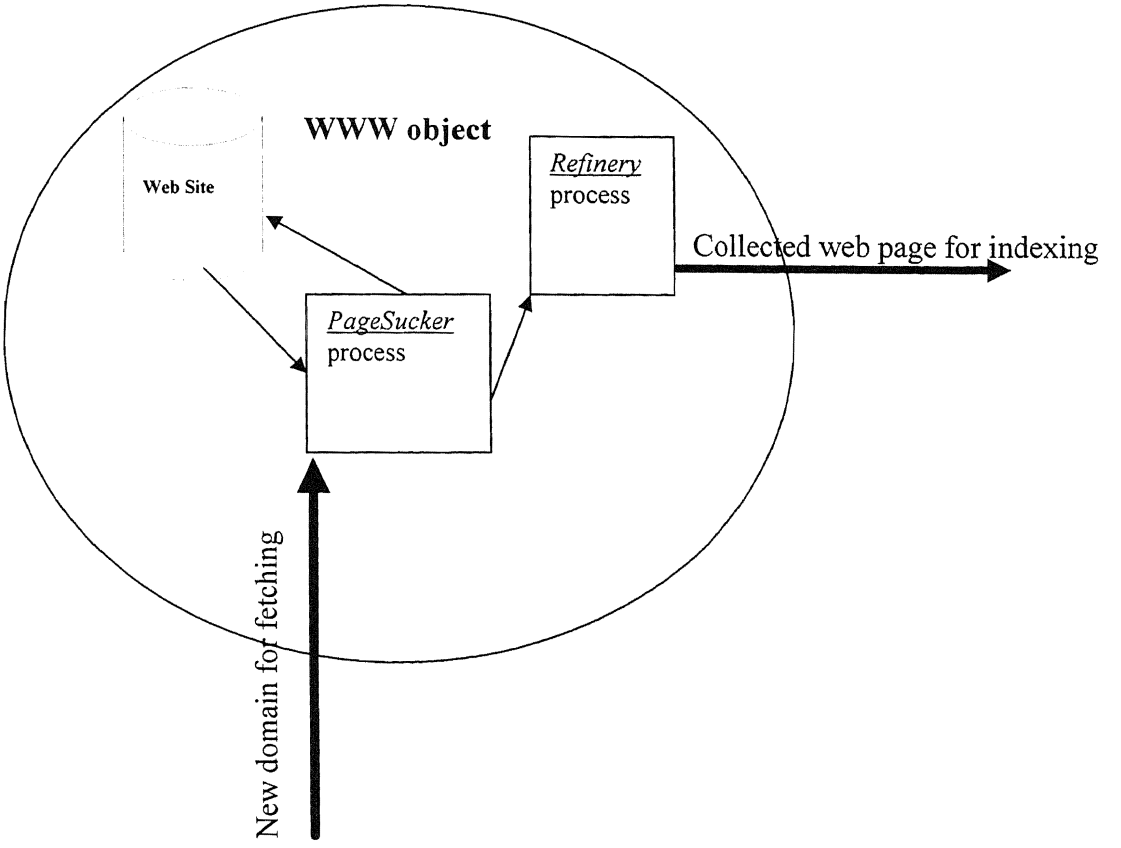


Figure 6.4 Decomposition of WWW object.

After the process of the *PageSucker* is completed, the retrieved web pages are then sent to the *Refinery* Process. The *Refinery* Process eliminates files with zero size as well as files which are located on a directory lower than level twenty this is because of limitation of Microsoft Windows and is discussed in section 6.4.

In addition, if the main page of a web site retrieved by the *PageSucker* is called something other than the '*index.htm*', it is renamed to the '*index.htm*' this can help to classify main pages. After these processes the web pages are stored into the *Sample Web*, with each location in the *Sample Web* page named the same as the original URL. Once the web pages are placed in the *Sample web*, instructions are sent to Collection Object on how exactly to generate the Fingerprints for indexing.

6.3.1.2 User Object

The User object provides the search query for ISA. Figure 6.5 shows the internal elements of the User object. There are number of ways by which a user can express its target concepts.

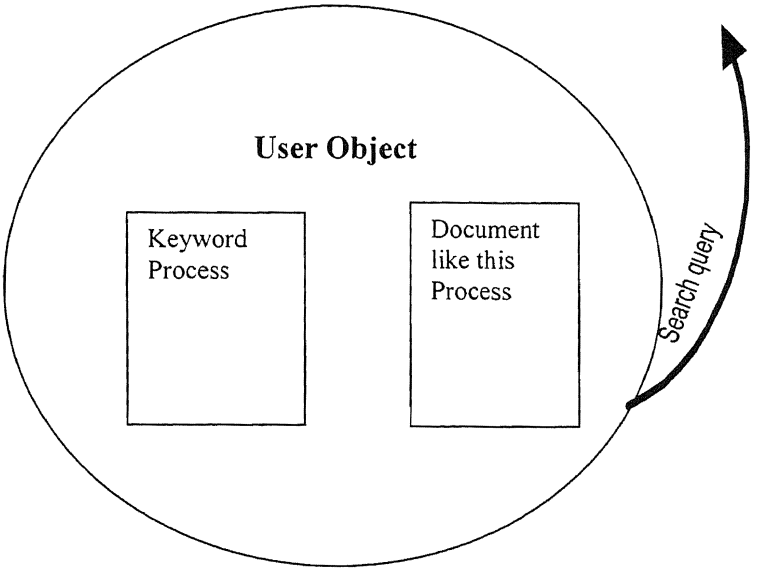


Figure 6.5 Decomposition of User object.

One way is for the user to supply the system with a file containing the keywords only for searching.

Another method for searching the *Sample Web* is by applying the "other documents like this" approach, in which the user supplies a file containing some text and then requests similar documents. After the user has provided the specifications, the user object passes the search file to the system for further processing.

In this part of the process, each keyword passes from the User object to the Interrogation object. After that the Interrogation object can create a query string either for searching the *Sample Web* or for performing a keyword searching by the selected search engines to get the resulting lists from each search engine and their URL addresses which can be used for building *Sample Web*.

6.3.2 ISA Output System

In this section the way the Storage object in the Output System provides output facilities for the ISA system is explained. Figure 6.6 illustrates the main element of the Output system called Storage object. The Storage object is organising the ISA search result so that it can be analysed by different tools such as Microsoft Excel.

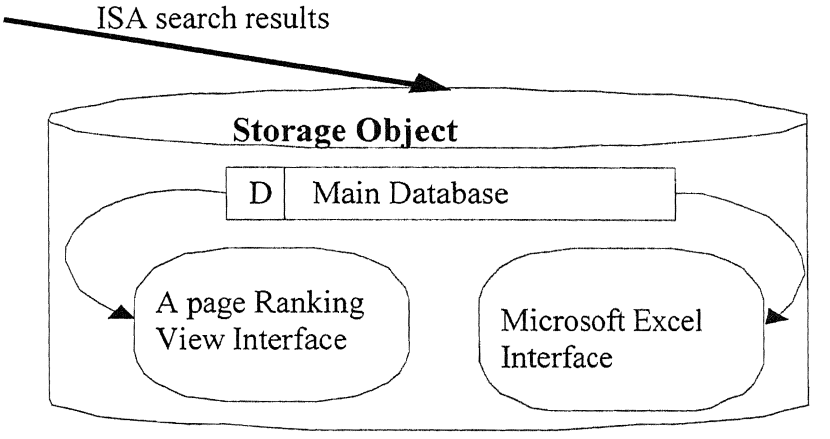


Figure 6.6 Decomposition of Storage object which is part of the Output system.

The Storage object stores information in a database, which upon request provides tables of information about Web page files and Folder files.

In order to compare the effort required to find the occurrences of a given search term within the *Sample Web*, the number of files or folders opened are recorded. The details recorded in tables are as follows:

- The location of the web page and folder file (URL),
- The search word,
- The ISA fingerprint score,
- The ranking position according to the ISA score system for the web page and folder file,
- The number of total results found in the entire *Sample Web*,
- The total number of Fingerprint web pages and folder files opened.

As the Storage object is linked to Microsoft Excel, the data collected for each experiment can be analysed.

ISA also displays the document references, in a format similar to other search services, returned from the underlying ISA search services on a single page in a organised ranked relevance list (see Figure 6.7).

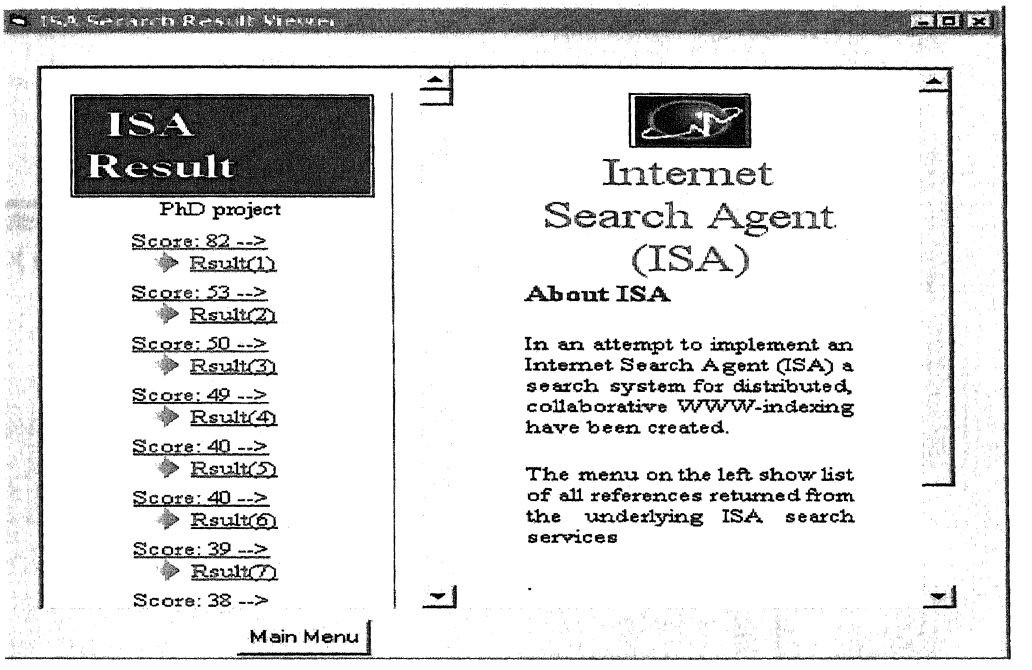


Figure 6.7 ISA search result viewer.

A ranked relevance list orders document references by their ISA scores. The ISA score is just a number indicating the relevance of the document in terms of frequency of keyword (see chapter 5).

6.3.3 ISA Operation System

ISA Operation System has three objects, the Collection object, the Searcher object and the Interrogation object (see Figure 6.2).

6.3.3.1 Collection Object

In this section the operation of the collection object is explained. Figure 6.8 shows the elements inside this object. This object uses the fingerprint model for storing files (see section 5.1 for Fingerprint definition).

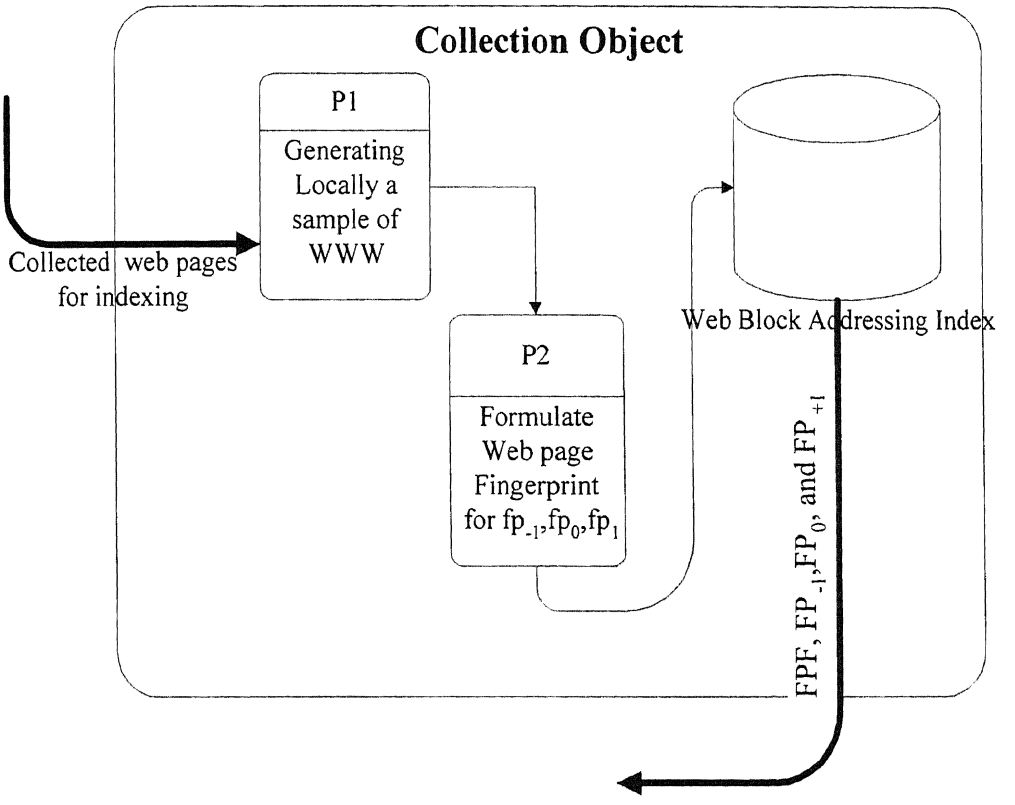


Figure 6.8 Collection object elements.

6.3.3.1.1 Process One (P1) for Collection Object

Web pages are processed one by one by removing mark-up tags and punctuation marks. After these two elements are removed all "Noise words" are removed from the document. This process also removes all the non-alphabetical words. During the development stage it was observed that some web pages contain uninformative words such as words with a large number of characters.

For example, some words were more then 25 characters or were less than 3 characters. Even some words were a repeated single character (e.g. "aaaaaaaa", "nnnnnn"). These are removed for purification of text on a web page. In this research, only the text part of the HTML file, without the unnecessary words, is retained.

Frakes and Yates [1992] used 'The simple plural remover' to removes plural words. The reason why the 'simple plural remover' was not chosen for this research is because having plural words preserves the content of a fingerprint. Therefore chance of retrieving more relevant information could be increased.

The process one consists of two main sub processes:

- HTML Page Preparation
- Normalisation
    - Stripping Non-Alphabetical Characters.
    - Creating Frequency List.
    - Noise Word Filtering.

6.3.3.1.2 HTML Page Preparation

The most important part of any IR systems is making decisions about how to organise the documents that are to be searched. If the documents are automatically indexed, they will be managed in a different way, than if they are manually indexed. Building the automatic index is as important as any other module of the ISA system.

Search Engines commonly only extract text from the title tags, the header tags and the first characters of the file, which can range from the first 275 characters to the first 70,000 characters [Bröder and Bharat, 1998]. In addition, some search engine may select the first 100 significant words or the first 20 lines (which could contain a code or script rather than text).

However ISA extracts all content text from the web page, and generates a Web Text structure. The HTML Page Purification function consists of five phases:

1. Get the HTML file name and address of the Web page (sub directories).
2. Collect all the hyperlinks in the Web page.
3. Extract the text content of the Web page from HTML tags.
4. Prepare hyperlinks for storing on a Web Reference structure.
5. Prepare text contents for storing on the Web Text structure.

6.3.3.1.3 Normalisation

This subsection describes converting process of a Web Page to a fingerprint in the ISA system. Generating a *Web Page Fingerprint* involves three stages:

1. Stripping non-alphabetical characters
2. Creating a frequency list
3. Noise-Word filtering

6.3.3.1.3.1 Stripping Non-Alphabetical Characters

The first phase is to refine the text by removing non-alphabetical characters. All these non-alphabetical characters are replaced with space characters.

6.3.3.1.3.2 Creating Frequency List

When dealing with document files, it is vital to choose how to take care of and define single words. In this research, words are the smallest units that make up an unbroken text. So a word can be defined as an uninterrupted string of alphabetical characters, including all national variants, as defined in the ISO-8859-1 (extended ASCII) standard, while not having any other characters such as space, tab, punctuation and numbers. To locate each word, the ISA scans the text stream until an alphabetical character is found.

The ISA continues scanning until a non-alphabetical character is found, which marks the end of the word. This process recurs throughout the entire text of all web pages in the Internet domain corpus to generate a word frequency list, the *Web Page Fingerprint*.

At this point the processed text is passed to a program which generates a word frequency list, i.e. a list of all instances of a word in the entire body, together with the corresponding number of occurrences (word frequencies). All the uppercase letters are transformed into lowercase before this process begins. This list is sorted in alphabetic order, with the frequency number of each word is next to it. Table 6.1 shows a list of words from an experiment run by the ISA.

1081	access	921	directory	1271	model
4994	agent	1225	document	163	network
1191	application	115	files	167	number
1154	artificial	956	group	1555	program
94	class	1999	information	1126	retrieve
104	command	2855	intelligent	1006	search
97	control	1655	internet	123	set
3126	data	1343	key	104	statement
1076	database	1109	library	3076	system
940	description	125	message	39	table
1192	type	109	user	1746	web

Table 6.1 Frequency list words on the www.aaai.org web site

As can be seen in table 6.1 above, the most common words in the document body are *agent, data, system, intelligent, information, web and internet.* By looking at the other words, it is obvious that much of the material at this domain is leaning towards Internet technology.

6.3.3.1.3.3 Noise Word Filtering

The Agent is faced with another problem: words that have no contextual meaning, such as conjunctions, prepositions and pronouns, (see Table 6.2). These word are generally not helpful when attempting to model the contextual or semantic relationships between the documents, since they do not add any meaning whatsoever to a piece of text in the ISA Fingerprint model. If they are left in the text, they slow down the process.

A	AGAIN	ALTHOUGH	ALONG
An	AGAINST	ALWAYS	BUT
ABOUT	ALL	AM	BEFORE
ABOVE	ALLOWS	AMONG	BY
ABSOLUTE	ALMOST	AMOUNGST	OF
ACROSS	ALONE	AMOUNT	THE
AFTER	ALREADY	AN	WHAT
AFTERWARDS	ALSO	AND	WHY

Table 6.2 An abstraction of the Noise-word-list

The Fingerprint (word frequency) list is simply matched with a list of Noise words, and all co-occurrences result in the removal of that word from the Fingerprint list.

Since most of these words are relatively common the total number of words removed from the corpus is significant. It was noticed that it is useful to remove other words that do not add any value as well.

For example non-alphabetical words (excepting those with and embedded hyphen), words consisting of more than 25 or less than 3 characters, and words that contain 3 or more repeated characters (it was observed in this study that some web pages contain uninformative words).

6.3.3.1.4 Process Two (P2) for Collection Object

Process two, in summary, takes care of the following:

- Creating the Fingerprint for FP_0, FP_{-1} and FP_{+1} ,
- Saving all Fingerprints for the search function.

6.3.3.1.5 Constricting FP_0 ,FP_{-1} and FP_{+1}

The screen shot of ISA for generating a process of Fingerprint for FP_0 , FP_{-1} and FP_{+1} is shown in Figure 6.9. In this section the steps taken for creating FP_0, FP_{-1} and FP_{+1} are explained

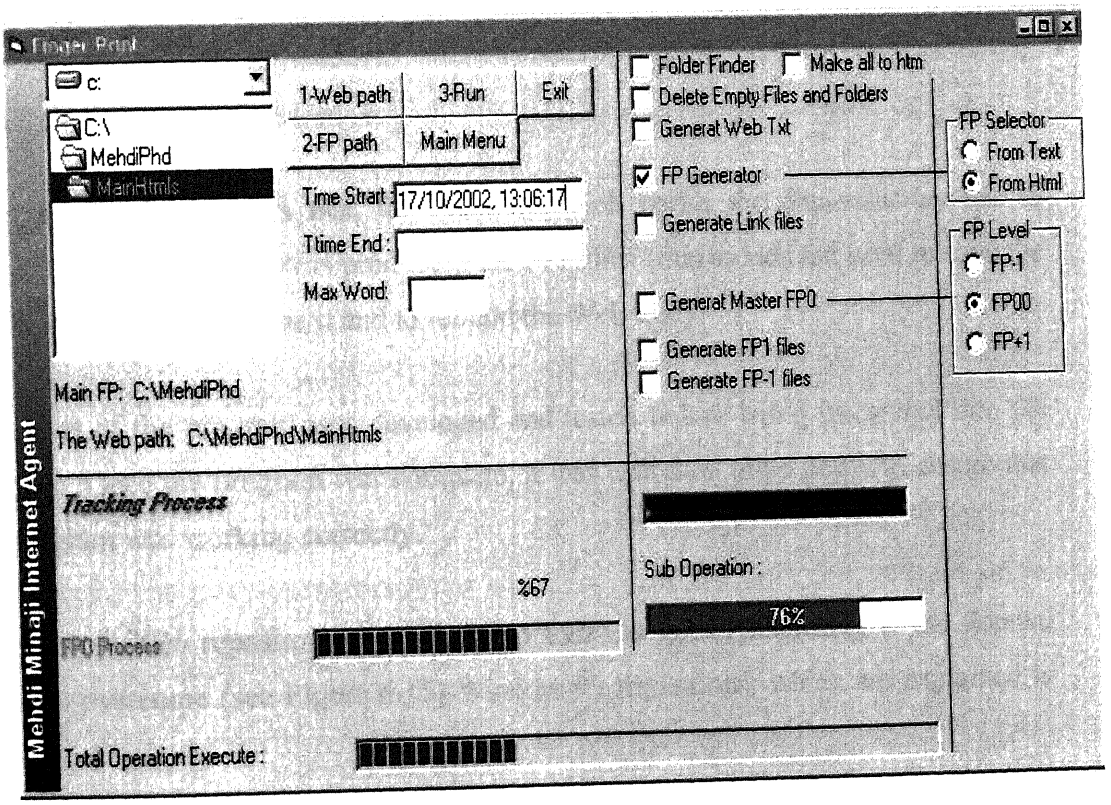


Figure 6.9 ISA Fingerprinting interface

Creating FP_0

Creating FP_0 consists of three actions:

- Generating text file from HTML files (see appendix A Figure A.1)
- Generating *"fpf"* file from text file (see appendix A Figure A.3)
- Generating FP_0 file from *"fpf"* file (see appendix A Figure A.4)

Each one of these actions can be executed individually or can be carried out automatically after each other. A sample of each kind of the above files are shown in appendix (A). Once FP_0 files are generated, ISA can make FP_{+1} files and FP_{-1} files using hyperlinks of '1-predecessor' and '1-successor'. .

6.3.3.1.6 FP repository collections

To make the FP process fast, the function which builds the data-structure has to remain "active" while ISA is producing the FPF files (this avoids the need to save the data to the disk, to re-parse it and to rebuild the Web tree).

Each part of the program was developed and tested before being integrated into the program. Once the program was complete, it was tested on other pages to ensure that the program was working correctly.

The *Sample Web* repository is divided into four categories according to the domain name to overcome (see Figure 6.15) Window 95 limitations, which are explained in section 6.4.

The four categories for the *Sample Web* are:

- The domain name does not start with "WWW",
  - ➢ the folder name is called 'not_www'
- The domain name starts with "WWW"
  - ➢ the first character of the name starts with one of the characters A to L, thus the folder name is called 'ww_a_L
  - ➢ the first character of the name starts with one of the characters M to S, thus the folder name is called 'ww_m_s
  - ➢ the first character of the name starts with one of the character T to Z, thus the folder name is called 'ww_t_z

6.3.3.2 Searcher Object Decomposition

The *Searcher* object is the core of the ISA system (see Figure 6.10) which gets search result and decides the appearances of search result and in which ways search result will be sent to the final output to the users.

User can choose one of the FP_{-1}, FP_0, and FP_{+1} of the *Sample Web* repository, from user interface this command pas to the *Searcher* object in order to locate the pages, which match with the search term to satisfy user's query.

Figure 6.10 shows the ISA search interface, in which the searcher object has been implemented to support keyword query for the FP_{-1}, FP_0, and FP_{+1} of the *Sample Web* repository

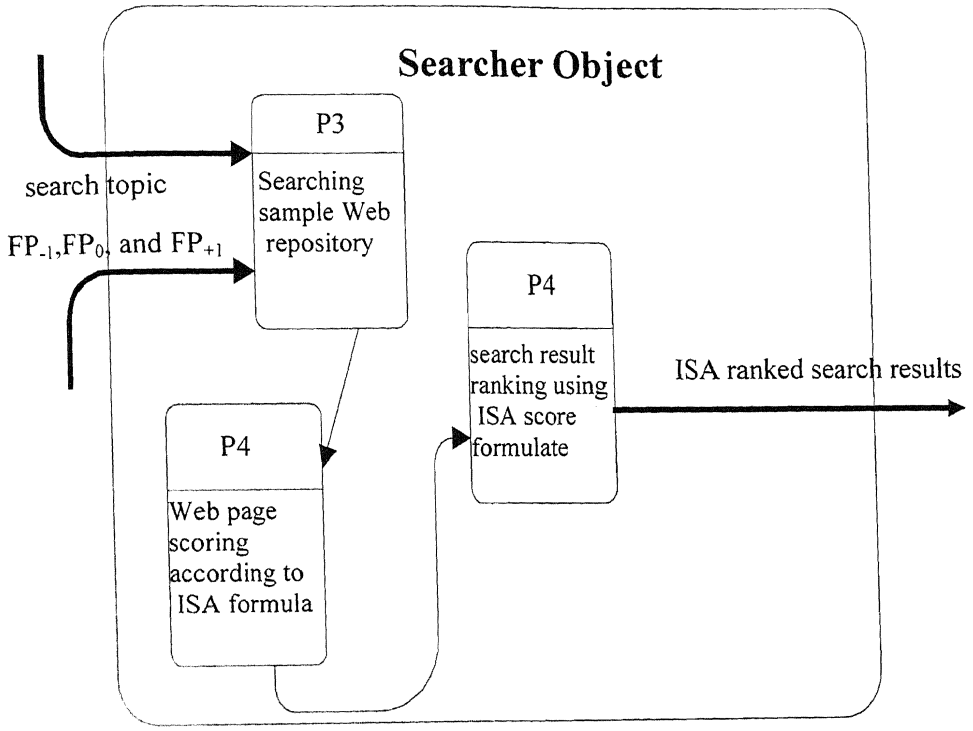


Figure 6.10 Decomposition of Searcher object

The *Searcher* object collects the outline information from the *Interrogation* object to figure out adaptation instructions and layout rules that will present the experimental results in an efficient way for further investigations.

The *Searcher* object consists of three steps:

- The first step is the Decision process (P3, see Figure 6.10) in this process, binary search method is used (binary search is a search algorithm for searching a set of sorted data).

- The second step to be taken is the Preference process (P4_a). In this section priority scores are assigned to contents according to the search topic's preferences.

- The third step to be taken is called the Ranking (P4), which decides if there is a better score for displaying these contents on the current list, by choosing from a sorting list (the order of their score of fingerprint) that may best represent these contents on the result list.

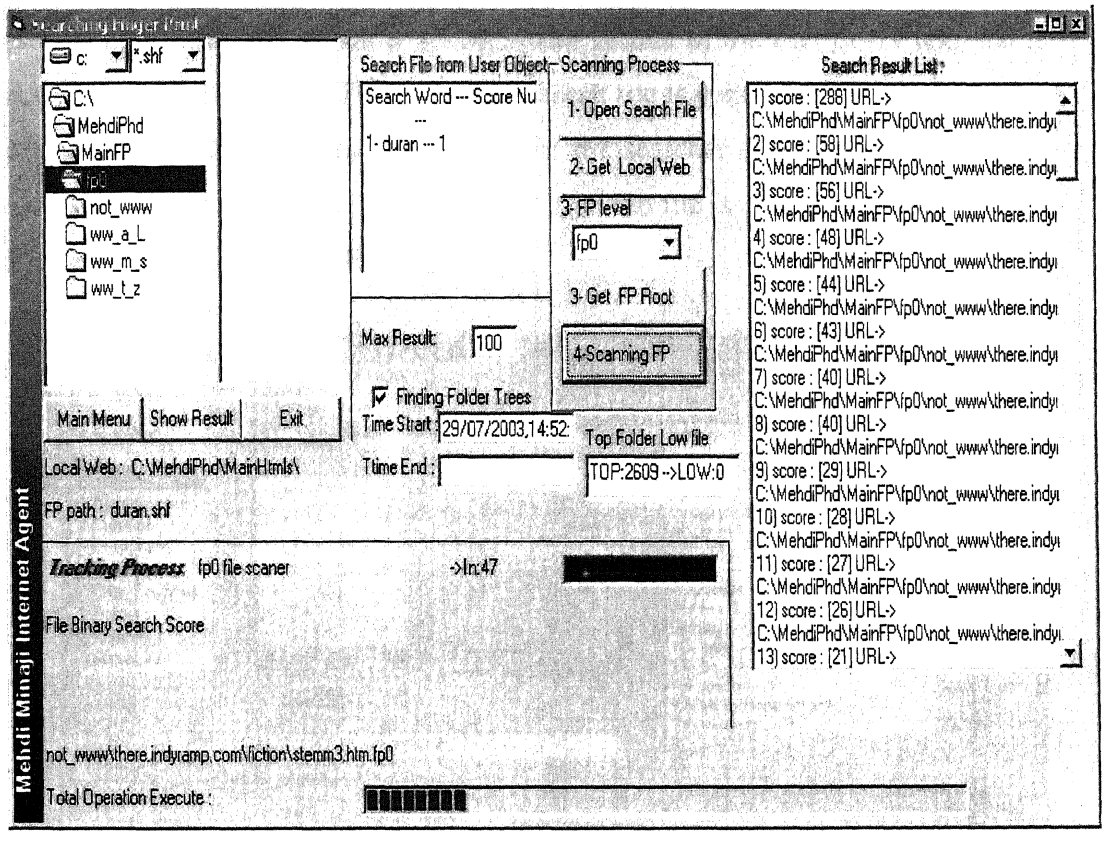


Figure 6.11 ISA search interface

6.3.3.3 Interrogation Object Decomposition

The *Interrogation* object is the interface between the ISA system and the user (see Figure 6.12).

Firstly in this process a signature file is generated (P5, see Figure 6.13) for representing a user query, which is a fingerprint format of the user query text and is saved with a 'shf' extension. Thus after the search file is generated for the query then it pasts to the searcher processors. The job of the searcher object is to evaluate the search file as if it is a Fingerprint's file. After this the file is compared with the other FP files present in the *Sample Web* repository.

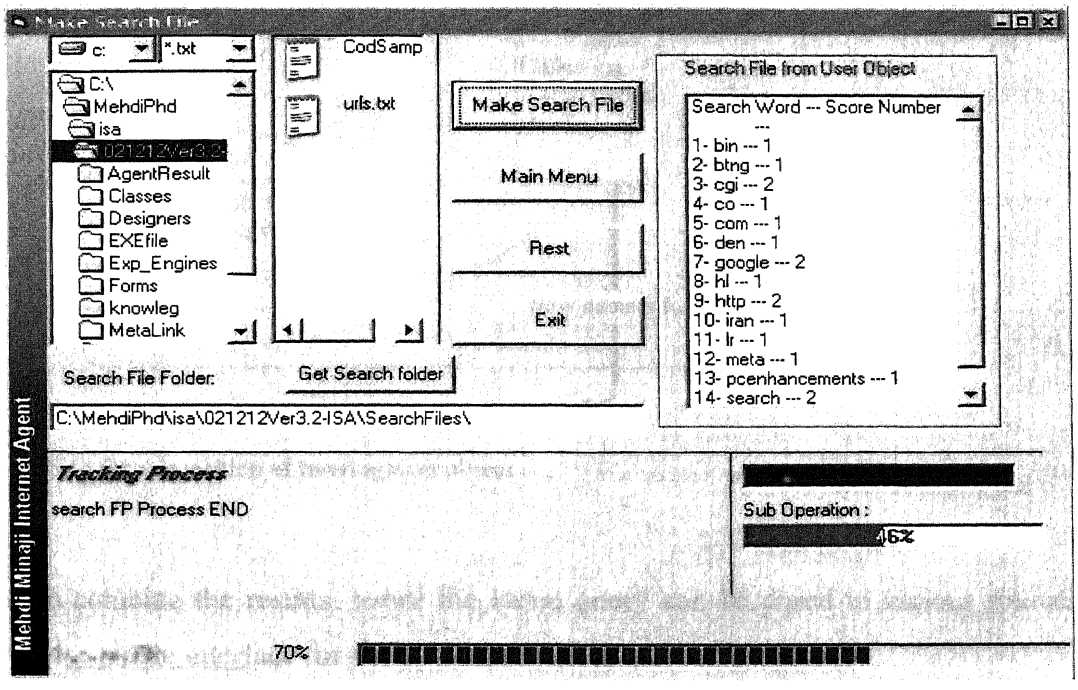


Figure 6.12 Search file maker interface

In addition to providing a specific search interface, the ISA acts as an Information Retrieval interface that may be used to experiment with different ranking and query configurations and with other commercial search engines (see Figure 6.14) to compare their results.

For the data collection for the *Sample Web* and for comparison of search engine results an interrogation object is used.

The interrogation object sends a given query to several search engines and collects the answers. After collecting the answers it combines them, while the collecting process is executing (P6, see Figure 6.12).

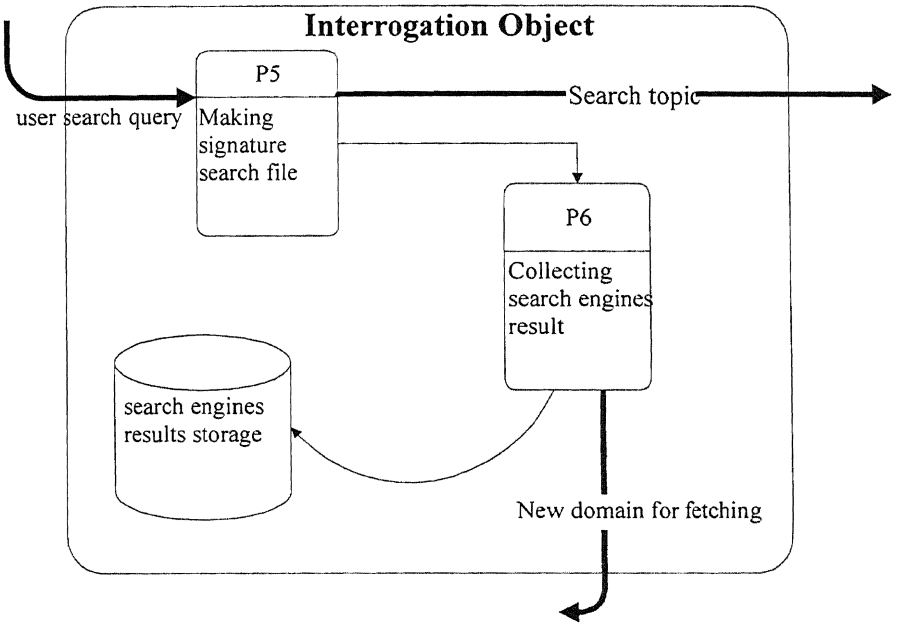


Figure 6.13 Decomposition of Interrogation object

P6 can combine the results; hence the same query can be posed to various sources through a single interface for further examination and comparisons.

The major challenge for this process is that each search engine has its own specific query language and search engines often change their query languages. Therefore the collecting process translates the user query to the specific query language used by each search engine. In this research four major search engines are used for comparison; Yahoo, Alta Vista, Google and WebCrawler.

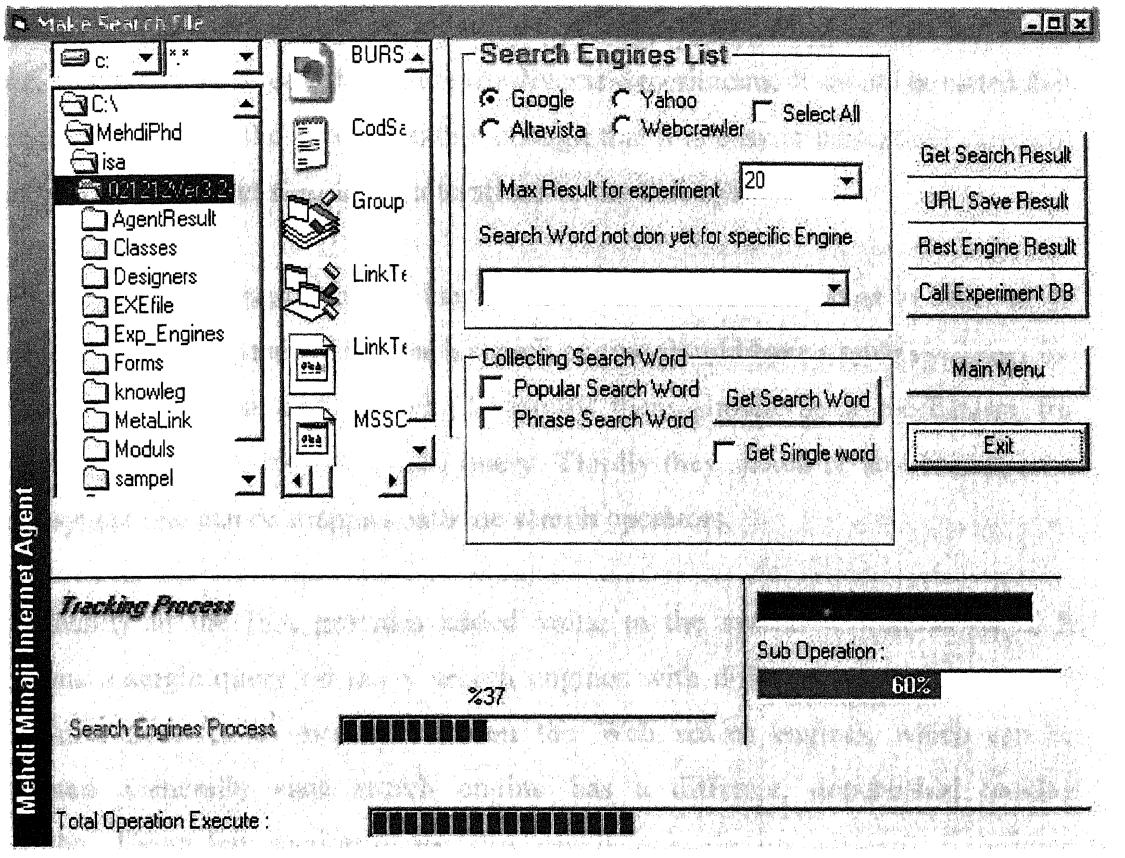


Figure 6.14 ISA search engines interface.

After the search engines results are collected URL from their result can be used for following processes:

- *Domain* process: This is to select the main domain of each collected URL and list them in an order, so that the WWW object can fetch web pages.

- *Scoring* process: This stage is to assign a score for each collected Web page, so that further examination and investigation can take place on how the ISA evaluates these Web pages.

- *Storing* process: This is to store all the information about each result (e.g. ranking position on search engine, ISA score) which can help in the process of comparison during the experimental stage.

The search engines results can be sorted by different attributes such as host, keyword, score, which can be more informative for diverse experiments. It should be noted that the code structure of the ISA is flexible enough that it is easy to incorporate different search engines without significant alterations to the ISA system.

There are minimum requirements that each search engine must meet in order to be supported by ISA. These are that each search engine should have a ranking system for results returned from each search. Secondly there should be a mechanism for identifying the search terms for each query. Thirdly they should be able to support a query syntax that can be mapped onto the search operators.

This facility in the ISA provides added value to the system in various ways. It performs a single query on many search engines with different query syntaxes. So sometimes there is an overlap between the Web search engines, which can be examined. Generally each search engine has a different, unpublished ranking algorithm. Using this section of the ISA can help in the investigation of different search techniques on the Internet.

6.4 Challenges and Assumptions

ISA is based on certain assumptions. For example a misspelling in the text may cause the search tool to not recognise a relevant page. Misspelling can be seen as a minor problem which can be solved by a few corrections, but when it comes down to extracting information from the HTML page the PDF, ASP, PHP and other files are not included in automatically extracting information by the tool. So to overcome this problem with these kind of formats, there needs to be an automatically generated system of cross format recognition. In this system only those web pages were accessed, which were free to be accessed and were written in English while being formatted in HTML were accepted. Special care was taken to ensure that none of the web pages in the *Sample Web* came from the same Web site as any other Web page which already exists in the test set. Web pages with null characters are removed.

As for removing material from the Web page, soft directory links and path remapping features in an HTTP server are used to find an infinitely "deep" Web site and remove them from the *Sample Web*. In order to prevent the "infinite site" problem a test of simple URL length or the number of slashes in the URL is carried out. Due to the nature of the Internet no automatic system can be perfect to detect all problems. Therefore a checking system is introduced to check for any new problem, for example in many instances the lower level of the sub-directories are removed manually in order to make the system running efficiently

The Figure 6.15 explains how the *Sample Web* is organised (see appendix A). Because of the limitation of Windows, which does not allow more than 10000 items in each directory, also the number of each sub directory in a single directory is also limited [Jung and Boutquin, 2000].

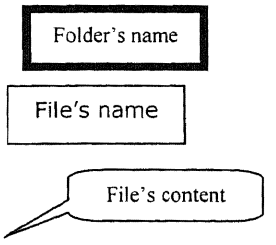
6.4.1 Experimental setup

All the experiments were conducted on a IBM compatible PC equipped with a single 1Ghz Pentium III processor, 512MB RAM and 2 drives for internal storage (an 8GB main drive and in addition a 20GB second drive). The decision was taken to stay with the Windows 95 operation system used when the project stared in 1998. However, for information tests were carried out with Windows 98 and Windows XP. ISA worked well on these operating systems.

The machine was connected via a 100Mbit/sec Ethernet card to the University of Luton LAN for Internet access through JANET. Microsoft Windows 95 was installed on the machine. The ISA was implemented in Visual Basic 6.0 and compiled using the Visual Studio compiler. A sample of the source code for ISA is presented in Appendix C. Microsoft Access 97 and Microsoft Excel 97 were used to produce all the graphical output, using VB to link with Microsoft Access and Excel.

Key shape:

- This shape represents a Folder
- This shape represents a File
- This shape represents a File' content



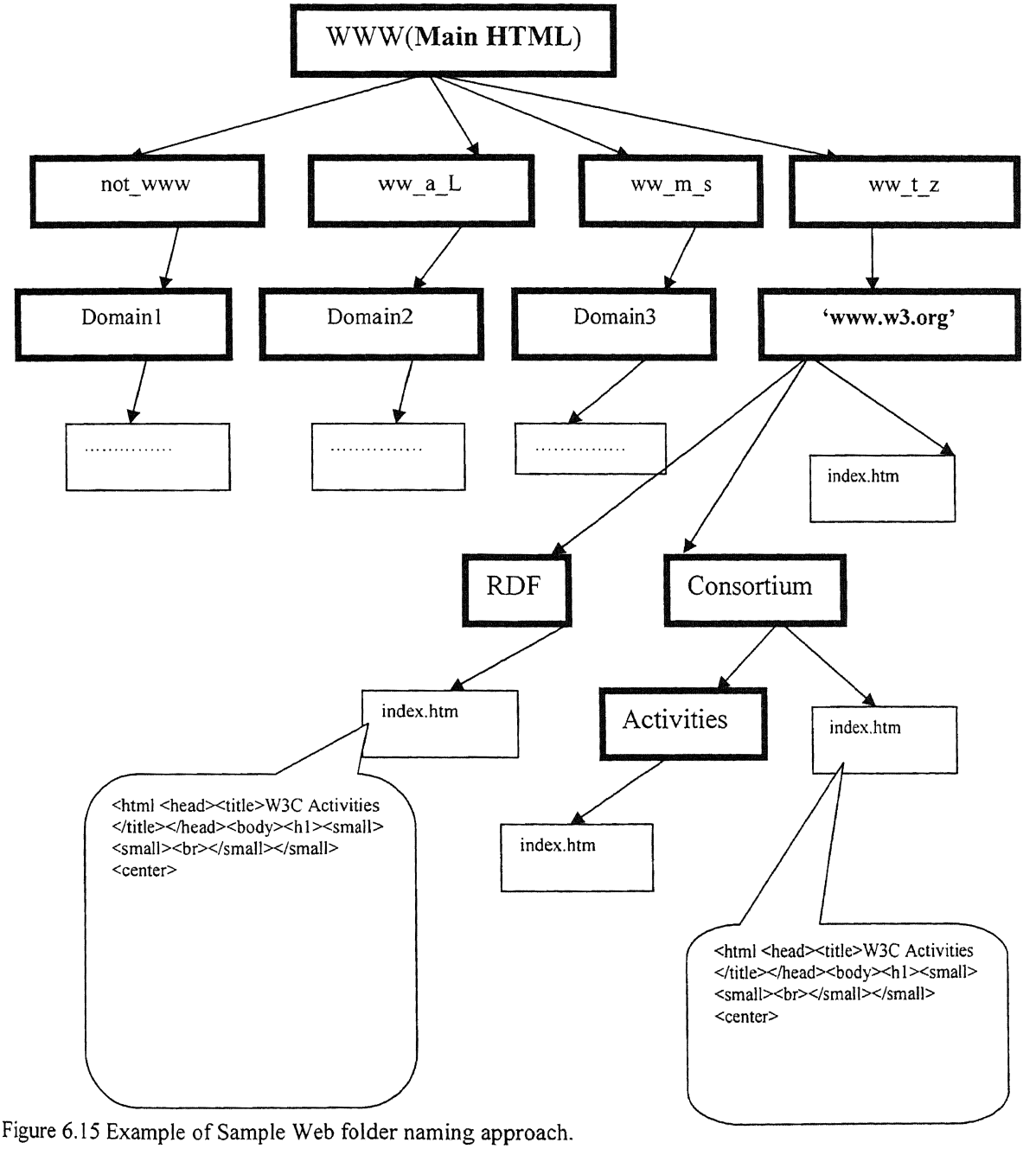


Figure 6.15 Example of Sample Web folder naming approach.

6.5 Summary

The focus of this chapter has been on explaining the main structure of ISA and how the input and output of the system interface with other programs. The process of Noise Word (words that don't have meaning out of a sentence) is described and how ISA selects the Noise Word and their details are discussed. Chapter 7 explains the results that are obtained from the experiments that are explained in chapter 5 by means of using ISA and outcome of these experiments are discussed in more details in discussion section of chapter 7.

CHAPTER 7

7 Experimentation and Discussion

This chapter describes a series of experiments that investigate the research topics discussed in chapter 5. Measurments of the relevance of the search results of four common search engines (Google, Alta Vista, Yahoo and WebCrawler) are compared with ISA. This chapter illustrates the three methods for classifying web pages: FP_0 , FP_{+1} and FP_{-1} used by ISA and also discusses the measurement of the search progress after the search has been formulated.

The chapter is divided into the six main sections; Section 7.1 is a general explanation of the experimental design. Section 7.2, describes all the experiments carried out by the search engines. Section 7.3, explains some of findings about the *Sample Web*, such as file size and the number of folders. Section 7.4, describes the main experiments investigating the approaches to searching documents incorporated into ISA. Section 7.5, evaluates the search engines' results compared to ISA and the last section reviews the experimental results obtained by the research.

7.1 Experiment Outline

The results of the experiments are presented so that the web pages identified as relevant by ISA can be compared to the results of the search engines, evaluating for the similarity between the pages found by them. In this study also measurements of the progress of searching FP_0, FP_{+1}, and FP_{-1} on *Sample Web* are examined and compared to each other.

As explained in chapter five, this research involved several experiments:

- A comparison of the results obtained by different search engine.
- Measurements of the progress of searches, against the effort required.
- Measurement of the results needed to accumulate a certain percentage of the possible score and the 'cost' of obtaining these in terms of files analysed.

7.2 Results of Using the Search Engines

This section reports on the results returned by the four search engines used to develop the *Sample Web*. Table 7.1 shows the number of web sites downloaded for each search term against search engine. Each search engine was requested to give fifty sites, but because of the difficulties associated with multilingual text, sites depending heavily upon alternate character sets or foreign languages were excluded from this work.

		Search Engines				
		Google: **g**	Alta Vista :**a**	Yahoo : **y**	Webcrawler :**w**	
Search Terms	**Duran**	48	50	49	50	
	Gemba	47	50	18	50	
	Liquid Marbles	50	50	50	50	
	Lean	49	50	50	50	
	Triz	50	50	15	50	

Table 7.1 Results returned (English text only). Number of English language results returned by the four search engines for the search terms *Duran*, *Gemba*, *Liquid Marbles*, *Lean* and *TRIZ*.

The figures in Table 7.1 show that Yahoo did not return the full fifty results asked for at the time of the search. This was as expected because the words Gemba and *TRIZ* were known to have been introduced from foreign languages, Japanese and Russian respectively. Both are used in manufacturing management circles all over the world. It is interesting to note that only Yahoo and to a much smaller degree Google had this property.

It seems that the others may have limited exposure to non-English speaking web sites. More recently, however, Google has added a feature allowing the user to filter results according to their language (see http://www.google.com/language_tools?hl=en).

One of the most surprising results of the investigation was the relative lack of overlap between the results of the different search engines (see discussion in section 7.10).

Armed with the results obtained from the various search engines and having filtered those results to remove 'problematic pages', the resulting web sites were downloaded.

As explained in chapter 5 the downloaded structure for *Sample Web* was chosen to be similar to the structures adopted on the Web in terms of domain names and directory structures. There were a number of technical difficulties to do with limits on files and directory limitations in Microsoft Windows. These were reported in chapter 6.

7.3 The Size of *Sample Web*

The size of the structures that were built during this research are shown in Table 7.2.

The size of the sample is measured in terms of number of files (HTML ones only as others such as .gif and .mp3 are beyond the scope of this work), the folders (which include domains and sub-domains) and the physical size on the disk.

For the sake of clarity a typical HTML file on the sample web is referred to as 'example.htm'. The figures given for HTML in Table 7.2 are for the original HTML files downloaded from the Internet. The FP_0 fingerprint of this contains a sorted list of all the words contained in example.htm with their frequencies. The FP_{+1} fingerprint is the combined FP_0 of all the files (in the local web) that are pointed to by the original file example.htm. The FP_{-1} fingerprint is the combined FP_0 of all the files pointing to example.htm. Finally there is a structure called 'links' which contains just the links in the HTML files.

Name	Files	Folders	Domains	Size (MB)
FP_0	118857	37730	8830	358
FP_{+1}	13042	5727	4783	70
FP_{-1}	101307	29552	6125	914
HTML	82967	43350	8830	1220
Links	76008	36302	8830	162

Table 7.2 Sizes of data structures in terms of files, folders, and domains. Sizes of tree like data structures used in the research in terms of files, folders, domains and size on disk in MB.

From Table 7.2 it can be seen that 1220 (MB) of HTML generated 358MB of FP_0 files, 914 MB of FP_{-1} and 70MB of FP_{+1}. If a file linking to an html file does not exist in the *Sample Web* it cannot be included in the calculation of the FP_{+1} structure. On the other hand if a target file does not exist in the sample its FP_{-1} file or at least the approximation to it that is implied in the html files, it can still be formed.

7.4 Search Progress

To illustrate the search progress, a typical search was carried out for the term '*Duran*', which has a high frequency in the *Sample Web*. The progress of the search is displayed in Table 7.3.

The more important columns of the table are 'cumulative score' and 'HTML Files Opened' which shows the way that the total score increases as a function of HTML files accessed.

As the search is directed towards those directories and files which contain the most occurrences of the target word, it would be hoped that once the search has navigated to the part of the *Sample Web* that contains target pages the score would increase rapidly with html files opened.

No	Score	File opened	Cumulative Score	Folder's files opened	HTML files opened
1	1	3912	1	3911	1
2	2	4767	3	4763	4
3	3	4769	6	4764	5
4	48	4771	54	4764	7
5	40	4772	94	4764	8
6	4	4774	98	4765	9
...	...	...	...	...	...
...	...	...	...	...	...
3400	1	15650	28301	9609	6041
3401	1	15652	28302	9610	6042
3402	1	15655	28303	9611	6044
3403	1	15660	28304	9615	6045
3404	1	15667	28305	9621	6046
3405	1	15668	28306	9621	6047

Table 7.3 Table showing search progress as target words are found in an ISA FP_0 search for Duran.

However, some directories, which are rich in the target word, may have files which are themselves not rich in the target. This can clearly be seen in the table where the first three html files contain *Duran* only once, twice and three times respectively. The fourth file, however, contains *Duran* 48 times.

As was pointed out earlier, although occurrences of target words are being used to direct the search, it may be that the word appears in files that are not of value to the searcher. This might happen in a number of ways, the most annoying is probably the inclusion in a file of terms that, despite occurring many times, bear no relation to the main content of the file but are added to attract searchers. This 'spam' is an increasing problem for searchers and search engines alike.

Another way of visualising the progress of the search is by means of a graph showing cumulative score (word count in the examples) versus files opened (or effort expended). These graphs are called 'type A' in this thesis.

Here by 'files opened' is meant the number of html fingerprint files that were opened during the search. This number is used as a measure of effort expended. Percentages are based upon the total number of files that were opened to find all of the target words during the experiment.

The cumulative score plotted against effort (Figure 7.1) shows how the total score increases with length of search. As such it shows how the search progresses. Both axes have been normalised to aid comparison. One may have expected that following the Pareto principle the curve would show about 80% of the score appearing after only 20% of the effort. Unfortunately things are not quite as simple as this because web pages are not as conveniently distributed. To obtain about 80% of the score about 50% of the files need to be searched.

In this graph (Figure 7.1) a vertical jump represents a file containing an appreciable number of search terms whilst a near horizontal line occurs where few terms are found in a sequence of files. Because of its exhaustive search process ISA returns all of the files containing the search phrase in decreasing order of the number of occurrences of the search phrase and this list can be used to find out more about the search space.

Sample Graph of Cumulative Score against Effort Expended.

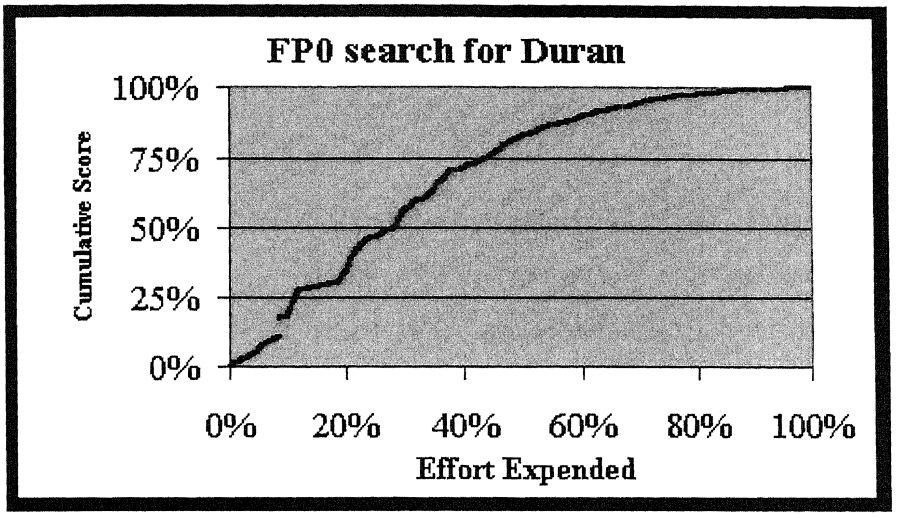


Figure 7.1 Search progress: Cumulative score vs effort expended. Graph showing how score accumulates as html files are searched for the case of *Duran* using FP_0 to direct the search. This shows how return on effort declines after half of the search has been undertaken ("A").

Checking the total number of target words occurring in the *Sample Web* against the number of target words returned by ISA (derived by adding the scores of all the results) is used as a check that ISA works as intended and should find all of the occurrences of the search term.

How many results (as a percentage of all) are needed for a given percentage of search terms to be included in these results can be read from a graph of results needed against score required (see Figure 7.2 for example). This shows, for example, that 75% of the search terms occur in the top 25% of the results. This is not a graph of search progress as it requires all of the results to be on hand (at least in principle) before it can be calculated, it is a graph describing the way that target words are distributed throughout the search. This type of graph is referred to as 'type B'.

Sample Graph of Results Needed vs Score Required.

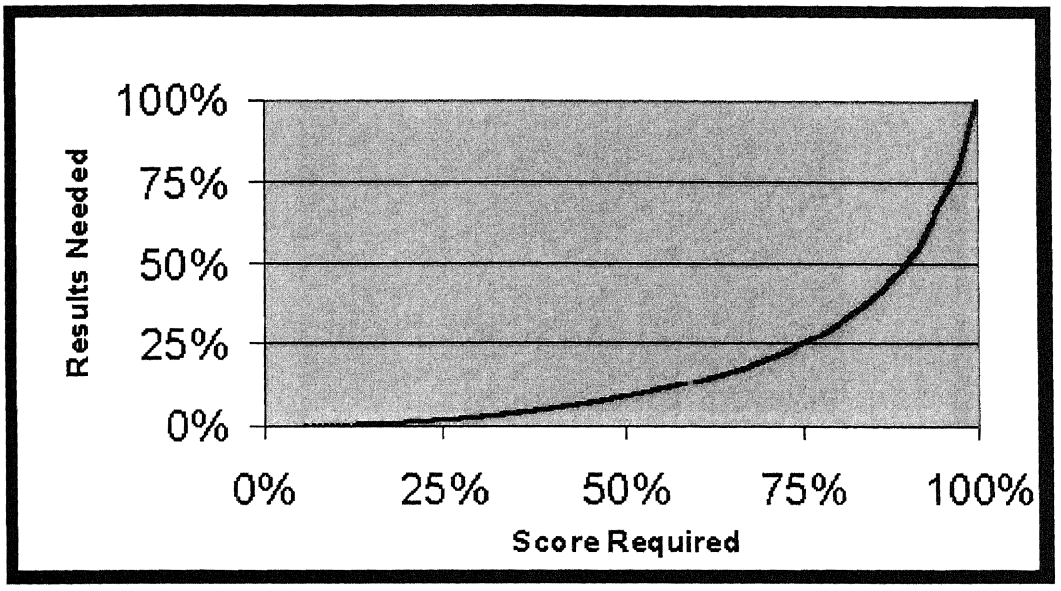


Figure 7.2 Results needed vs score required. Graph showing how many result files must be taken in order to have a particular percentage score in the case of *Duran* using FP_0. This indicates how search terms are concentrated at the top of the list of search results ("B").

This diagram is much nearer to obeying the Pareto principle.

Unfortunately the system cannot necessarily give the top 25% of the results without first searching through its entire web. What is needed is a measure of progress of an ISA search.

Although Figures 7.1 and 7.2 say much about the search space, a third (effort required against cumulative score) is also useful. Graphs of effort required versus cumulative score "best results" show the effort required (in terms of the percentage of html files) that must be searched to achieve a given percentage score when this score is required in the *best results order,* that is results with the highest number of search term occurrences first and this is shown in Figure 7.3.

("C") Effort required vs cumulative score (best)

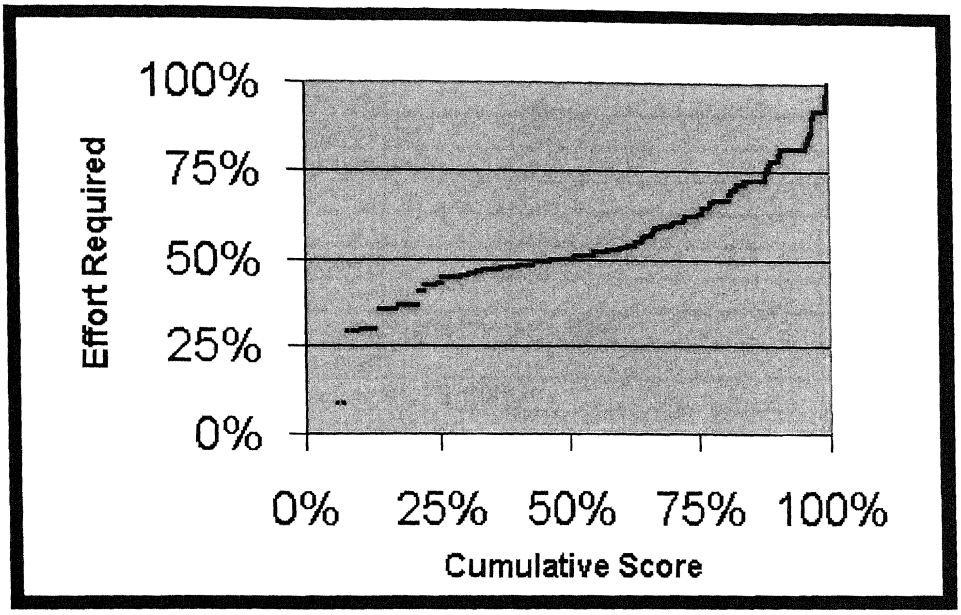


Figure 7.3 Effort required vs cumulative score "best results". Graph showing how many html files need to be visited in order to accumulate a particular score in the case of *Duran*, using FP_0. This shows three types of behaviour. At first (between 0% and about 20%) there is little 'return on effort'. This is followed by a flat region (between 20% and 70%) where much score results from relatively little extra effort and finally a region where again little return results from increasing effort ("C").

Thus ISA found the best 25% of its results only after searching through nearly 50% of its data. Thus this graph is a better representation of ISA search effectiveness than the cumulative score versus effort expended and results needed versus score required graphs because users usually prefer a small number of highly relevant files to a large number of less relevant files.

Up to now the progress of a search has been measured only in terms of the html files searched. This is only part of what goes on. Folder fingerprints need also to be looked at and Table 7.4 shows the ISA score for folders when the search term is *Duran*. Those folders, which have zero score, have been omitted from this table.

The first line of the table shows how many times the term appears in the sample web and lines 1 to 6 its occurrences in the sub webs. These figures are used by ISA to direct its search towards the directories most likely to contain the search terms.

No	Folder address	Folder Score
	Whole web	27847
1	not_www\	13376
2	ww_a_l\	9875
3	ww_m_s\	4312
4	ww_m_s\www.schott.com\	3181
5	ww_a_L\www.geocities.com\	2826
6	not_www\there.indyramp.com\	2822
...	...	...
...	...	...
1614	not_www\j.talkcity.com\	1
1615	not_www\j.talkcity.com\ht\	1
1616	not_www\j.talkcity.com\ht\guestbookdisplay\	1
1617	not_www\geocities.com\	1
1618	not_www\geocities.com\rio73_2000\	1
1619	not_www\duran.hypermart.net\	1
...	...	...

Table 7.4 Table showing folders and their scores as the search progress.

ISA does not have this information at the start of the search (although in a production engine one might choose to have these in tables). Instead ISA builds up this list as the search progresses by exploring those folders that show the most promise. In this way ISA might be described as using a heuristic search strategy.

7.4.1 The Relationship between Results, Score and Effort

This section considers the results relating *results returned* with *effort expended* and *score achieved* and is organised as follows:

(A) Results for search progress,

(B) The number of results needed for a given percent of the score,

(C) The effort needed for a given percent of the score.

Within each of these major sections the results for FP_0, FP_{+1} and FP_{-1} are presented in subsections. In these subsections results for *Duran* are given, then a comparison of three words (*Duran*, *TRIZ* and *lean*), and finally averages for a set of classes of words are added.

The classes were chosen to contain words representing those that occur a given number of times in the *Sample Web*. (see table 5.2 for the words and their frequencies) The frequencies occurring here are very low compared with their values on the web itself. This is because the *Sample Web* is about $\%10^{-5}$ of the web (which contains about 167 Terabytes see section 3.2) and so any discernable trends occurring as the frequency increases would be of interest but must await further research.

7.4.2 The Terms Revisited

It is worth briefly reminding readers of the variables that are plotted on the graphs in these three major divisions and that is done in this section.

- **Effort Expended:** Effort expended measures the number of 'html fingerprints' that are processed. In a production search where indices are built to enable rapid search this measure will not be directly relevant. However effort expended is a rough measure of the complexity of distribution of score in the sample web.

- **Cumulative score:** The score associated with a page can be any numerical measure of relevance of that page to a search. Often it is important to look at how the total score accumulates as a search progresses or as an ordered list of results is traversed. A simple 'score', that of the number of target words occurring, is used throughout this research. Depending upon the application many other 'scores' are possible; for example a search might require results to be weighted according to their currency, length or the affiliations of their authors.

- **Ordered results:** In order to understand the distribution of score throughout the results they can be ordered in two ways. Firstly there is the order in which the results are found by the search engine. This will vary from search engine to search engine and may well vary for the same search on the same search engine at different times. When ordered results are mentioned here the order is the second ordering, that is of decreasing score.

7.4.3 Accumulation of Score as the Search Progresses

As described in section 5.2.1 the term 'score' is used to describe a numerical value associated with a web page given *a fixed search criteria*. In the examples that follow fingerprints FP_0, FP_{+1} and FP_{-1} are used, which are based upon a simple word count as score.

This section looks at how the score increases as the search progresses. Naively one might expect an 'excellent' search would follow a Pareto type of curve with a rapid increase in score at the beginning slowing down as the total score is reached.

An indifferent search might well be expected to follow a roughly straight line from (0, 0) to (100%, 100%) with occasional steps where a relevant area of the web is being explored. The result of any real search strategy is likely to lie somewhere in between.

7.4.3.1 FP_0

FP_0 represents a simple fingerprint based upon local score. At one level of sophistication one would expect that the higher the score the more valuable the result is to the user.

Figure 7.4 shows the search progress for *Duran*. Although the graph does resemble one for a subject obeying the Pareto Principle some features are worth noting. Firstly, at the start, the search does not seem to be going too well. Later at about 10% there is a sharp rise in the score. Similar features can be seen throughout the search although things seem to go more smoothly later.

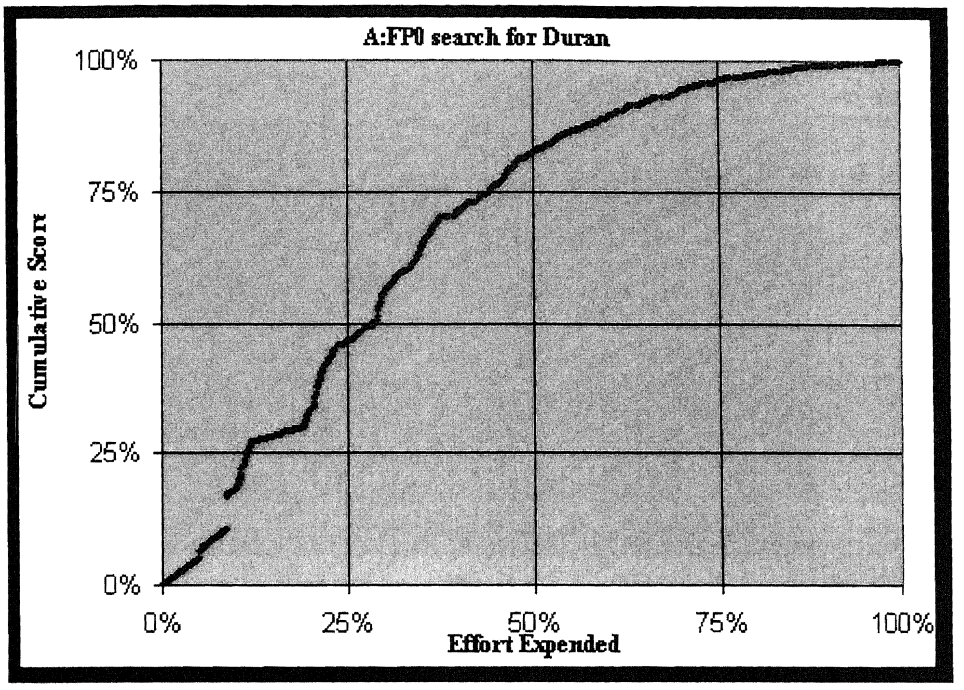


Figure 7.4 *Duran*: Search progress: Cumulative score vs effort expended. Graph showing how score accumulates as html files are searched for the case of *Duran* using FP_0 to direct the search. This shows how return on effort declines after half of the search has been undertaken ("A").

These artefacts are due to the detailed nature of the particular search space which itself depends upon the particular choice of search term and sample web.

Variation of the search progress between words *Duran*, *TRIZ* and *lean* is shown in Figure 7.5. It can be seen that although the three words make similar progress, each has a different path.

These differences are due to the particular words and pages downloaded. However as will become clear later, when these results are considered together with others, some of the differences are due to the differing frequency of occurrence of the three words.

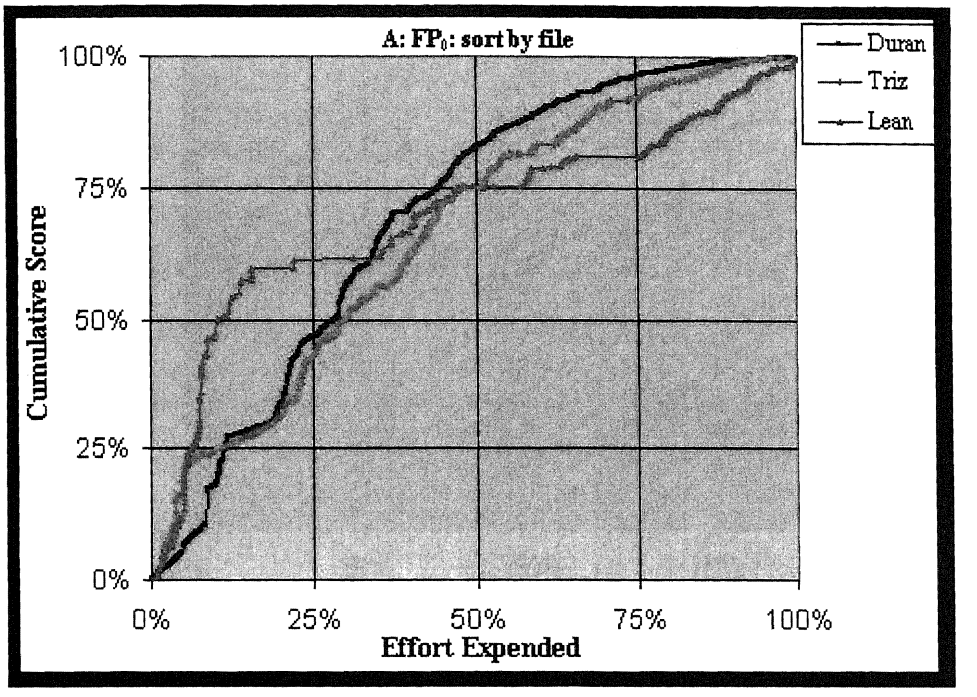


Figure 7.5 FP_0: Cumulative score vs effort expended of *Duran*, *TRIZ* and *lean*. Graph showing how score accumulates as html files are searched for the case of *Duran*, *TRIZ* and *lean* using FP_0 to direct the search ("A").

7.4.3.2 FP_{+1}

As detailed in section 5.1.2 the FP_{+1} Fingerprint of a file F combines the Fingerprints of those files pointed to by *F*. As such it gives an indication of the 'forward neighbourhood' of F in the sample web.

The FP_{+1} Fingerprint of a file may be a good indicator of its importance in guiding searchers towards the information that they require. Computations for FP_{+1} similar to those described above for FP_0 are reported here.

When a search is conducted of FP_{+1} similar results are obtained, although there are important differences as will be seen later. Figure 7.6 shows the search progress for *Duran* where sharp rises in score are followed by small plateaus.

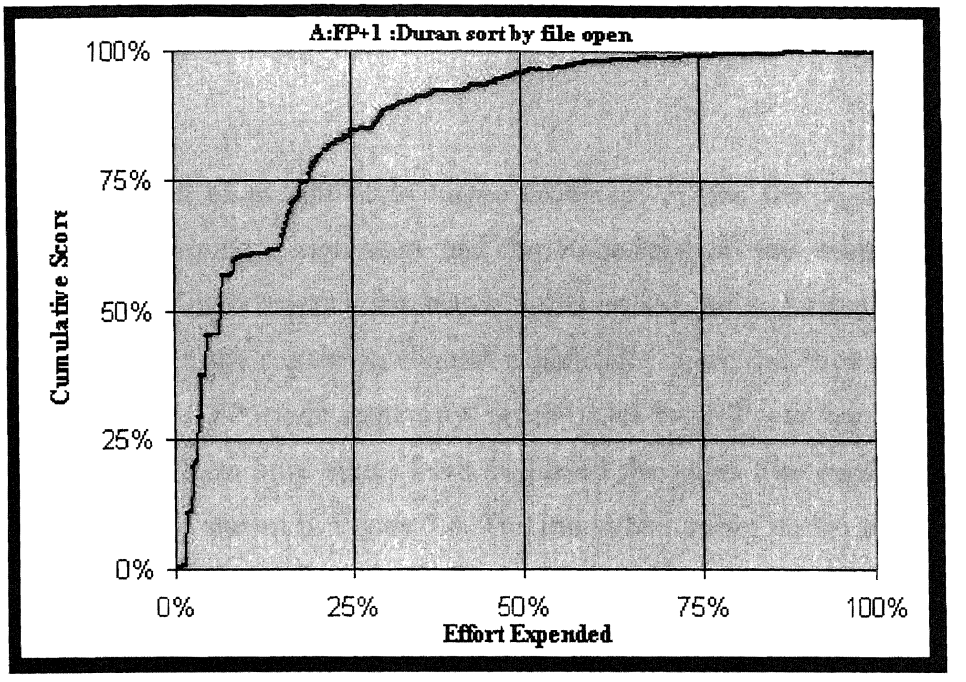


Figure 7.6 *Duran*: FP_{+1} Cumulative score vs effort expended. Graph showing how score accumulates as html files are searched for the case of *Duran* using FP_{+1} to direct the search ("A").

The 'stepping behaviour' observed in Figure 7.6 for *Duran* is apparent also for the cases of *TRIZ* and *lean* in Figure 7.7 but with a different scales.

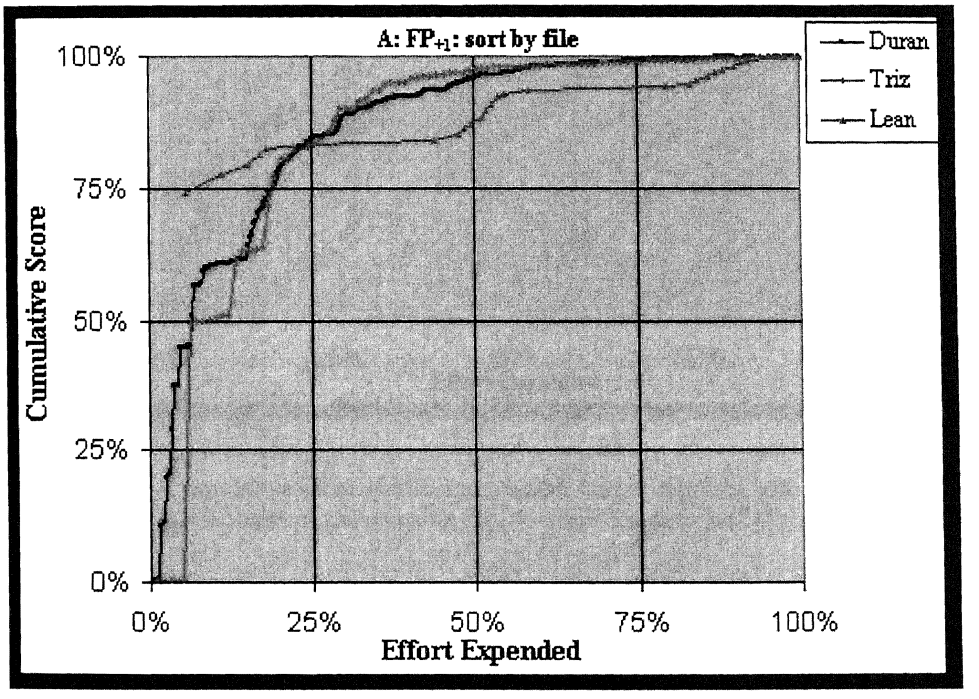


Figure 7.7 FP_{+1}: Cumulative score vs effort expended of *Duran*, *TRIZ* and *lean*. Graph showing how score accumulates as html files are searched for the case of *Duran*, *TRIZ* and *lean* using FP_{+1} to direct the search ("A").

7.4.3.3 FP_{-1}

FP_{+1} can be thought of as similar to 'name dropping' in that the page's author is responsible for obtaining a high score and the author can do this using spamming techniques such as including popular search terms or web sites, displayed in a way that hides them from the reader. In contrast a high FP_{-1} score indicates that a file is cited by many others. FP_{-1} is in some ways much better than FP_{+1} in that a high score on this measure reflects how others have evaluated the page. The results of a FP_{-1} search for *Duran* are shown in Figure 7.8. The individual points on the curve depend upon the particular files included in the sample web but the figure shows that about 75% of the score is obtained after only 25% of the effort has been expended.

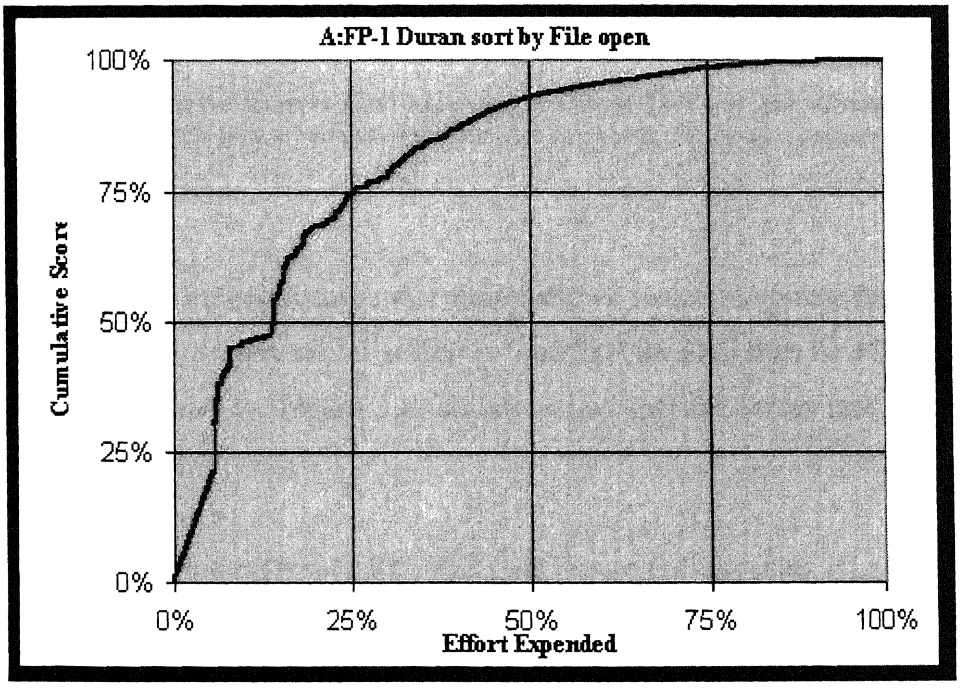


Figure 7.8 *Duran*: FP_{-1} Cumulative score vs effort expended. Graph showing how score accumulates as html files are searched for the case of *Duran* using FP_{-1} to direct the search ("A").

Superimposing the fingerprints FP_{-1} of the three words in Figure 7.9 shows that they have similar behaviour. Although large variations can be seen in this diagram it does indicate that the higher frequency words, listed in order of decreasing frequency in the legend, tend to have a higher cumulative score (%) for a given effort expended.

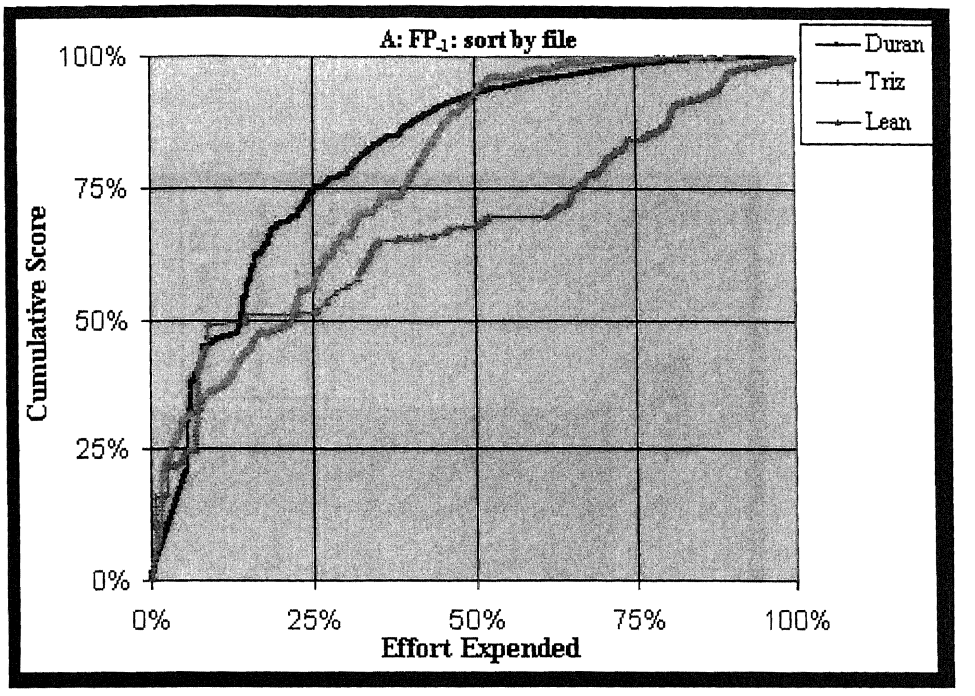


Figure 7.9 FP_{-1}: Cumulative score vs effort expended of *Duran*, *TRIZ* and *lean*. Graph showing how score accumulates as html files are searched for the case of *Duran*, *TRIZ* and *lean* using FP_{-1} to direct the search ("A").

A plot of the three Fingerprints on one graph shows some interesting features. Most notably it seems that in general the search is 'quicker' for FP_{+1} than for FP_{-1} , which is in turn 'quicker' than FP_0. Figure 7.10 clearly shows that the search rate, for *Duran* , $FP_{+1} > FP_{-1} > FP_0$.

Plots of the corresponding results for *TRIZ* (see Figure B.1 appendix B) are consistent with these findings. In the case of *lean*, however the curves for FP_{-1} and FP_0 cross a number of times and so no generalisation can be made.

These and other results reported later indicate that the results for lean suffer from it having a relatively low frequency in the *Sample Web* (*Duran* =28306, *TRIZ* =22031, *lean*=757, see Table B.1 appendix B).

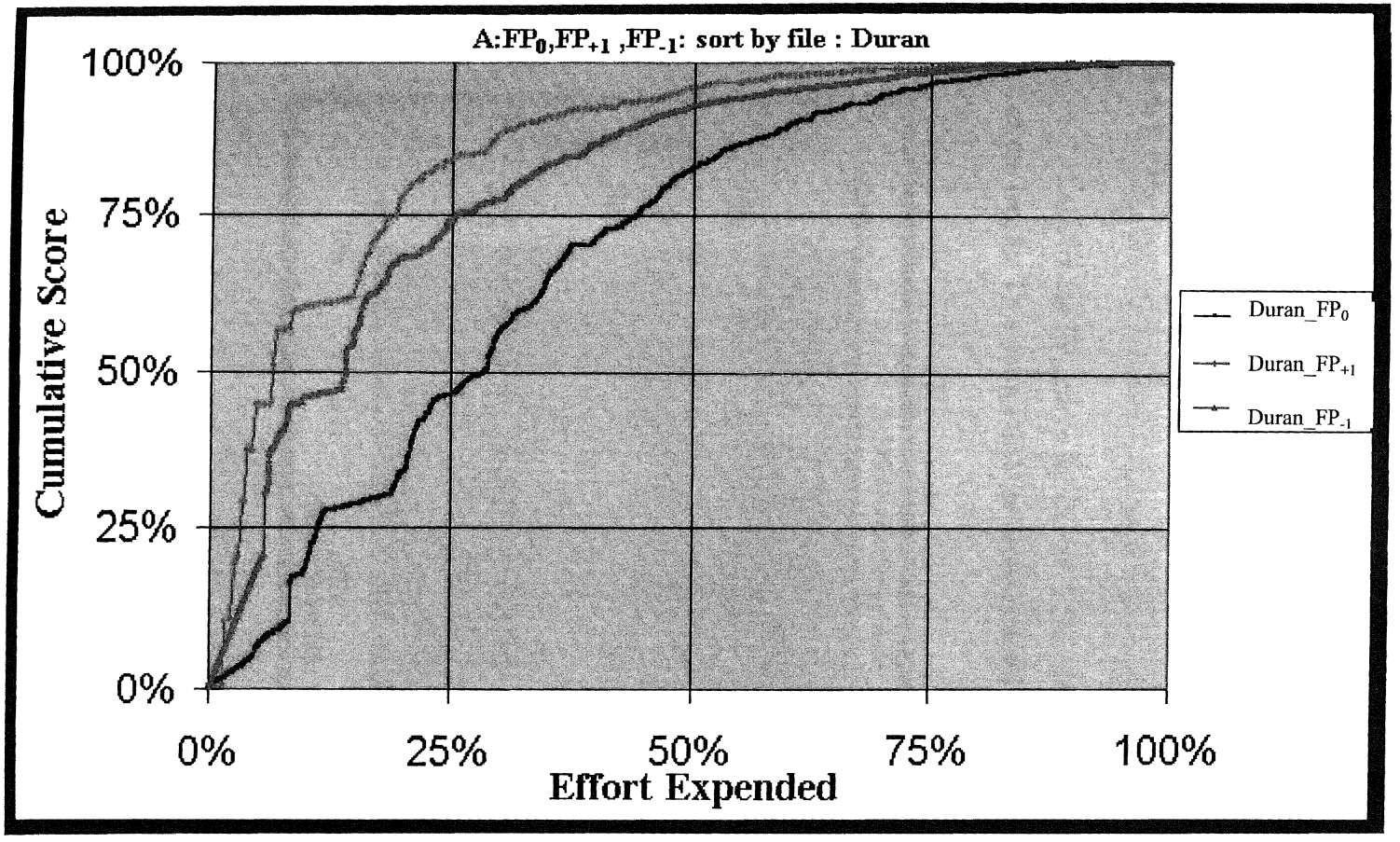


Figure 7.10 Combined graph of FP_0, FP_{-1} and FP_{+1} searches, *Duran*. Combined graph of ISA FP_0, FP_{-1} and FP_{+1} searches, showing score accumulating as a function of html files opened in the case of *Duran* ("A").

7.4.4 The Distribution of Score within the Ordered Set of Results

In this section the results for the way the score is distributed through the ordered set of results is examined. A plot of the number of results (in order) needed to achieve a given percent of the total possible score is the focus of study. This shows the structure of the sample web, and may give insight into the problem of searching the "real" web.

7.4.4.1 FP_0

Figure 7.11 shows the FP_0 curve for *Duran*. 75% of the score is contained in the first 25% of the results. This shows that the score is distributed Pareto-like throughout the results.

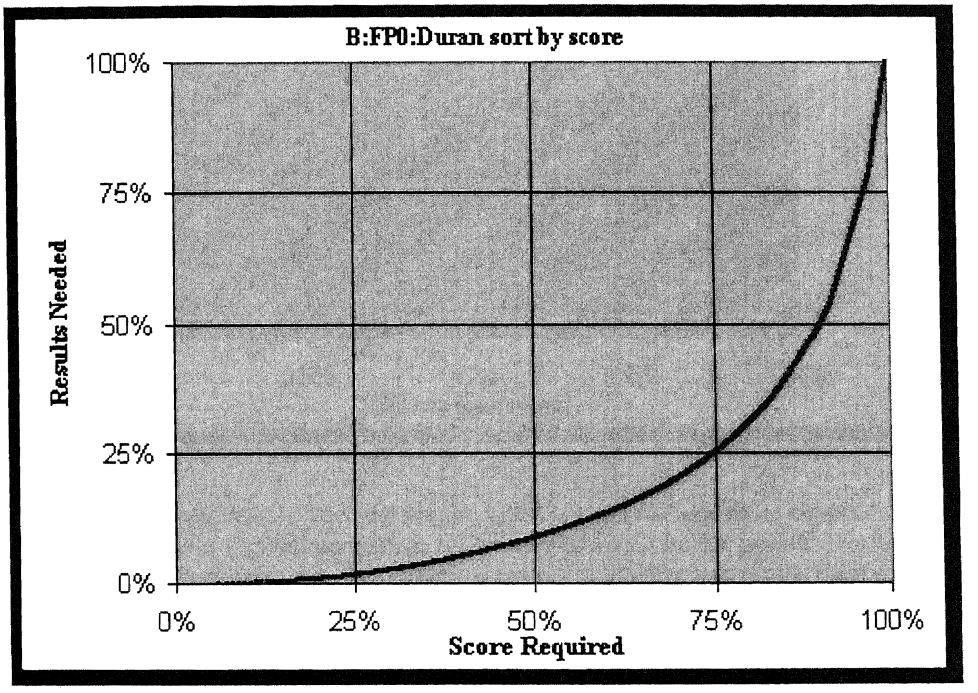


Figure 7.11 *Duran*: Results needed vs score. Graph showing how many result files must be taken in order to have a particular percentage score in the case of *Duran* using FP_0. This indicates how search terms are concentrated at the top of the list of search results ("B").

The curves for *Duran*, *TRIZ* and *lean* are plotted together in Figure 7.12. The curves show the same general shape, with those for *Duran* and *TRIZ* being very close. The curve for *lean*, however, does show some interesting features. Although the shape is similar to the other two curves it is clear that for percentages from 50% upwards the curve is made up of line segments. These segments correspond to gradients representing files with equal scores (in this case the word *lean* occurs the same number of times in each).

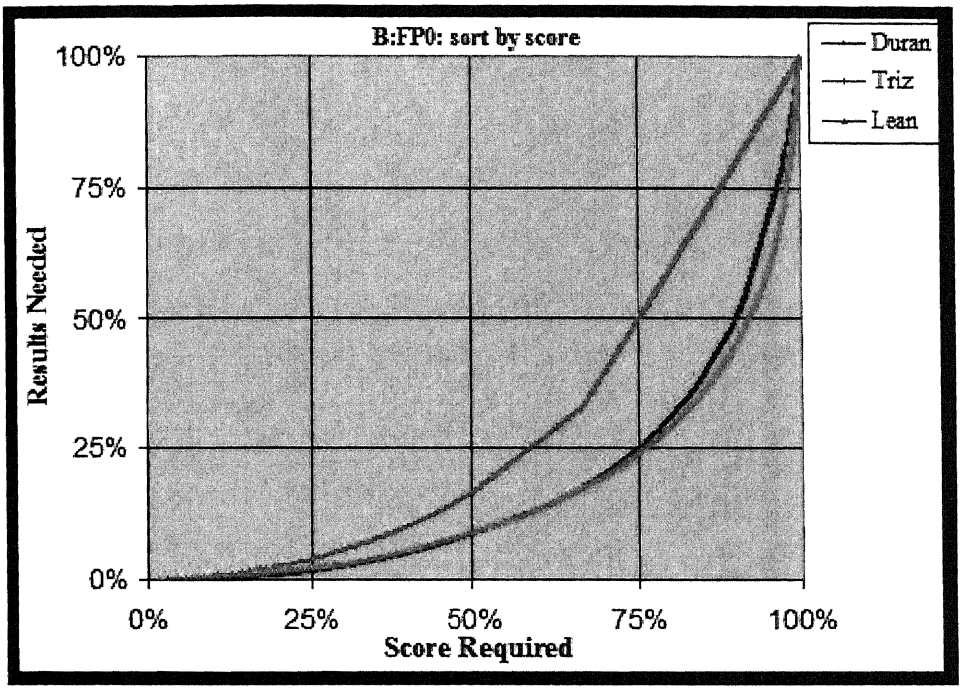


Figure 7.12 FP_0: Results needed vs score required of *Duran*, *TRIZ* and *lean*. Graph showing number of results needed to obtain a given percentage of the target words in the case of *Duran*, *TRIZ* and *lean*. This shows that words occurring more frequently require relatively less results for the same percentage score ("B").

The artefacts just noted may well be expected to be more noticeable for words which occur less often in the sample web. Figure 7.13 shows plots of curves for words, which occur with frequency 10, 100 and 1000 in the sample web, superimposed on Figure 7.12. Because these curves represent the averages over a number of different words, each with their own individual behaviour the artefacts are averaged out.

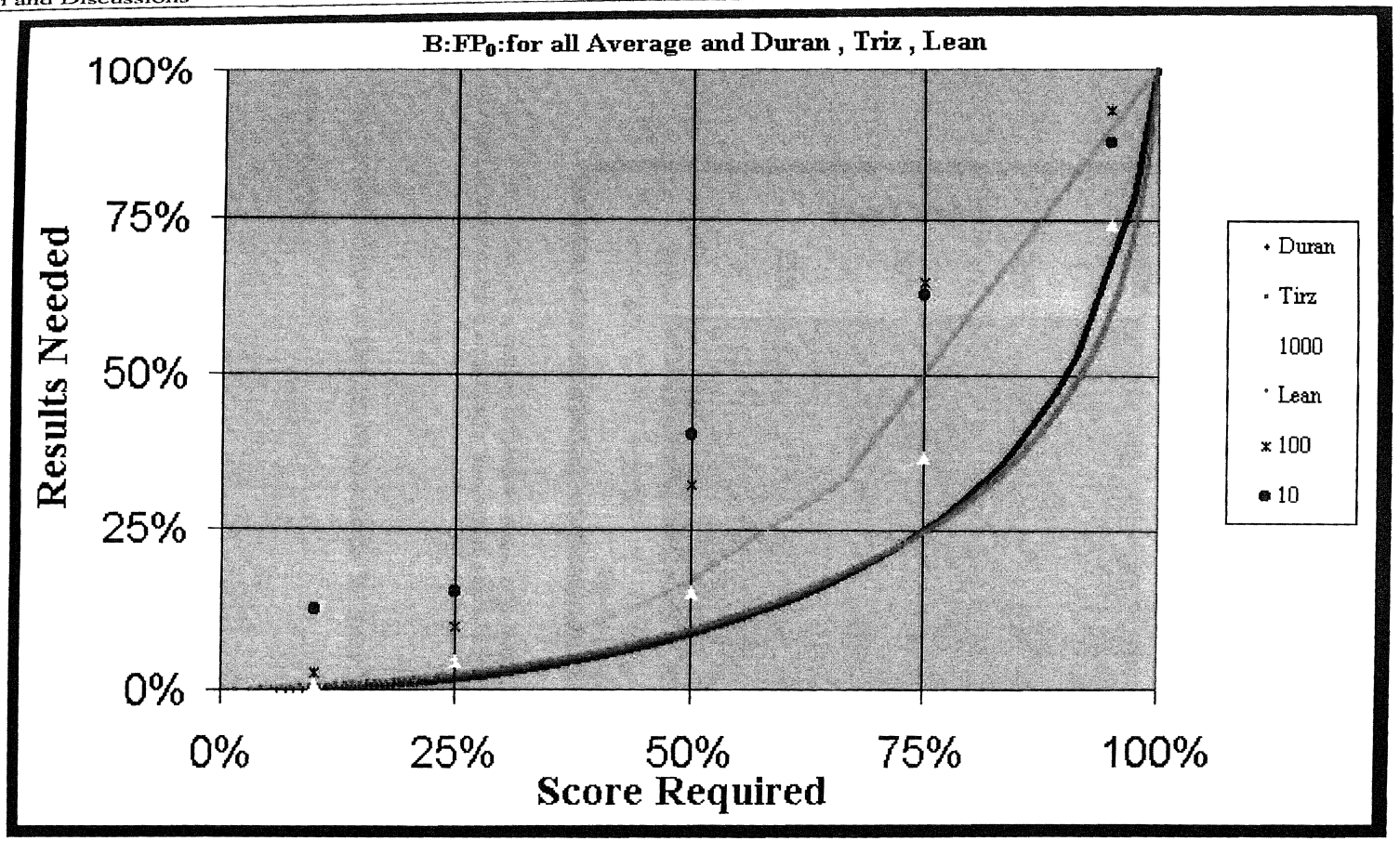


Figure 7.13 FP_0: Results needed vs score required for all search terms. Graph showing results needed to obtain a given percentage of the total score using ISA FP_0. Results for all 6 cases in order of frequency (*Duran*, *TRIZ* and *lean* and frequencies of 10, 100 and 1000) are combined and shows that relatively less results are needed for words occurring more frequently in the *Sample Web* (B).

7.4.4.2 FP_{+1}

The results for FP_{+1} can be analysed in a similar way. Such an analysis is presented plotted for *Duran* in Figure 7.14 approximately 92% of the score is contained in the first 25% of the results. This plot provides a way to visually that the score is distributed throughout the results.

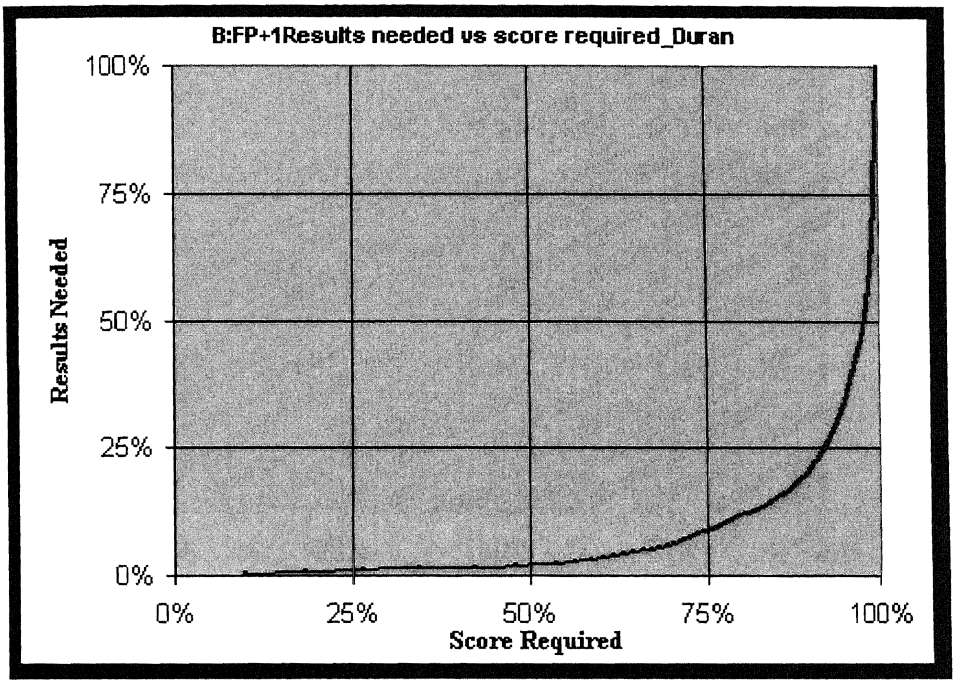


Figure 7.14 *Duran*: FP_{+1} results needed vs score required. Graph showing how many result files must be taken in order to have a particular percentage score in the case of *Duran* using FP_{+1} ("B").

The general shape of the curve is similar to the FP_0 case though the curve seems more pronounced.

The effects of the discrete nature of the scoring function are just visible in this figure. Plots for *TRIZ* and *lean* are added to that of *Duran* in Figure 7.15. The figure shows that the same general behaviour are present for these two words.

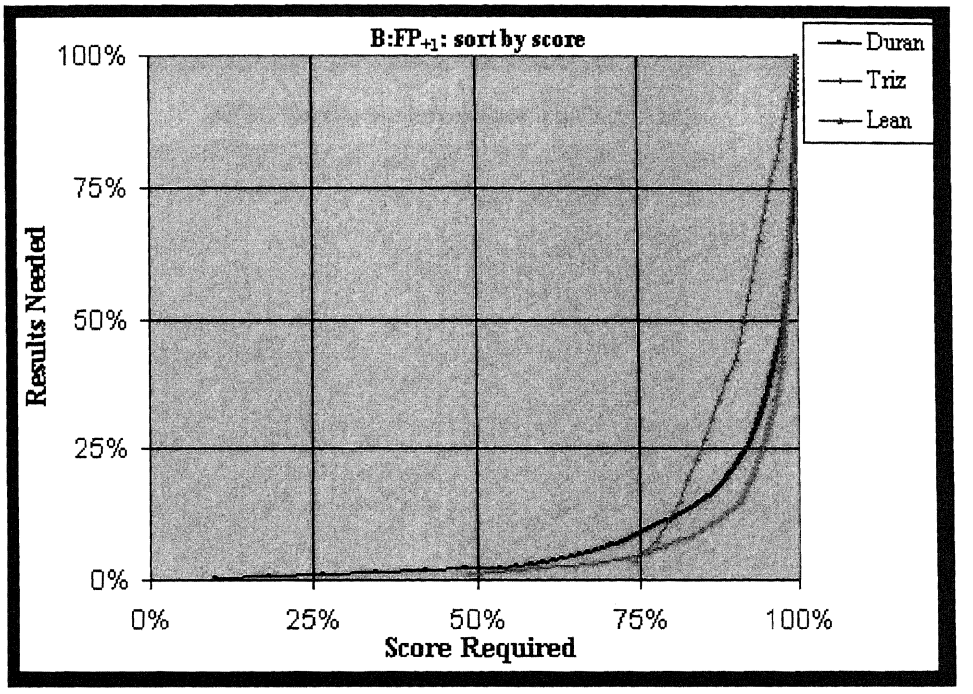


Figure 7.15 FP_{+1}: Results needed vs score required of *Duran*, *TRIZ* and *lean*. Graph showing how many result files must be taken in order to have a particular percentage score in the case of *Duran*, *TRIZ* and *lean* using FP_{+1} ("B").

7.4.4.3 FP_{-1}

FP_{-1} also behaves in a similar way to the other two Fingerprint types as shown in Figure 7.16 where FP_{-1} for *Duran* is plotted.

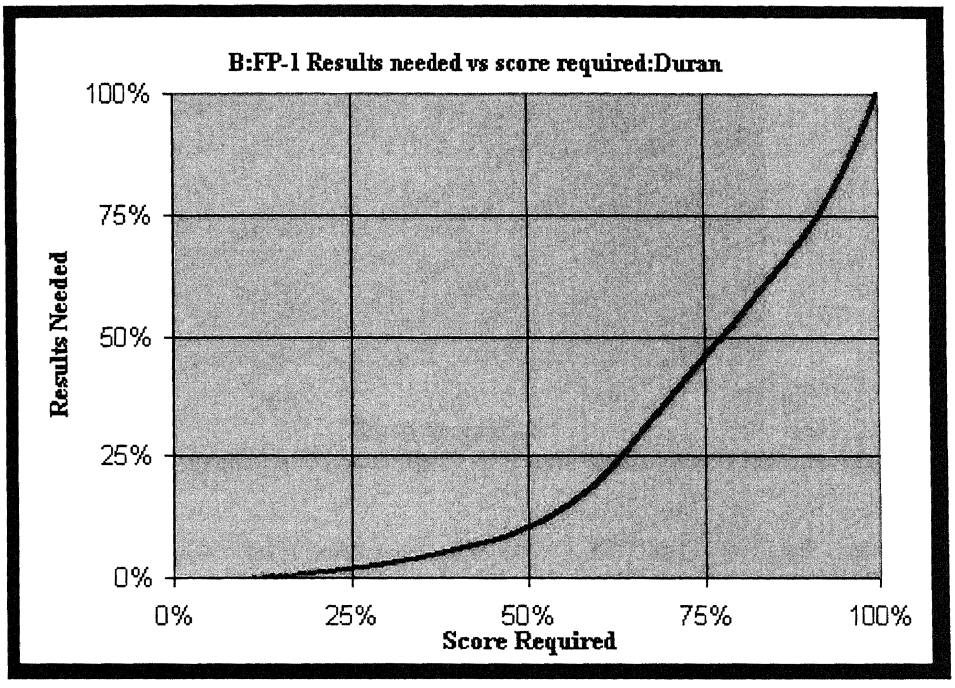


Figure 7.16 *Duran*: FP_{-1} results needed vs score required. Graph showing how many result files must be taken in order to have a particular percentage score in the case of *Duran* using FP_{-1} ("B").

At first sight this curve may look quite strange but its shape is due to the fact that score is an integer and the particular values that the score has in the *Sample Web*.

The curves are thus a series of major line segments representing particular frequencies of occurrence of the words. The lengths of these lines depend on the number of files with that particular frequency of occurrence.

A comparison of the FP_{-1} plots for *Duran*, *TRIZ* and *lean* is shown in Figure 7.17. The effects of discrete nature of the measure can easily be seen.

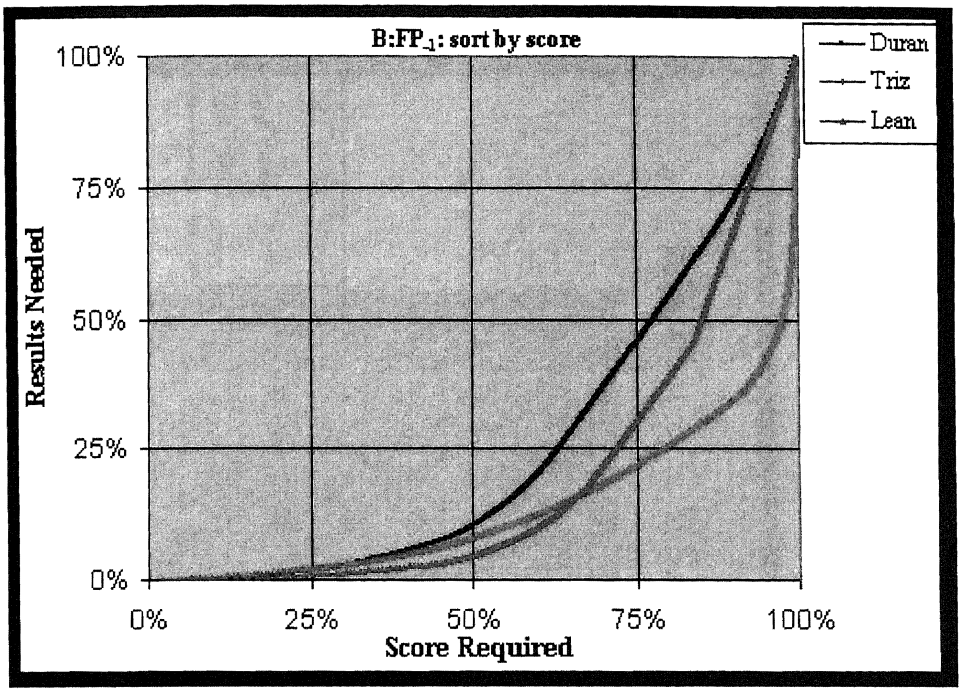


Figure 7.17 FP_{-1}: Results needed vs score required of *Duran*, *TRIZ* and *lean*. Graph showing how many result files must be taken in order to have a particular percentage score in the case of *Duran*, *TRIZ* and *lean* using FP_{-1} ("B").

In Figure 7.18 all three Fingerprints for *Duran* are plotted on the same graph. The three curves show similar behaviour. In the case of *Duran*, FP_{+1} requires fewer results for the same proportion of score than does FP_0 , which in turn requires less than FP_{-1}.

When corresponding results are plotted for *TRIZ* and *lean* (Figure 3 and Figure 4 Appendix B), a different behaviour is apparent. For these FP_{+1} requires less results for the same proportion of score than does FP_{-1} which in turn requires less than FP_0.

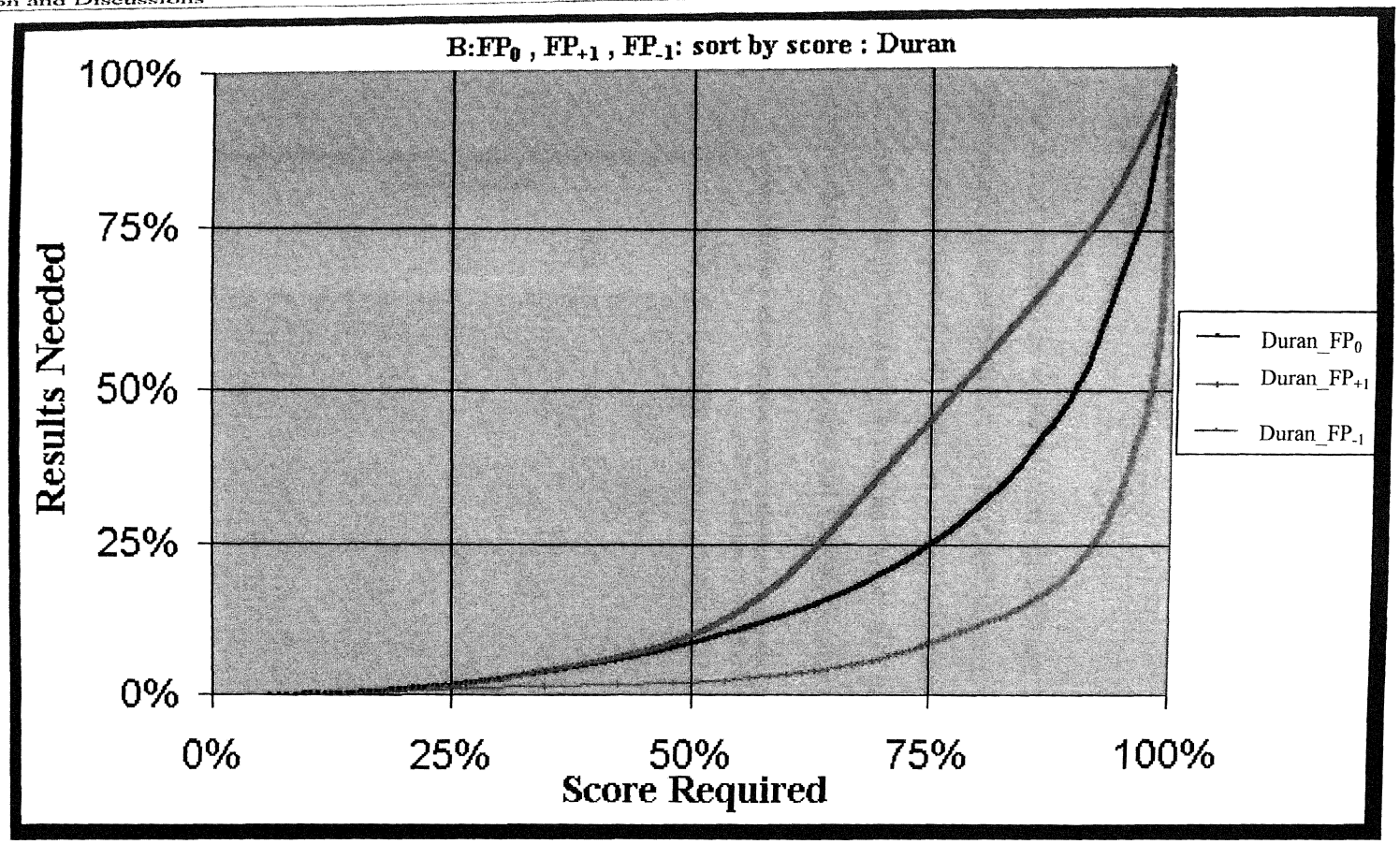


Figure 7.18 *Duran*: FP_0 , FP_{+1} , FP_{-1} for Results needed vs score required. Combined graph of ISA FP_0, FP_{-1} and FP_{+1} searches, showing many result files must be taken in order to have a particular percentage score in the case of *Duran* ("B").

7.4.5 Return on Effort: How Score Accumulates through the Search

In section 7.3 *(graphs category A)* how score accumulates as the search progresses was considered and in section 7.4 *(graphs category B)* how the score was distributed throughout the score-ordered results was the focus of attention. In this section *(graphs category C)* the distribution of the score within the results is considered against the effort expended in obtaining those results.

7.4.5.1 FP_0

The effort required is again plotted (in this case the number of html fingerprint files processed) against the score accumulated in the score-ordered list of results plotted for *Duran* (see Figure 7.19). Graphs of this type show how the search term is distributed amongst files in the sample web and when they are found during an ISA search. The data plotted correspond to the full version of table 7.3.

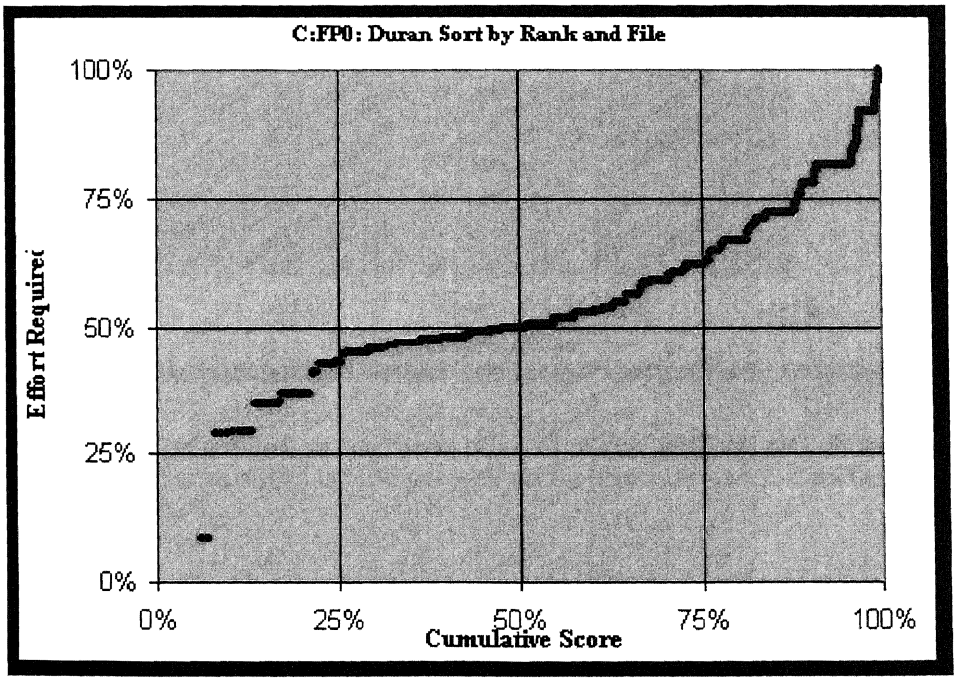


Figure 7.19 *Duran:* FP_0 effort required vs cumulative score. Graph showing how many html files need to be visited in order to accumulate a particular score in the case of *Duran*, using FP_0. This shows three types of behaviour. At first (between 0% and about 20%) there is little 'return on effort'. This is followed by a flat region (between 20% and 70%) where much score results from relatively little extra effort and finally a region where again little return results from increasing effort ("C").

Thus how much effort is required to find the best results containing 50% (say) of the score can be read off the graph. In the case of *Duran*, 50% of the FP_0 html files need to be processed before the best results with 50% of the score are found (see Figure 7.19). As shown in the graph little extra effort is required to go from obtaining 25% of the score to 50% of the score, however, obtaining the best 25% of the score takes a lot of effort. To see if this holds true generally, the corresponding graphs for *TRIZ* and *lean* are plotted in Figure 7.20. The graph shows that there is variability between the results for the words, but all show an initial increased rate of effort required in the early cumulative score (<25%).

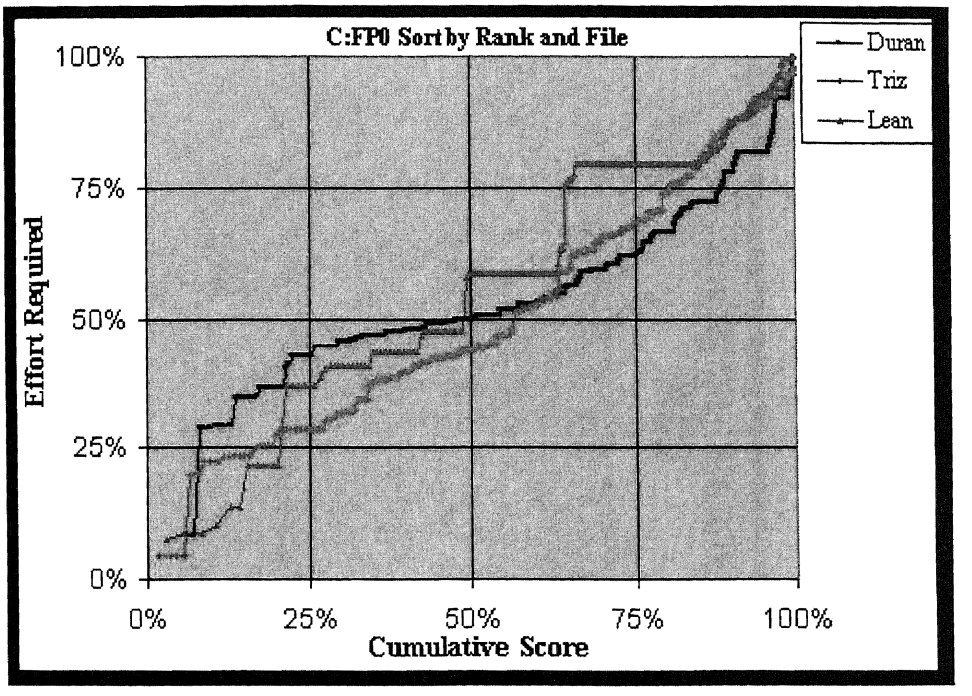


Figure 7.20 FP_0: Effort required vs cumulative score of *Duran*, *TRIZ* and *lean*. Graph showing how many html files need to be visited in order to accumulate a particular score in the case of *Duran*, *TRIZ* and *lean* using FP_0 ("C").

Adding the corresponding curves for the averages of the classes with frequency 10, 100 and 1000 is shown in Figure 7.21. It seems that there may well be a strong dependence of the curves on the frequency of occurrence of the words within the web with higher frequency words requiring relatively less effort (except at the ends of the curve which are of course fixed at 0 and 100% respectively.)

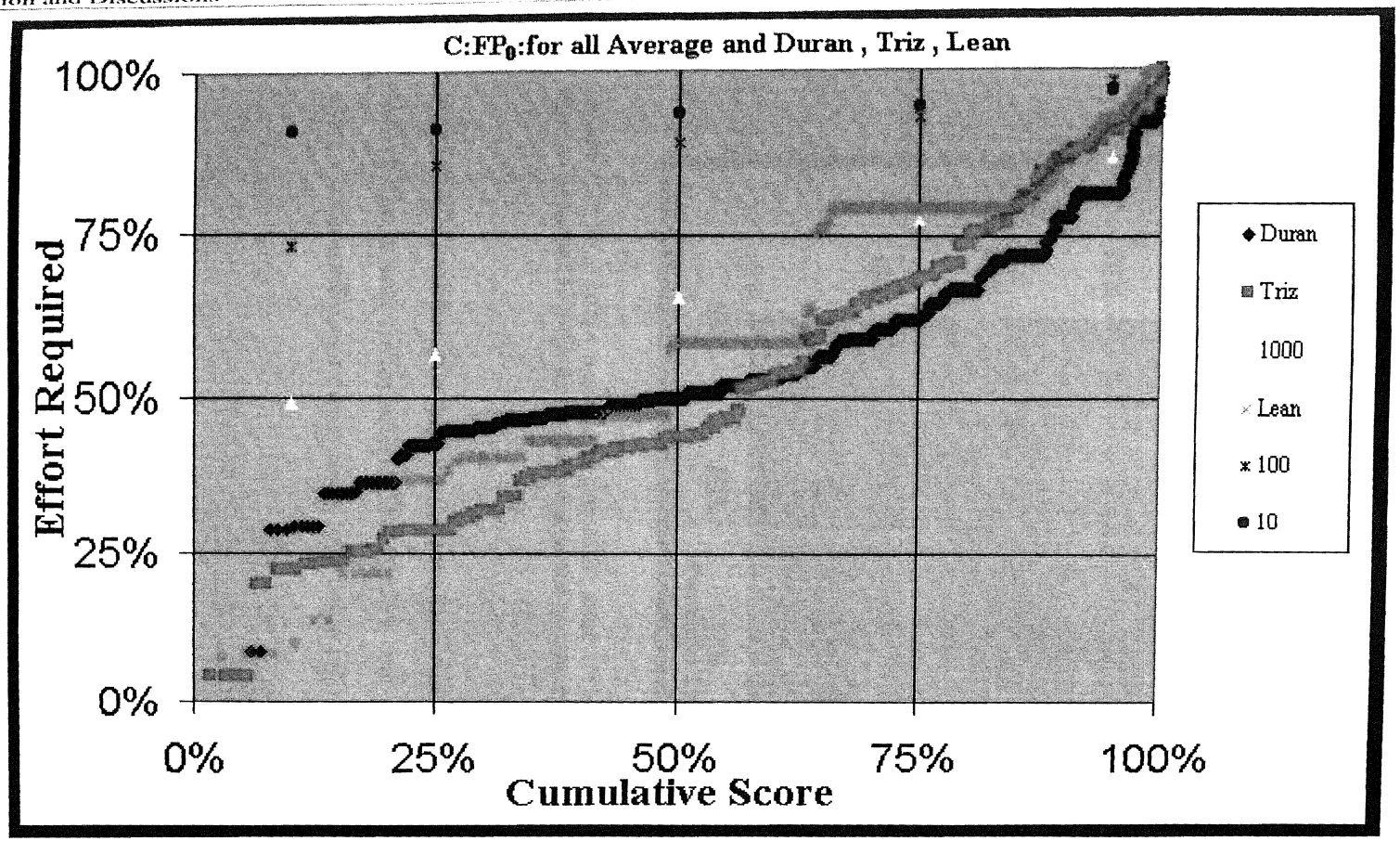


Figure 7.21 FP_0: Effort required vs cumulative score of all search terms. Combined graph showing effort required against cumulative score for the 3 classes (frequencies of 10, 100 and 1000) and 3 search terms (*Duran*, *TRIZ* and *lean*). This shows how generally effort is relatively lower for words, which occur frequently in the *Sample Web* ("C").

7.4.5.2 FP_{+1}

Having looked at the results for FP_0 the focus now moves onto FP_{+1}. Figure 7.22 shows the graph for FP_{+1} for *Duran*. It shows that in comparison with FP_0 this needs relatively much less effort for scores less than 100%.

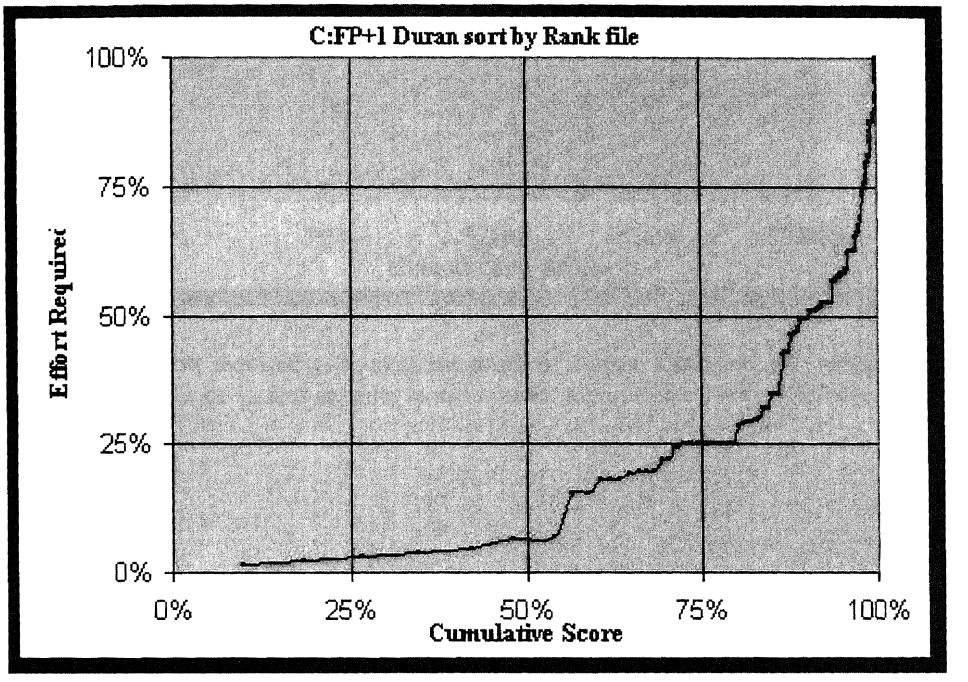


Figure 7.22 FP_{+1}: Effort required vs cumulative score of *Duran*. Graph showing how many html files need to be visited in order to accumulate a particular score in the case of *Duran*, using FP_{+1} ("C").

Figure 7.23, where plotting the curves for *TRIZ* and *lean* alongside that of *Duran* shows the same relationship. This shows that FP_{+1} manages to find clusters of concentration of score more easily than using FP_0.

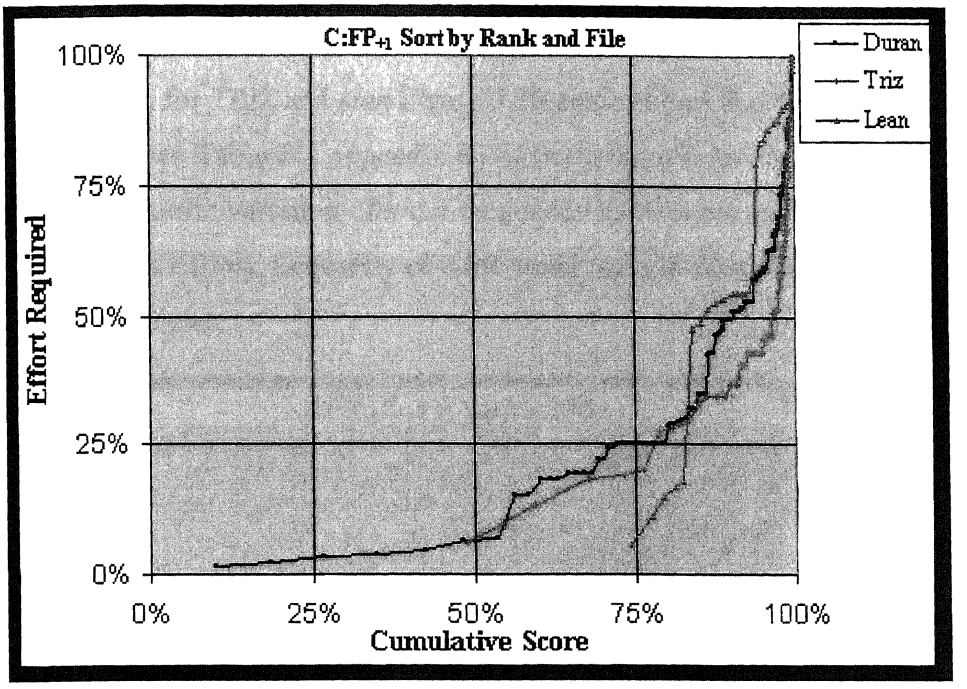


Figure 7.23 FP_{+1}: Effort required vs cumulative score of *Duran*, *TRIZ* and *lean*. Graph showing how many html files need to be visited in order to accumulate a particular score in the case of *Duran*, *TRIZ* and *lean* using FP_{+1} ("C").

7.4.5.3 FP_{-1}

Finally moving on to FP_{-1} Figure 7.24 shows the corresponding graph. The plot of FP_{-1} lies between that of FP_0 and FP_{+1}.

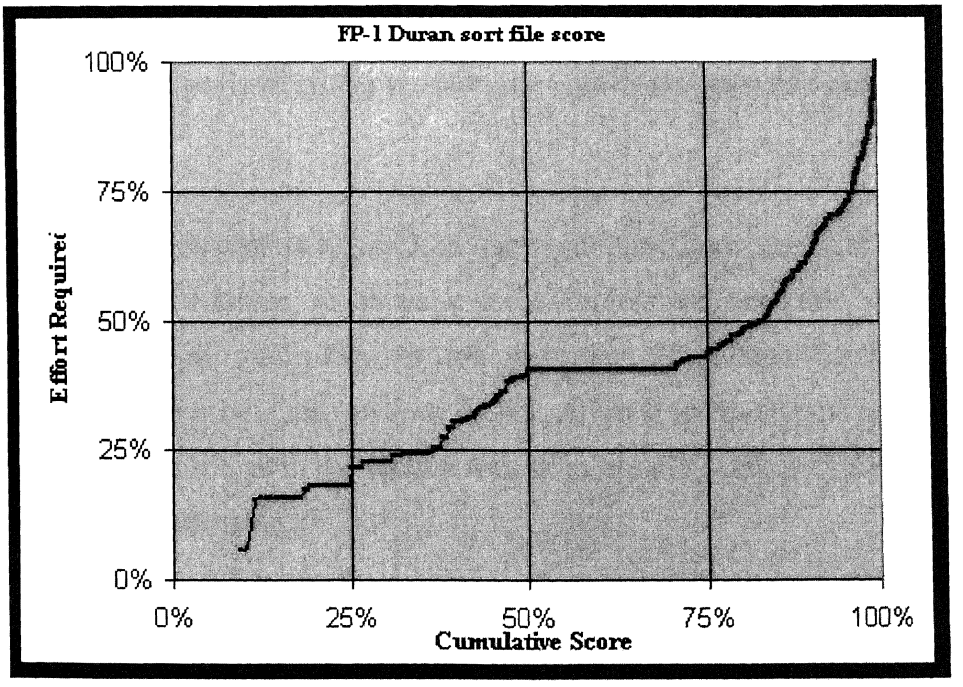


Figure 7.24 *Duran*: FP_{-1} effort required vs cumulative score. Graph showing how many html files need to be visited in order to accumulate a particular score in the case of *Duran*, using FP_{-1} ("C").

Plotting the curves for *TRIZ* and *lean* Figure 7.25 again shows that *lean*, because of its lower frequency (see Table B.1 appendix B) of occurrence behaves 'differently'. This difference is due to the variation of word frequency as files are encountered showing up more for words of low frequency of occurrence than it does when treating words with higher frequencies.

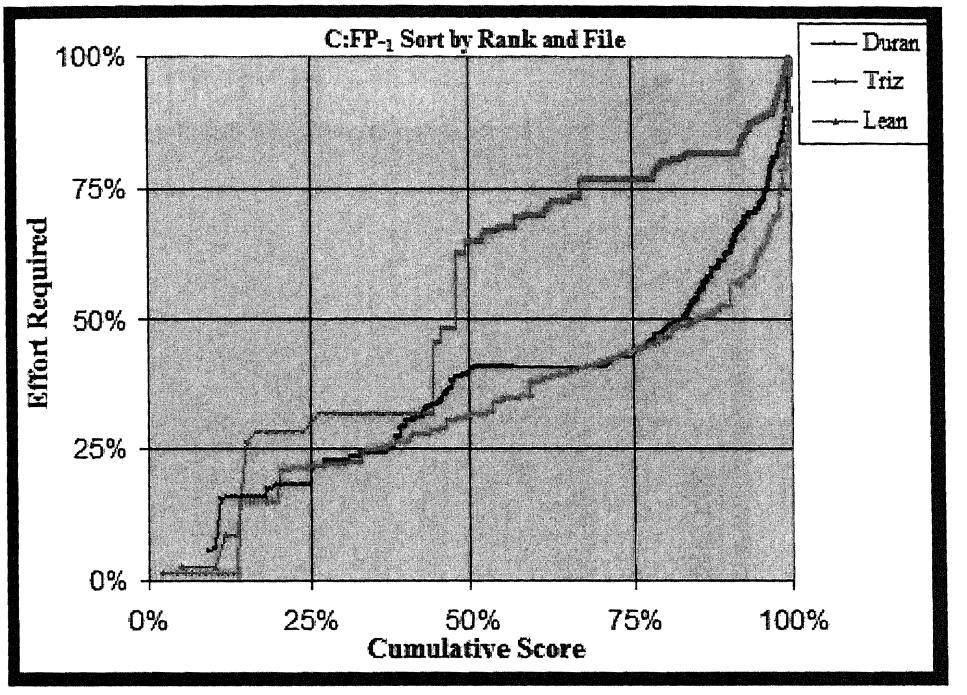


Figure 7.25 FP_{-1}: Effort required vs cumulative score of *Duran*, *TRIZ* and *lean*. Graph showing how many html files need to be visited in order to accumulate a particular score in the case of *Duran*, *TRIZ* and *lean* using FP_{-1} ("C").

The final graph in this section Figure 7.26 shows all three fingerprints for *Duran*. The graph shows that for Duran, to obtain a given cumulative score FP_{+1} requires less effort than FP_{-1} which in turn requires less effort than FP_0. Graphs for *TRIZ* show a very similar behaviour but those for *lean* do not. Again it seems that the relatively low frequency of the word lean in the sample web does not allow curves for FP_{-1} and FP_0 to be clearly distinguishable.

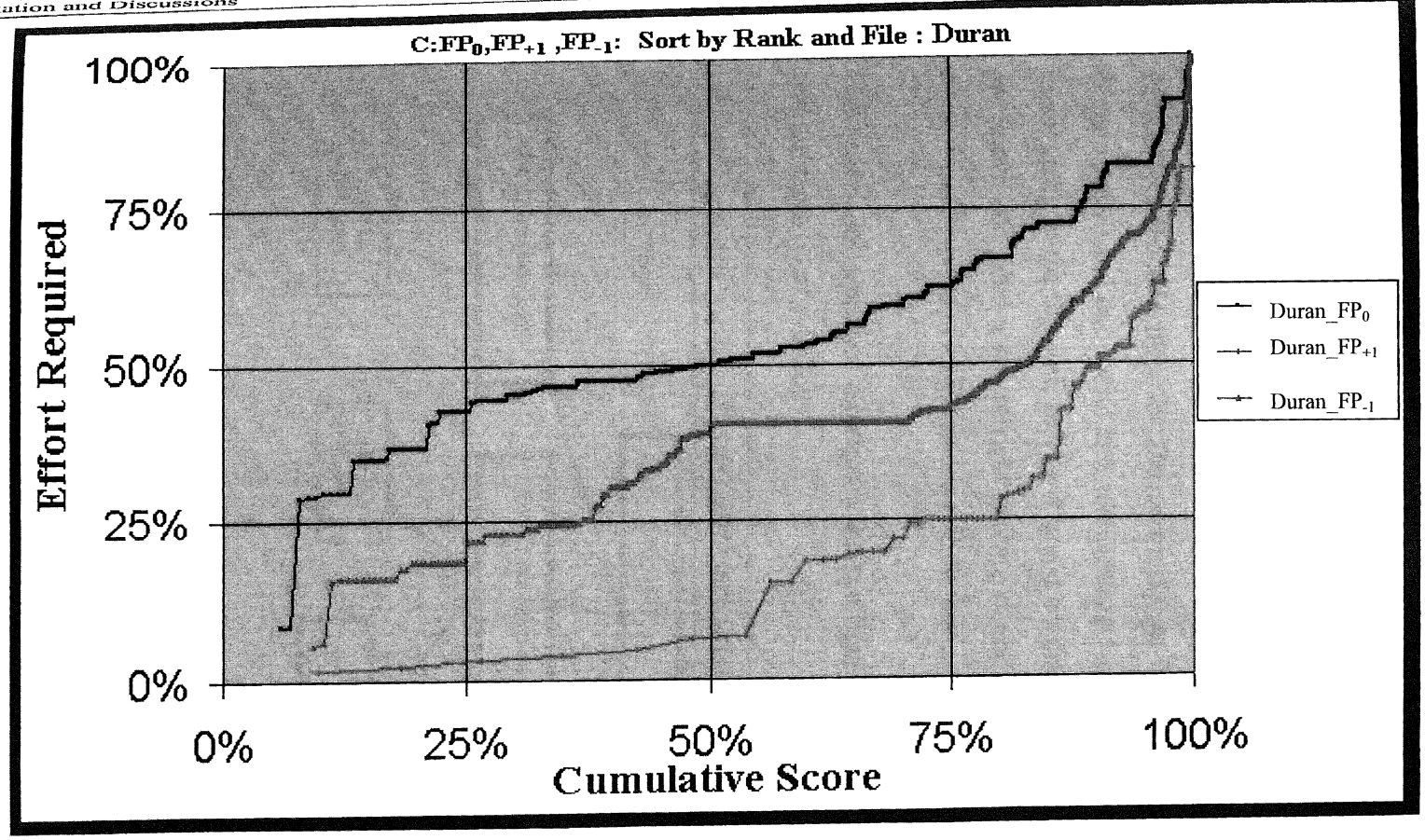


Figure 7.26 FP_0 , FP_{+1} , FP_{-1} effort required vs cumulative score of *Duran*. Combined graph of ISA FP_0, FP_{-1} and FP_{+1} searches, showing how many html files need to be visited in order to accumulate a particular score in the case of *Duran* ("C").

7.5 Evaluation of the Search Engines

Having described how the progress of an individual search may be evaluated, attention is now turned to looking at how individual search engines perform.

7.5.1 Search Tree Fragments

One traditional way of visualising the search space is by means of a search tree. Parts of the trees for searching for *Duran* are drawn in Figures 7.27 and 7.28, one for a Google search and one for an ISA search respectively. For clarity, only that part of the tree leading to the top four results returned by the search engines is drawn. The trees are drawn such that a left to right reading of the leaves gives the top four results in order. In other words Google ranking of the leave is left to right Figures 7.27.

Non-leaf nodes represent directories and the figures in brackets represent the rank (***r***) and the score (***s***) of that node.

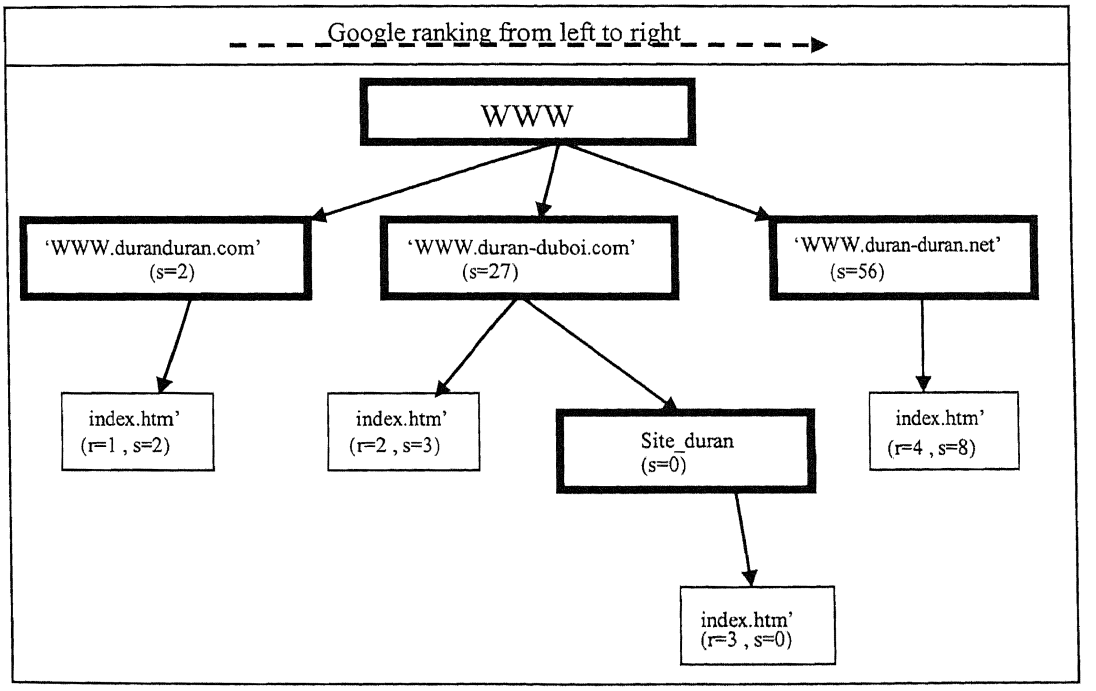


Figure 7.27 Results show in a tree for Duran from Google. Extract of search results tree for *Duran* using a Google search showing the first four results as leaves of the tree. Numbers in brackets (r, s) show rank ***r***, and ISA score, ***s*** (Bold boxes is folder).

Figure 7.27 shows part of the result tree for *Duran* using Google and shows Google's top four results. It is easy to see that the search engine is returning parts of the tree that apparently have few instances of the required term. In the case of the Google search it is clear that despite the results returned having low scores in terms of the number of target words in them (the total score for the four pages is only 13) they do belong to relevant domains.

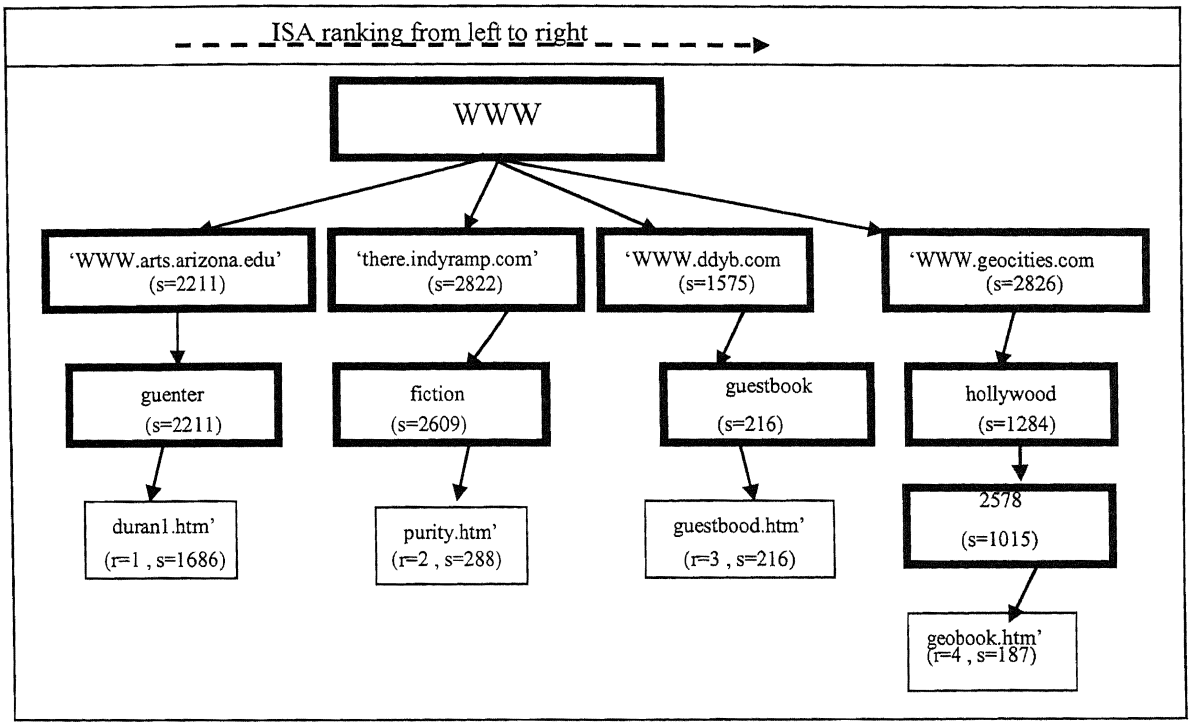


Figure 7.28 Section of result tree: Duran from ISA. Extract of search tree for *Duran* using ISA search showing the first four results as leaves of the tree. Numbers in brackets (r, s) show rank ***r***, and ISA score, ***s*** (Bold boxes is folder).

On further investigation the first site, www.duranduran.com proves to be very relevant to those seeking information about the pop group, it is their official site. A natural question to ask is 'Why does ISA give this site such a low score of 2?' It seems that the site, like many others, uses Flash and Active Server pages. The former is unreadable by ISA and the latter, as virtual pages are ignored by ISA. This shows the effect of these two factors on the search results.

Yahoo has this site as its number one site. Not all search engines recognise the importance of this site. MSN search, for example, has this 'official site' at 10th on the list for search term *Duran* whereas Lycos has this as first.

For the ISA search the relevance of the domains is much less clear despite the total score of these four pages amounting to 2377 (the sum of the scores for the four leaves). Of these 1686 turned out to be spam whilst the second page ("http://there.indyramp.com/fiction/purity.htm") is in fact part of a Duran Duran 'web ring' and as such might well be an excellent source neglected by the other search engines. It is part of the 'grey web', which though often poorly indexed often contains a great deal of useful information placed on the web by keen amateurs.

7.5.2 Score and Relevance

Measures of *Relevance* have been defined in section 5.2.2 and here the community relevance of a document, which is calculated from its positions in the results of the search engines are compared with its score.

Tables 7.5 and 7.6 show part of a summary of the 195 different results (filtered as described earlier in section 5.3) found by the different search engines for *Duran* when 50 results were requested from each.

The second column of Table 7.5 shows the community relevance of each page whilst column three shows its score. It is clear from this table that there is no obvious connection between the score and the relevance as measured by the community relevance measure. Clearly the search engines are not using the frequency of occurrence of the search term directly to rank pages. In fact many (69%) do not contain the search term at all.

No	r_p	Score (σ)	URL
1	3	2	http://www.duranduran.com/
2	1.04	0	http://www.angelfire.com/ma2/duranpictures/index.html
3	1	1686	http://www.arts.arizona.edu/guertner/duran1.htm
4	0.86	0	http://www.ddyb.com/
5	0.83	3	http://www.duran-duboi.com/
6	0.63	2	http://www.lizardkingduran.com/
7	0.56	40	http://www.hollywoodandvine.com/duranduran/
8	0.46	0	http://www.duranie.com/
9	0.41	0	http://www.duran-duran.net/
10	0.4	0	http://www.duranduran.co.uk/
11	0.37	22	http://duranduran.20m.com/index.htm
12	0.36	0	http://home6.swipnet.se/~w-69190/duranduran.html
13	0.33	0	http://www.duran-duboi.com/site_duran/
14	0.33	288	http://there.indyramp.com/fiction/purity.htm
...	...	...	...
...	...	...	...
190	0.02	0	http://www.gmms.mcmail.com/
191	0.02	347	http://www.trusttheprocess.com
192	0.02	0	http://www.rock-infodatenbank.de/duran_duran_5393808.htm
193	0.02	0	http://www.gmms.mcmail.com/wd.htm
194	0.02	0	http://www.eonline.com/Facts/People/0,12,4779,00.html
195	0.02	53	http://dir.clubs.yahoo.com/music/genres/rock_and_pop/artists/complete_category_listing/duran_duran/index.htm

Table 7.5 Search results for *Duran* using all 5 search engines. Extract of combined search results for *Duran* using all 5 search engines in order of community relevancy, showing how ISA score varies with community relevancy.

An illustration of this is shown in Figure 7.29 below. It shows the ISA score of the 50 results returned by Google as a scattergram. The results are arranged in Google's rank order and the graph shows that the web page scores are not correlated with the position that Google puts them in.

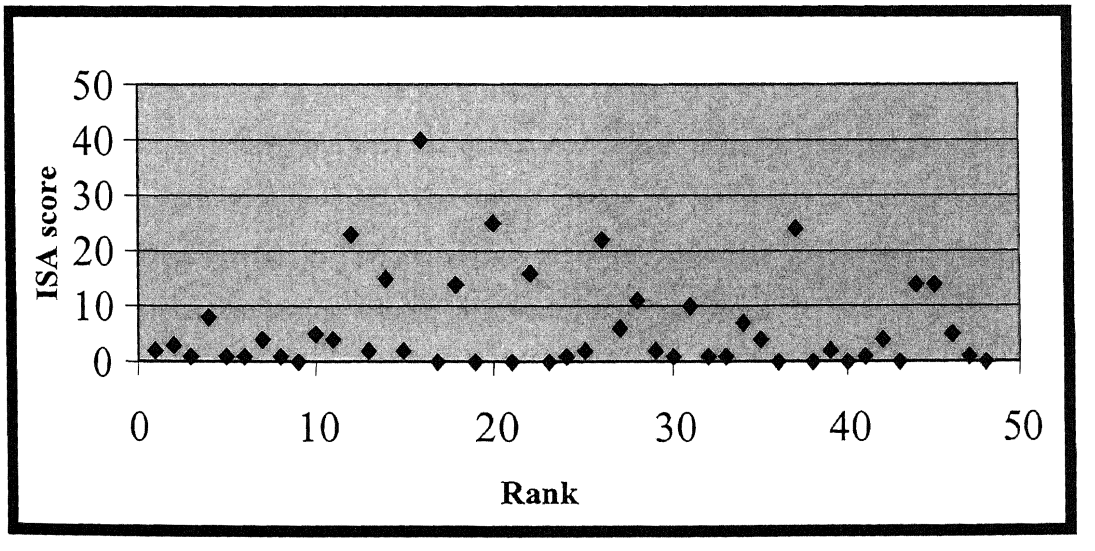


Figure 7.29 Scattergram for the Google search for *Duran*. Scattergram showing lack of correlation between ISA score and Rank.

7.5.3 Strength, Vigour and Union Scores

These terms were defined in section 5.2.4 as measures of how well a search engine performs. Here we compare these measures for the 5 search engines.

Table 7.6 shows the calculations of *strength*, *vigour* and *union scores* for each of the search engines. This table can be read in conjunction with table 7.5, which shows the pages in the same (rank) order and gives their rank, URLs, community relevance and number of target words contained in the file. The diagrams have been split for ease of use and presentation.

For *Duran* the search engines (including ISA's best 50) found one hundred and ninety five different files. The table shows the rank of each in the results for Google (g), Alta Vista (a), Yahoo (y), Webcrawler (w) and ISA (i). When the file was not in the results for a particular search engine a zero is placed as the rank (this occurred 248 times). Thus line one shows that Google (g), Alta Vista (a) and Yahoo (y) each agreed on a particular web page as the most relevant. This is the official Duran Duran web site. However, Webcrawler (w) and ISA (i) did not rank this page at all (amongst the top 50). The results returned first by Google (g), Alta Vista (a) and Yahoo (y) do not have the search term in them as text but it does occur in the HTML.

The occurrence of so many zeros may seem quite surprising but reflects the fact that the search engines do not in general give the same page. Not so many zeros in the ISA column might be expected but they do occur and this is because the files returned by other search engines do not contain the target phrase, or contain it in the meta data or HTML code. This results in ISA receiving zero strength and vigour scores despite finding many pages containing a large number of target phrases.

The next five columns show 'reciprocal rank' which is used as a measure of the relevance that the search engine gives the page, and is used in later calculations as described in section 5.2.2. A zero in this column should be read as 'not applicable'.

In the next five columns the calculation of strength of the search engines are shown. Google (*g*), Alta Vista (*a*) and Yahoo (*y*) have comparable scores on this measure with Alta Vista coming out slightly ahead of the other two. This is rather surprising since Yahoo is 'powered by' Google and so it might be expected that they agree with each other rather more than with Alta Vista. WebCrawler (*w*) has a much lower score than the others excepting ISA. ISA (*i*) stands out with a strength of zero because it does not return results (in its top 50) that are returned by the other engines, as it finds different ones which might be considered 'better' in that they each contain the search term many times.

Next the *union score* is calculated and it can be seen that the scores here are quite different for the search engines. Obviously ISA has the highest score, as it was able to search the whole of the sample web.

Yahoo comes next with under half of the score of ISA. Google and WebCrawler have comparable scores to each other but these are only about a 20^{th} of the Yahoo score. The file that makes Yahoo do so well has a high score but as it was a dynamic (virtual) page it was not in the sample web. Alta Vista comes in a very poor last reporting only 29 occurrences of the search term against the 5653 contained in the top 50 results as measured by occurrences.

No	Rank					Reciprocal of Rank					Calculation of Strength					Calculation of Union Score					Calculation of Vigour				
	g	a	y	w	i	r_{pg}	r_{pa}	r_{py}	r_{pw}	r_{pi}	s_g	s_a	s_y	s_w	s_i	u_g	u_a	u_y	u_w	u_i	v_g	v_a	v_y	v_w	v_i
1	1	1	1	0	0	1.00	1.00	1.00	0.00	0.00	2.00	2.00	2.00	0.00	0.00	2	2	2	0	0	2	2	2	0	0
2	0	23	0	1	0	0.00	0.04	0.00	1.00	0.00	2.00	2.04	2.00	0.04	0.00	2	2	2	0	0	2	2	2	0	0
3	0	0	0	0	1	0.00	0.00	0.00	0.00	1.00	2.00	2.04	2.00	0.04	0.00	2	2	2	0	1686	2	2	2	0	1686
4	6	2	0	5	0	0.16	0.50	0.00	0.20	0.00	2.12	2.23	2.00	0.18	0.00	2	2	2	0	1686	2	2	2	0	1686
5	2	3	0	0	0	0.50	0.33	0.00	0.00	0.00	2.28	2.39	2.00	0.18	0.00	5	5	2	0	1686	3.5	3	2	0	1686
6	29	26	2	17	0	0.03	0.03	0.50	0.05	0.00	2.30	2.42	2.07	0.21	0.00	7	7	4	2	1686	3.57	3.08	3	0.12	1686
7	16	0	0	2	0	0.06	0.00	0.00	0.50	0.00	2.34	2.42	2.07	0.24	0.00	47	7	4	42	1686	6.07	3.08	3	20.12	1686
8	23	4	0	6	0	0.04	0.25	0.00	0.16	0.00	2.35	2.47	2.07	0.29	0.00	47	7	4	42	1686	6.07	3.08	3	20.12	1686
9	4	6	0	0	0	0.25	0.16	0.00	0.00	0.00	2.39	2.51	2.07	0.29	0.00	47	7	4	42	1686	6.07	3.08	3	20.12	1686
10	0	15	3	0	0	0.00	0.06	0.33	0.00	0.00	2.39	2.53	2.09	0.29	0.00	47	7	4	42	1686	6.07	3.08	3	20.12	1686
11	26	0	0	3	0	0.03	0.00	0.00	0.33	0.00	2.41	2.53	2.09	0.30	0.00	69	7	4	64	1686	6.92	3.08	3	27.45	1686
12	0	9	0	4	0	0.00	0.11	0.00	0.25	0.00	2.41	2.56	2.09	0.33	0.00	69	7	4	64	1686	6.92	3.08	3	27.45	1686
13	3	0	0	0	0	0.33	0.00	0.00	0.00	0.00	2.41	2.56	2.09	0.33	0.00	69	7	4	64	1686	6.92	3.08	3	27.45	1686
14	0	0	0	0	3	0.00	0.00	0.00	0.00	0.33	2.41	2.56	2.09	0.33	0.00	69	7	4	64	1974	6.92	3.08	3	27.45	1782
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
190	0	49	0	0	0	0.00	0.02	0.00	0.00	0.00	2.51	2.68	2.17	0.44	0.00	99	29	1597	80	5600	9.48	3.95	230.11	28.49	2092
191	0	0	49	0	0	0.00	0.00	0.02	0.00	0.00	2.51	2.68	2.17	0.44	0.00	99	29	1944	80	5600	9.48	3.95	237.19	28.49	2092
192	0	0	0	49	0	0.00	0.00	0.00	0.02	0.00	2.51	2.68	2.17	0.44	0.00	99	29	1944	80	5600	9.48	3.95	237.19	28.49	2092
193	0	50	0	0	0	0.00	0.02	0.00	0.00	0.00	2.51	2.68	2.17	0.44	0.00	99	29	1944	80	5600	9.48	3.95	237.19	28.49	2092
194	0	0	0	50	0	0.00	0.00	0.00	0.02	0.00	2.51	2.68	2.17	0.44	0.00	99	29	1944	80	5600	9.48	3.95	237.19	28.49	2092
195	0	0	0	0	50	0.00	0.00	0.00	0.00	0.02	2.51	2.68	2.17	0.44	0.00	99	29	1944	80	5653	9.48	3.95	237.19	28.49	2093

Table 7.6 Search engine ranks, reciprocal rank, *Strength, Union Score* and *Vigour*. Continuation of table 7.5 showing search engine ranks, reciprocal rank, strength, union score and vigour for the combined results in *community relevance* order.

The last five columns of the table show the *vigour* of the engines. A surprising feature here is the relative vigour of WebCrawler to Google. They had comparable union scores indicating that they returned roughly the same number of target phrases. However it seems that WebCrawler returns these much higher in its list of results.

A surprising lack of consensus amongst the search engines tested is indicated by these results. It might have been expected that there would be more commonality of results in the links returned in response to a search than was actually found.

The number of results that the search engines have in common is shown in Table 7.7. The four commercial engines have some pages in common but these represent less than half of their results. ISA does not share any results with these. ISA relies solely on the number of target words as a score and does not use humans to evaluate the pages, whereas for the other search engines the number of occurrences of the target word seems to play a small part (if any) in the position of the page on the list of results.

	Alta Vista	**Google**	**Webcrawler**	**Yahoo**	**ISA**
Alta Vista	50	17	15	10	0
Google	17	48	11	7	0
Webcrawler	15	11	50	6	0
Yahoo	10	7	6	49	0
ISA	0	0	0	0	50

Table 7.7 Number the results have in common for all search engines. Table showing number of pages that the results have in common for the top 50 results of the 5 search engines.

7.5.4 What Search Engines Find and What They Miss

The scores of the first pages returned by each search engine are shown in Table 7.8. Given that the search is for pages about '*Duran*' it does seem surprising that for each search engine, besides ISA, the search term only occurs once or twice in the top reported result. ISA's first result, which contained 1686 occurrences of Duran was spam and so does not appear in this table.

Example best	Duran	ISA Score
ISA	http://there.indyramp.com/fiction/purity.htm	288
Google	http://www.duranduran.com/	2
Altvista	http://www.duranduran.com/	2
Yahoo	http://www.lizardkingduran.com/	2
Webcrawler	http://www.angelfire.com/ma2/duranpictures/index.html	1

Table 7.8 Scores of the top results for the 5 search engines showing the low scores returned.

In fact, except possibly for Webcrawler, the results returned are good *Duran* sites according to an informal inspection.

The scores for a number of web pages not returned but that, on the basis of the number of occurrences of the search terms, seem to warrant inclusion in the results of the search are given in Table 7.9. Each belongs to a domain that is known to the search engines in that the domain is represented in the combined search results. However the search engines did not return these pages. There are a number of possible reasons for this omission. For example the pages might have been too deep for the indexing process to find or the human who evaluated the page did not think it sufficiently important to warrant inclusion.

URL	Score
http://there.indyramp.com/fiction/purity.htm	288
http://www.fortunecity.com/tinpan/klf/67/story.htm	233
http://www.ddyb.com/guestbook/guestbook.htm	216
http://www.geocities.com/hollywood/2578/geobook.htm	187
http://www.arts.arizona.edu/guertner/durancollection.htm	161
http://www.links2go.com/topic/duran_duran/index.htm	134
http://duranduran.20m.com/fanzine/issue1/ch2.htm	129
http://www.imissthe80s.com/duranduran.htm	122
http://www.geocities.com/hollywood/2578/dpressm.htm	113
http://duranduran.20m.com/duran/articles/index.htm	113
http://www.geocities.com/hollywood/2578/amazon.htm	94
http://www.fortunecity.com/tinpan/morrison/200/index.htm	89
http://www.geocities.com/hollywood/2578/bio.htm	89
http://members.aol.com/duranshop/page2/RockMags.htm	88
http://www.ddyb.com/thankdd/guestlog.htm	87
http://dir.clubs.yahoo.com/music/genres/rock_and_pop/artists/complete_category_listing/duran_duran/~white_pages/0.htm	85

Table 7.9 High scoring results that were missed by the search engines. Table of high scoring web pages that were missed by the traditional search engines (spam pages have been removed).

The final table in this section shows the total number of target words found for each search term by each search engine. In all cases except ISA the figures are for the top 50 results returned.

Columns labelled 'Raw' are the unadjusted scores whilst those labelled Net have been adjusted to exclude pages that are considered 'spam'. It is clear from a comparison of this table 7.10 with table 7.9 that the total scores for each search engine's 'top 50' results are less than the combined score of ISA's top two

Query	Alta Vista		Google		Webcrawler		Yahoo		ISA	
	Raw	Net	Raw	Net	Raw	Net	Raw	Net	Raw	Net
Duran	37	29	84	77	63	58	62	44	28306	26620
Liquid Marbles	62	57	96	88	36	28	69	56	24753	24381
Triz	32	24	92	86	56	47	24	22	21456	20123
Plagiarism	53	47	86	81	25	21	56	42	34263	31049
Total	184	157	358	332	180	154	211	164	108778	102173

Table 7.10 Total target words in the top 50 search results for the 5 search engines and a selection of 5 search terms.

It is also clear that the search engines place much more emphasis on other criteria at the expense of score. The search engine results are, however, still valid as can be judged by looking at the web sites themselves as they all are relevant to the search.

7.6 Discussion

In this work word frequency in the file (FP_0), word frequency of files pointed to by the file (FP_1) and word frequency of files pointing to the file (FP_{-1}) are studied as a means of identifying the contents of the files. These are collectively referred to as Fingerprints.

This research investigated the following question: (see section 1.2.3)

'Can the use of 'fingerprint' type ideas improve the automatic retrieval of information from large poorly structured databases such as the web?'

Fingerprints give an indication of the content of the web page and that of its link-wise neighbourhood (FP_{+1}, FP_{-1}) in the web. As such they can be used to identify resources that might be of use to the searcher and can thus benefit a traditional search engine by complementing the information that these search engines already have.

The fact that the results returned by ISA, have more occurrences of the search term than those currently returned by conventional search engines, indicates that the use of FP_{+1} and FP_{-1} fingerprint by search engines will give them the ability to provide the searcher with different items than is currently the case. Fingerprints, as defined in this work, are determined automatically and so little additional overhead would be added.

Presented in this section are brief outcomes of three main experiments:

- Consensus amongst search engines
- The use of word counts, links and directory structure as an aid to Internet search
- The nature of search results found by the ISA technique

7.6.1 Consensus Amongst Search Engines

Search engines return links to files that have been judged by the search engine to be of relevance to the search although the basis of this judgement is mostly quite opaque to the users. Complex formulae are used which take into account not only the content of the text appearing when the resource is viewed but other features such as the contents of Meta tags. Experimentally there is no discernable connection between the frequencies of search terms appearing in the resulting pages and their rank in search engine results. Although formulae are published [Chowdhury, 2004] concerning the indexing algorithms used by search engines, commercial interests means that details are very hard to find.

Although broadly similar in effect, the search engines studied did not produce a consistent set of results. It would be difficult to claim that, given a search term, there were a set of 'must have' results that all useful search engines would give. It appears that, at least at the time that the experiments were made and for the searches used, the results returned were just a useful set of results rather than 'the best results'. The answer to the question 'Do current search engines perform the same task?' is therefore yes in that they return reasonable results but no if the question is taken as meaning 'Do they return essentially the same results?'. As there is still no objective way of determining if a set of results is 'best' all that can be done is to determine if a set of results is 'good'. The fingerprints and measures derived from them are one objective way of measuring if the results are 'reasonable'.

7.6.1.1 Lack of Consensus Amongst Search Engines

Excluding ISA, this study looked at four search engines: Alta Vista, Google, WebCrawler and Yahoo. Five searches were initially conducted using each of the four search engines in order to obtain a sample web for later experiments. The search terms given to each engine were: *Duran, Gemba, Liquid marbles, lean* and *TRIZ.*

Comparing the results of these searches showed that, given the same search terms, these search engines return sets of results, which have little commonality although the results that they return on inspection seem to be 'relevant' to the search. Even when the search engines results have common pages they rarely agree on the ranking of the pages. Even the order that the pages appear differs.

From this it can be concluded that, although search engines do in general return useful material the results that one gets are likely to depend upon the search engine used. That users are (in general) able to satisfy their queries implies that there is much on the web that satisfies these queries and users are not too particular about what they accept as results. The results outlined above provide the search engine evaluator with a problem of how to compare search engines.

The strategy adopted here is to use the ranks of the pages as a measure of their relevance. Reciprocal rank is used so that the measure is higher for pages higher in the list of results. Adding these measures gives a measure that represents a consensus or *community measure.*

Having assigned relevance to each page, the search engines themselves can be evaluated by finding the sum of the relevancies of the pages that they returned, weighted by the rank that the engine gave the page.

Thus an engine which gave a high score to a page with high relevance should score highly. While one that gave a high rank to a page that was thought of (by the other search engines) of low relevance should score poorly as would an engine that scored poorly a page that the others scored highly.

One technique adopted here is to have each engine 'mark' each page of its results with the reciprocal of the search engines ranking of that page. This figure is a measure of the relevance that the search engine assigns to that page. This results in a number between 1 and 0 (with zero being given to those pages not in the list of results for that engine). Reciprocal rank is used so those higher scores are given to the pages returned higher in the list. When these reciprocal ranks for a fixed number of results are added together a measure is obtained that represents a 'peer review' of the engines. Search engines that score highly against this measure will be in agreement with the 'community' of engines.

The results obtained in this work gave Alta Vista a higher *strength* than Google, which in turn scored more highly than Yahoo. Webcrawler scored very poorly against this measure, whilst ISA received no score as its top pages were not in the top 50 of the other search engines (see Table 7.11).

Search Engine	Strength
Alta Vista	511
Google	481
Yahoo	417
WebCrawler	80
ISA	0

Table 7.11 *Strength* value for all search engines sorted by *strength*.

This research found no real consensus amongst the search engines studied for the search terms chosen. As shown in table 7.7 there was little overlap in the results returned by the major search engines.

Two possible explanations for this lack of consensus amongst search engines are:

a) There is a genuine lack of consensus amongst experts in the search topic about what are the important resources on that topic. A good search engine would, in this case, return pointers to all of the important competing resources.

b) The current state of search engine technology does not provide searchers with a good set of resources but produces search results that may only cover a part of the topic under investigation.

Had this lack of consensus been clearly due to the differing requirements of the search engine's users, then the different results may have been expected. However the search engines studied are general purpose ones and the choice of which to use is based upon personal preference. Moreover the searches were done in an identical manner and the resulting lists compared. Another possibility is that there may have been amongst the most relevant pages the relevance differed little between the pages. Investigation of individual cases did not support this possibility.

7.6.2 The use of Word Counts, Links and Directory Structure as an Aid to Internet Search

This section describes the work carried to investigate the Fingerprint approach. Three Fingerprinting techniques were defined, which when applied to a *Sample Web*, produce from it a search tree structure similar to the original web.

The fingerprint approach developed in this work consists of taking a *Sample Web* and forming from it structures which reflect its domain and directory structure and extract from its pages fingerprints that in some way characterise the contents of the pages.

Three examples or types of fingerprints are used here. The first, FP_0, forms an ordered list of words occurring in an html file together with the frequencies with which these words occur in that file. This fingerprint (list of words) represents the contents of the original file.

FP_{+1}, the second type of fingerprint, sums all of the frequencies of the words occurring in those html files in the *Sample Web* that are pointed to by the html file. This fingerprint represents the content of the 'authorities cited through html link' by the html file.

The final fingerprint type, FP_{-1}, sums all the fingerprints of those files in the *Sample Web* that point to the html file in question. This represents the content of those *Sample Web* files that cite a particular file through links.

Domains and directories in the *Sample Web* are reflected by the directory structure that the resulting fingerprints are organised into. Each of the three types of fingerprint, FP_0, FP_{+1} and FP_{-1} has its own directory structure. As well as fingerprints of individual files, the directories contain fingerprints of any subdirectories that they contain. These are formed by combining the fingerprints of the contents of the subdirectories.

These three structures were compared for:

a) How score accumulates as the search progresses as a function of leaf nodes searched?

b) How score accumulates as one goes down the final (ordered) list of results ?

c) How much effort, in terms of leaf nodes visited, was required before all of the results up to and including a particular leaf node are reached?

Here the information about a sample of the web (see section 5.4) and its structure are discussed with some statistics, summarised in table 7.12. This table shows the information about the FP_0, FP_{+1} and FP_{-1} and about HTML files and Links files.

Name	Files	Folders	Domains	Size (MB)	Files per folder	Folders per domain	File size /kb	Folder size /kb	Domain size /kb	Files per domain
FP_0	118857	37730	8830	358	3.2	4.3	3.0	9.5	40.5	13.5
FP_{+1}	13042	5727	4783	70	2.3	1.2	5.4	12.2	14.6	2.7
FP_{-1}	101307	29552	6125	914	3.4	4.8	9.0	30.9	149.2	16.5
HTML	82967	43350	8830	1220	1.9	4.9	14.7	28.1	138.2	9.4
Links	76008	36302	8830	162	2.1	4.1	2.1	4.5	18.3	8.6

Table 7.12 Statistical information about the *Sample Web*.

Although the *Sample Web* used is a very small part of the total web, it was large enough to characterise the web. However, it excludes many features (such as dynamic pages and non HTML pages) that have become increasingly important during the time that this research has been undertaken and could form the base of future research.

7.6.2.1 Search Engine Results and Target Word Count

Because the results returned by the search engines were so different except for the issues discussed under consensus it was not possible to compare search engines in terms of the number of common results. An indirect comparison could be provided by comparing current search engines using the naïve, but relatively simple (to calculate), measure of number of the occurrences of the target words.

Although current search engines do have the information on word frequency in a particular web page available to them, from the results obtained in this research it seems that they do not give this information much weight (see section 7.5).

7.6.2.2 Results, Score and Effort

When a search is performed an *effort* is made and *results* are returned. How 'good' the search was depends upon the 'quality' of the results compared to the 'effort' required to produce those results. In this work *effort* is measured by the number of HTML files that need to be opened. Quality of results is measured by giving each file a '*score*' which in the cases reported is simply the number of target words occurring in the file. For the sample web considered in this work and the particular method of searching and scoring system described in section 5.2.2 the following relationships were plotted:

a) Effort needed to find x% of the total possible score.

b) Search results needed to obtain x% of the total possible score.

c) Effort required to find the top x% of search terms in the results.

Each of these is discussed in the following section.

7.6.2.3 Return on Effort

To show return on effort a plot of cumulative score against the number of files opened as the search progresses is shown in Figure 7.1. This was done for FP_0, FP_{+1} and FP_{-1} in the cases of *Duran*, *TRIZ* and *lean*. Although, as might be expected, differing in the detail, the progress of a search using the FP_0 fingerprint shows the same behaviour for the three main search terms.

Expected differences are due to the distribution of the target words throughout the *Sample Web*. For a cumulative score of 50%, FP_{+1} requires about half the effort required for FP_0.

FP_{-1}, on the other hand does not give results that are so consistent but does show some improvement over FP_0. From these and similar results (see Figures 7.4 to 7.10) it appears that the use of links may well enable larger percentages of score to be found for a given effort. Except for the noise, caused by the granular nature of the search space, the curves obtained start at the origin and going steeply at first and flatten out as they reach towards (100%, 100%). When comparing two curves those that are 'higher' are better because they obtain a higher score for a given effort.

In the case of *Duran*, FP_{+1} is overall better than FP_{-1} which is in turn better than FP_0. This pattern is repeated in the case of *TRIZ* but is less clear in the case of *lean*.

The trends seem to be that the higher the frequency of the search word in the *Sample Web* the relatively higher the return on effort is. In a *Sample Web* that contains a larger proportion of the web than that of the current study, it is reasonable to expect that these results will be even more pronounced as the effects of individual pages will be on average less pronounced.

7.6.2.4 Rank and Cumulative Score

When using the results of a search it is often difficult to know when to look no further down the list. A means of showing how the score of the results reduces as the rank of the result increases a graph of rank against cumulative score is plotted. These graphs can be read by looking at the 'score required' on the x-axis and reading the number of results needed for that much of the score to be included, on the y-axis.

In this case a lower curve represents one which requires less results (as a percentage of the total) for a given percentage of score. This indicates that the score is concentrated in relatively fewer results than for a higher curve. In this work representative values of 10%, 25%, 50%, 75% and 100% are used. Here concern is focused on how score is distributed amongst the results. Looking at the percentage of results needed to obtain a given percentage score the results show that words such as *Duran* seem to have their score concentrated in relatively less results than do less frequent words such as *lean* when searching using FP_0. This may be due to the fact that *Duran* is a more specialised word whilst *lean*, with a broader range of uses, has many alternative ways of being expressed. With FP_{+1}, however, there does not seem to be the same clear trend. This could be due to the particular web pages downloaded or it may be a property of FP_{+1}. Plots of this are given for *Duran* (see Figure 7.18), *TRIZ* (see Figure B.4) and *lean* (see Figure B.3). On each plot curves for FP_0, FP_{+1} and FP_{-1} are included for comparison.

In the case of *Duran*, $FP_{+1} < FP_0 < FP_{-1}$, indicating that a larger proportion of the target words occur in relatively fewer FP_{+1} files than is the case for FP_0, which in turn has relatively more words than the FP_{-1} case. Corresponding graphs for *TRIZ* and *lean* (see appendix B) show a different ordering for the curves with $FP_{+1}<FP_{-1}<FP_0$.

7.6.2.5 Effort Needed to Obtain a Given Percentage of Score in the Best Results

Once results have been found it is possible to look at the 'cost' of finding them in terms of effort required. This is a different situation than in section 7.4.3 where the results were reported in the order found. Here the effort required for a given percentage of score, which occurs in the best results, is found. This can only be done with hindsight and is useful because, if the results can be shown to generalise to the full web, it indicates the sort of effort required to obtain the best results.

The graphs are again drawn for FP_0, FP_{+1} and FP_{-1} for the three cases *Duran*, *TRIZ* and *lean*. The general shape of the curves is quite different from that of the previous graphs. The curves are much less smooth. This is because of the nature of the relationship that is being plotted.

In a real search there is no reason for a result with final rank, n say, to be found before one with final rank m just because $n<m$ and so the effort required to find the n^{th} and the m^{th} might well be the same. For example, if the 10^{th} result is found before the fifth, then the effort needed for the fifth will be the same as that for the tenth. This causes relatively larger changes in the curves, giving them a much less smooth look.

The three sets of curves agree in general with $FP_0 > FP_{-1} > FP_{+1}$ for *Duran* and *TRIZ*. In the case of *lean* the curves for FP_0 and FP_{-1}, cross a number of times and so relative ordering of these two cannot be made. As implemented in this research, the Fingerprint technique requires that a web like structure is searched for score.

In order to obtain the best results containing a certain percentage of the possible score the search must expend a certain amount of effort which is here measured in terms of the percentage of files opened.

Three regions on these graphs can be identified:

- At first there is an overhead region in which little return results from the effort expended.

- Later as the search has found areas rich in score the results from a small effort are good.

- Finally nearer the end of the search when most of the files rich in score have been found much effort is required for each extra increment of score.

The curves in Figure 7.26 do indeed show that initially (ie., for low cumulative scores) much effort is required for little return. This represents the search overhead. After about 25% of the files have been opened, the score accumulates rapidly as the ISA has navigated to those parts of the web that contain files rich in the search term. After about 50% of the files have been opened the return on effort starts to decrease as the remaining files have smaller and smaller scores.

In Figure 7.21 it seems that there is not much difference in the behaviour of the graphs for this search when the search terms are *Duran*, *TRIZ* and *lean* and so it is concluded that word frequency is not an important factor for such search terms. However when the corresponding curves for the low frequency words are added in Figure 7.21 it does seem that the overhead is much less important the higher the frequency.

Fingerprints thus will help the search process, the user being given either a choice of which to use, or, more effectively, the ability to combine fingerprint types to produce the mixture that best matches their requirements.

7.6.3 The Nature of Search Results Found by the ISA Technique

Having discussed the Fingerprint approach, focus is then turned to looking at how individual search engines perform. In this study for visualising the search space a search tree approach is used. Figures 7.27 and 7.28 show parts of the trees for the search for *Duran* using Google and ISA. In this section the nature of search results found by current search engines compared to pages with the highest ISA scores is summarised.

Although each search engine found web pages that were related to the search term, these pages were in general not the ones that contained the search term many times. In fact some of the pages appearing near the top of the list of results of conventional search engines did not have the target word appearing in its text but instead only appearing in the meta data or html code.

Many of the highest scoring pages might well be expected to be 'spam' but there were some *bona fide* web sites that seemed to have been overlooked by the traditional search engines. These tended to be on the web sites of keen individuals. For example an important Duran Duran web site is "http://there.indyramp.com/". ISA found "http://there.indyramp.com/fiction/purity.htm" to be the page with the highest score for Duran in the sample web. The former page is easily accessible from the latter but neither page appeared in the results of the other search engines.

ISA (and fingerprinting) promises to be a useful extension to current search engine algorithms, it will be interesting to see how such techniques are best incorporated into the current crop of such search engines such as Yahoo and Google.

7.7 Summary

The *Sample Web* that was used in this research, though large enough to cause problems to the operating system(Microsoft Windows 95 see section 6.4.1) used, was a very small part of that which would be needed for a search engine to be useful to the general public. However, even for this sample it seems clear that the three Fingerprints FP_0, FP_{-1} and FP_{+1} do show differing behaviour that might be helpful in giving a search engine based on these ideas the possibility of giving a user more control over the results of the search performed.

The trends that appear to occur as one goes from words of low frequency (between 10 and 100 occurrences in the whole sample) in the *Sample Web* to the higher frequencies of *TRIZ* and *Duran* suggest that the techniques of fingerprints will prove to be more beneficial as the size of the *Sample Web* increases.

Only further experimentation with a *Sample Web* size one or two orders of magnitude larger than that of the present study will give sufficiently robust results to enable a judgement on whether the techniques will be worth developing into a commercial search engine.

Having discussed the experiments, one can move to the final chapter, which reviews the main results and conclusions of the overall research and identifying the possible direction for future research.

CHAPTER 8

8 Conclusions and Further Work

This research has looked at the distribution of search terms in a sample of the Web, the distribution of search terms returned by a number of commercial search engines and the consensus, or lack of it, amongst search engines, in terms of results returned when given the same search terms. As far as the author is aware this is the first time a systematic study of these questions has been attempted.

Although it is understood that no measure of consensus amongst search engines is available in the literature, one that seems reasonable to the author, *Strength*, has been proposed (see chapter5). Against this measure it can be concluded that, at least at the time that the measurements were made, very little consensus amongst the search engines tested was apparent.

The frequency of occurrence of a search term in a web page (called the *score* of the page in this work) has been used as a measure of the relevance of the page to a query. In principle, any 'good' measure of relevance could be used to obtain results parallel to those of this work. Good is in quotes in the previous sentence as there are quite stringent requirements on this relevance measure for it to be useful, the most important of which are: It must accord with the user's understanding of relevance (this is extremely hard to ensure) and it must be easy to calculate automatically. For the current study *score* is the only measure that satisfies these two criteria.

8.1 Conclusions

The conclusions of this work are divided into two parts:

1) The potential of fingerprints as an aid to search.

2) Search engine performance, the apparent lack of consensus amongst conventional search engines and the nature of search results

8.1.1 Fingerprints and the Presentation of Search Results

Even with score being used as a measure of relevance, alternatives to the use of the page's own score can be produced and have their uses in search. The graphs in sections (7.4.3 to 7.4.5) show that the three fingerprints studied in this research (FP_{-1} FP_0 and FP_{+1}). show similar behaviour to each other. Each has its strengths depending on whether the user wants pages that are rich in the target word (FP_0), pages that are often referenced (citations or FP_{-1}) or pages that link to those rich in the target word (references or FP_{+1}).

As a general rule FP_{-1} accumulates score faster than FP_{+1} which in turn accumulates score faster than FP_0. It seems likely that giving the user the three lists either separately or in some combination will make their searching experience more focused. The method of presenting the search tree fragments with these three scores, rather than bland lists of results, might well also improve the experience of the searcher. They could allow navigation of the neighbourhood of the results and the ability to explore areas of the web that might have otherwise escaped their notice.

8.1.2 Search Engine Performance

From the (fixed) number of the results returned by a search engine a measure of performance of the engine is defined as the sum of the scores of these pages. This measure is the *Union score*. The second column of Table 8.1 gives a normalised version of the results of table 7.6 for the search engines studied.

Search Engine	***Union score***	***Vigour***	***Strength***
Alta Vista	1	0	100
Google	2	0	94
ISA	100	100	0
Webcrawler	1	1	16
Yahoo	34	11	81

Table 8.1 Normalised values of *Union* score, *Vigour* and *Strength* for search engines results.

It is not surprising that ISA is best against the measure *Union score*, as it has all of the sample web pages available and returns them in score order, what is surprising is the relative low score of the others, though Yahoo scores reasonably well in comparison to the rest.

The *Union score* has two main weaknesses: one in the effects its use may have on the behaviour of web site designers and the second as a measure of value of a set of results returned by a search engine.

The first weakness is that if used as an indicator of web page content, it encourages web site writers to include irrelevant words in order to increase their page's score and thus the position of their pages on the results list of a search engine. Search engine developers already suffer from this problem.

The other main weakness of *Union score* is that it does not take into account the order in which the results are presented by the search engine. This means that two search engines that return the same set of results will obtain the same *Union score* despite one returning pages with higher score earlier in its list than the other.

A measure that does take this into account is *Vigour*, which is the sum of the scores weighted by reciprocal rank. Column 3 of table 8.1 shows a normalized version of the *Vigour* tabulated in table 7.6

Instead of using raw score as a measure of page relevance, the ranks of the page given by the search engines as a group can be used to derive a measure of relevance. The sum of reciprocal ranks was used here as it is higher for pages higher in the ranking. This property is an important one and the resulting measure is called *Community Relevance*.

This measure of relevance can be used to evaluate an individual search engine by adding the community relevancies of a fixed number of its results (or all of them if it does not return enough). It has the weakness that a search engine can be thought to 'self justify' its inclusion of a page in its results. This weakness can be overcome by adjusting the community relevance so that it excludes the contribution of the search engine being evaluated.

A further refinement is to weight the community relevance by the page's reciprocal rank that the search engine gives it. The resulting measure is called the *Strength* of the search engine. In the case of this research the maximum possible *Strength* is 6.5.

$$[= \sum s_{py} = (r_{pa} + r_{pg} + r_{pi} + r_{pw}) * r_{py} = \sum_{n=1,\,50} 4/n * 1/n]$$

Column four of table 8.1 gives this measure for the search engines studied. In this case Alta Vista is best and so its score is used to normalise the others. ISA obtains a score of 0. This latter result is not surprising since ISA's results do not agree (in the first 50 places at least) with any of the other search engines.

It seems strange that Alta Vista should do so well here despite its poor score against the other measures. This is because, although Alta Vista does not return many occurrences of the search term in its results pages, it does agree with the others more often than any other agrees with the rest.

A weakness of *Strength* is its dependence on the number of search engines under consideration. However, although this could easily be remedied, by for example dividing by the total number of search engines in a study, this improvement does not significantly add anything to the measure's ability to discriminate between search engines.

Union score, *Strength* and *Vigour* are all measures of search engine performance. The varying results here confirm the commonly held view that no one search engine is best. If raw score is needed, something like ISA should be sought or failing that Yahoo. If results that are common to search engines are valued then Alta Vista may be better.

8.2 Recommendations for Further Work

Given the current state of knowledge of effective search, it is not too difficult to find interesting directions for further work. Having said this it may prove extremely difficult to find solutions to the real problems of search. One is trying to impose on the web a topology or a metric (or more accurately a set of topologies or metrics) that tells when two or more pages are related by some common topic. They are then said to be 'close together' in the language of metric spaces.

The Internet and the Web are the beginnings of what is almost certainly to become an integrated global source of information, which has a property of persistence. At present it is neither truly global nor is it integrated in that pages produced by individuals may or may not be listed in the results of search engines depending, not upon their content but upon the efforts that are made to have them listed. The lack of persistence of the Web is also legendary.

Many avenues of research are suggested by this study. These range from investigations of how best to form or to collect fingerprints to how they could be tied in with user queries. Here are some examples:

a) The ISA method uses directory structures in parallel with the URL (domain and directory) structure of the web has proved to be a useful method of organising information to aid search. It has also suggested that the results might be better presented in a way that shows the relationship between the results and the URL. Extending the work presented here, by for example monitoring real users searching using an ISA based search engine that presents pages with their scores and in a way reflecting the relationships between the URLs of the results, is likely to be fruitful.

b) Different forms of fingerprint are possible. In this research FP_0, FP_{+1} and FP_{-1} have been studied. Natural extensions to FP_{-n} and FP_{+n} are possible but likely to be less obviously useful in application. However this does need to be looked at before being dismissed.

c) It is not clear that the extraction of fingerprints from a web page should be done centrally by search engine providers. An alternative would be for web page authors or Internet services providers, which host the web pages to provide these. Many mechanisms, which would allow this to happen, are possible, finding one that works well and assuring the quality of the resulting data requires investigation.

d) This research has looked at simple search terms as the basis for producing scores when matching fingerprints against search queries. Many other measures of the fit of a fingerprint against a user's query are possible and again investigation of the forms that these could take and the mechanisms by which users make their intentions known, need to be undertaken.

e) The use of 'Weblog' data as an alternative structure of the Web is an area of potential. Many commercial sites use (or at least claim to use) Weblog data to prompt the user. This often takes the form of statements such as 'other users also bought ...' etc.

f) The use of other 'measures of proximity' of web pages. In other words can one formulate ways of comparing web pages to give a measure of how closely their contents relate?

g) Finally, the web is a dynamic body of information changing both in content and in form. New formats are developed constantly and ways to incorporate these into the search engine repertoire need to be found.

In conclusion there is much left to be done but it may prove to be very hard to make real progress in this difficult subject. However, the rewards of success may be considerable for anyone who solves or circumvents these problems. A quick look at the fortunes of the major search engine providers will show what could be expected.

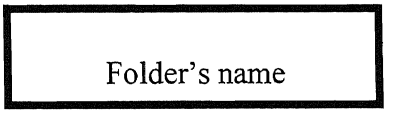
Appendix (A)

Graphical Representation of *Sample Web*

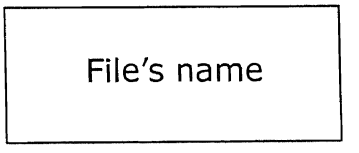
This section presents a summary of directory stricture used to store web page in *Sample Web*. These are included to reinforce background information on Fingerprint technique used to process searching *Sample Web*. The symbol is used for diagrams are:

Key shape

- This shape represents a Folder



- This shape represents a File



- This shape represents a File' content

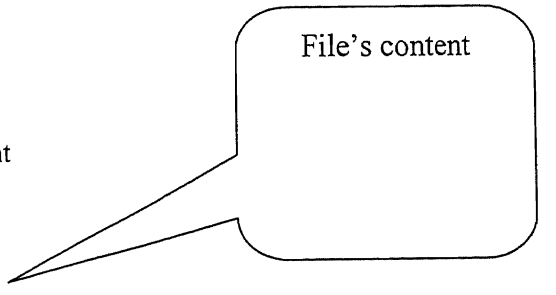


Illustration of the HTML structure in the *Sample Web*

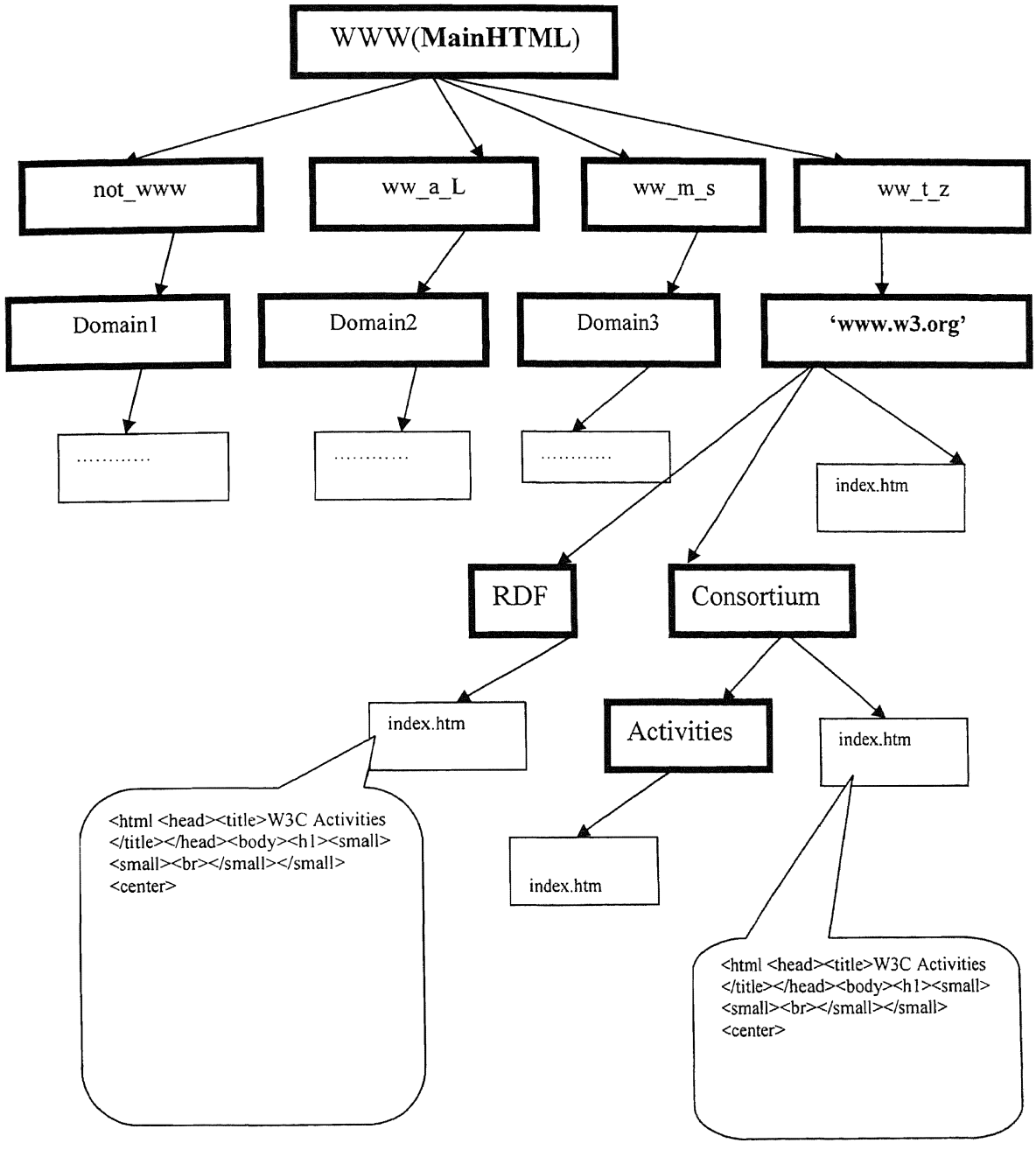


Figure A. 1 Tree structure for HTML files in the *Sample Web*.

Figure A.1 shows the HTML files in the *Sample Web* are organised in a tree structure with the HTML files as leaves of the tree.

Illustrating of the Link structure in the *Sample Web*

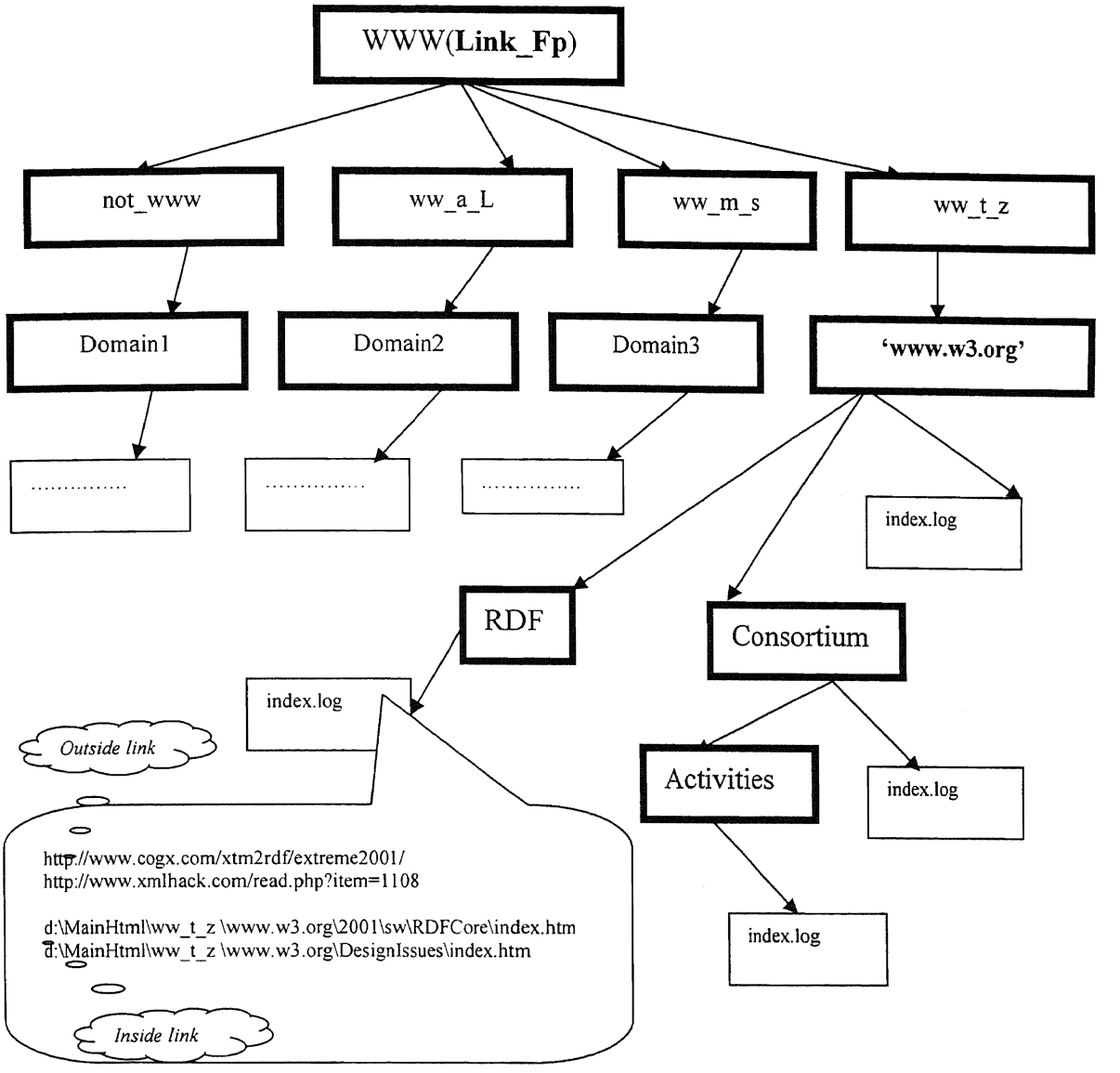


Figure A. 2 Tree structure for Reference files in the *Sample Web*.

Figure A.2 shows the Link files in the *Sample Web* are organised in a tree structure with the Link files as leaves of the tree. Link file contains only hyperlinks from original web page.

Illustrating of the FPF structure in the *Sample Web*

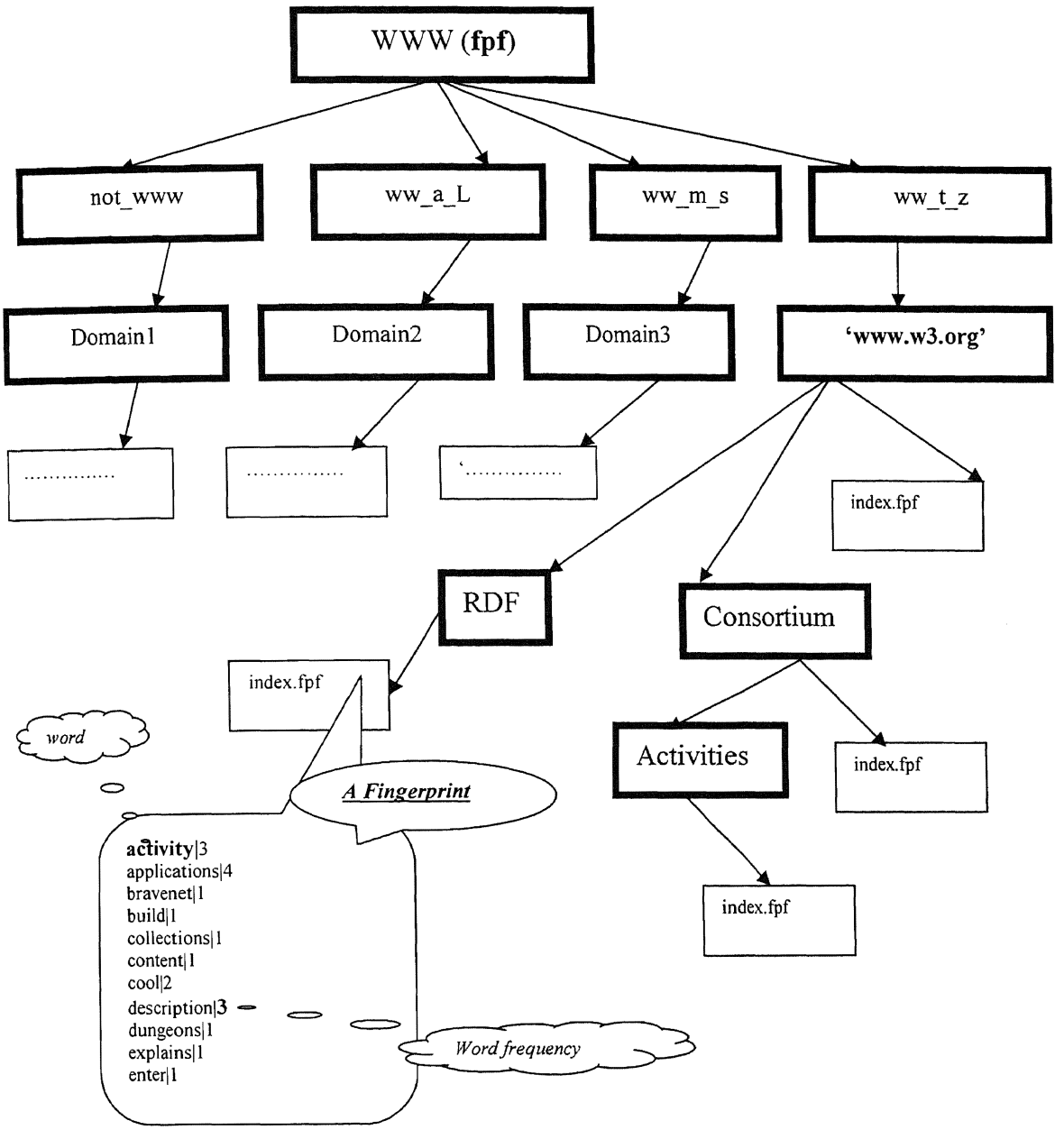


Figure A. 3 Tree structure for FPF files in the *Sample Web*.

Figure A.3 shows the FPF files in the *Sample Web* are organised in a tree structure with the FPF files as leaves of the tree. FPF file contains word count of a web page.

Illustrating of the FP_0 structure in the *Sample Web*

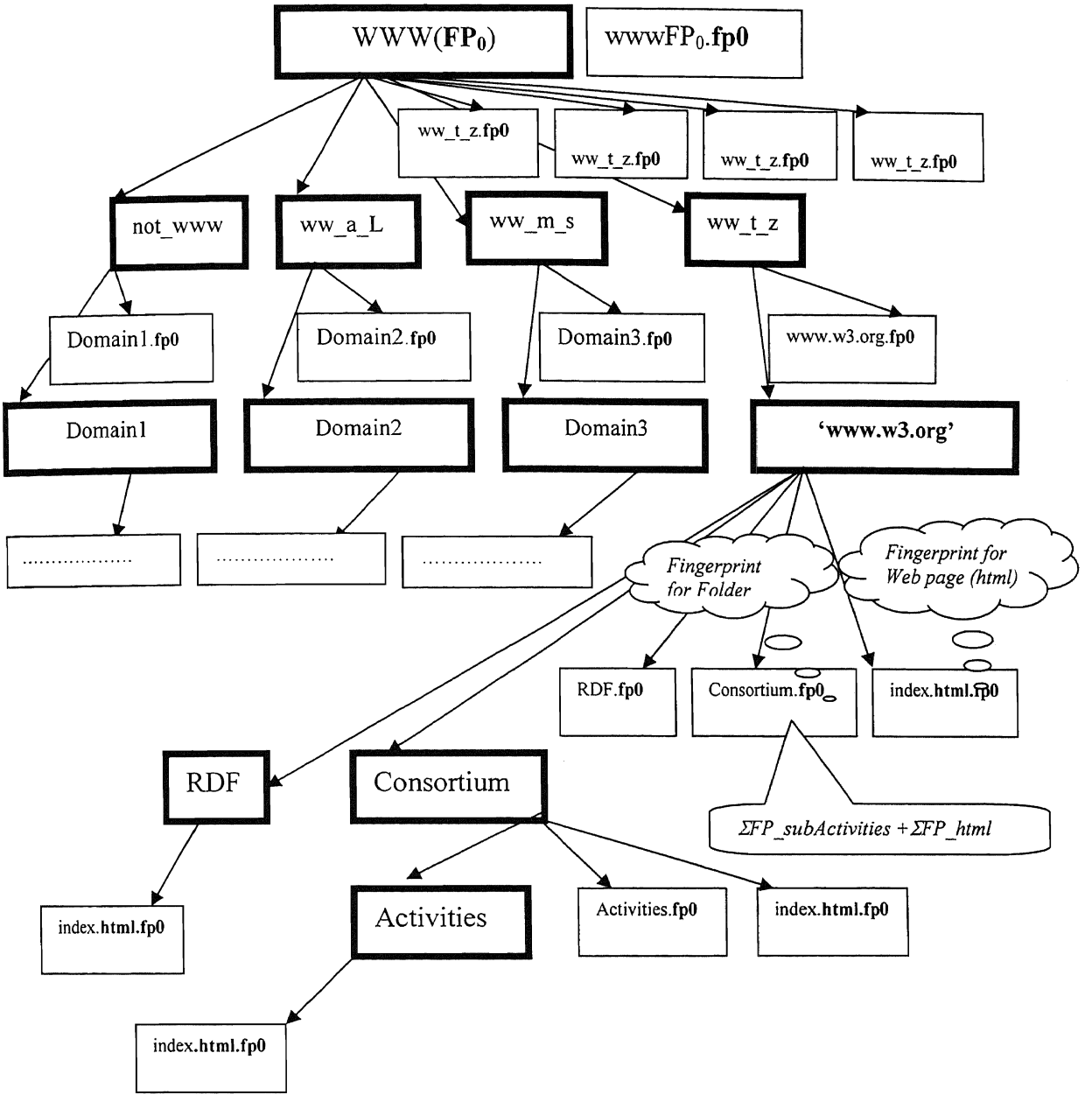


Figure A. 4 Tree structure for FP_0 files in the *Sample Web*.

Figure A.4 shows the FP_0 files in the *Sample Web* are organised in a tree structure with the FP_0 files as leaves of the tree. For this fingerprints of the folders have been added to the previous FPF structure.

Illustrating of the FP_{+1} structure in the *Sample Web*

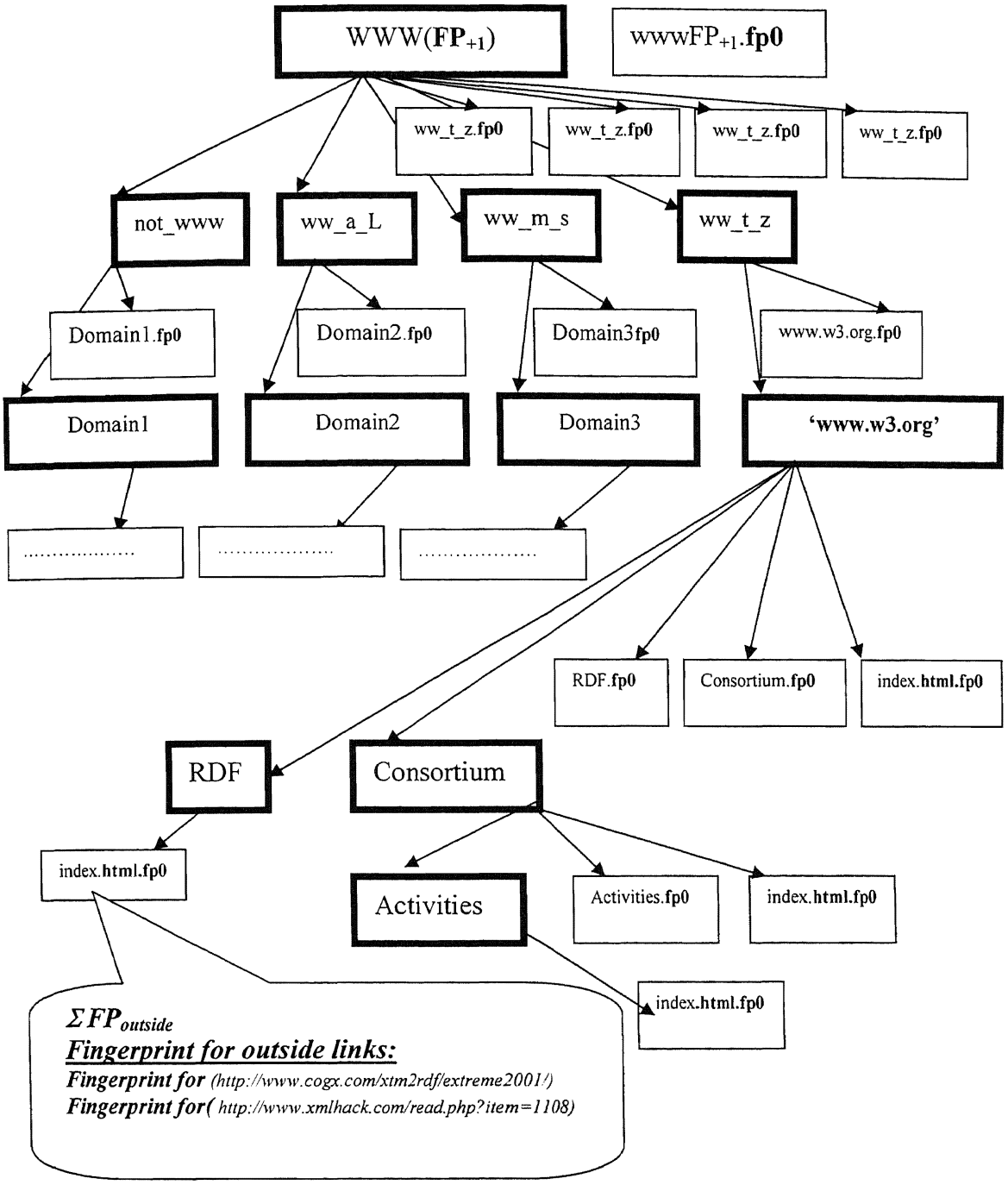


Figure A. 5 Tree structure for FP_{+1} files in the *Sample Web*.

Figure A.5 shows the FP_{+1} files in the *Sample Web* are organised in a tree structure with the FP_{+1} files as leaves of the tree. The FP_{+1} files contain word count of web pages and the sum of the histograms of its immediate **successors**.

Illustrating of the FP_{-1} structure in the *Sample Web*

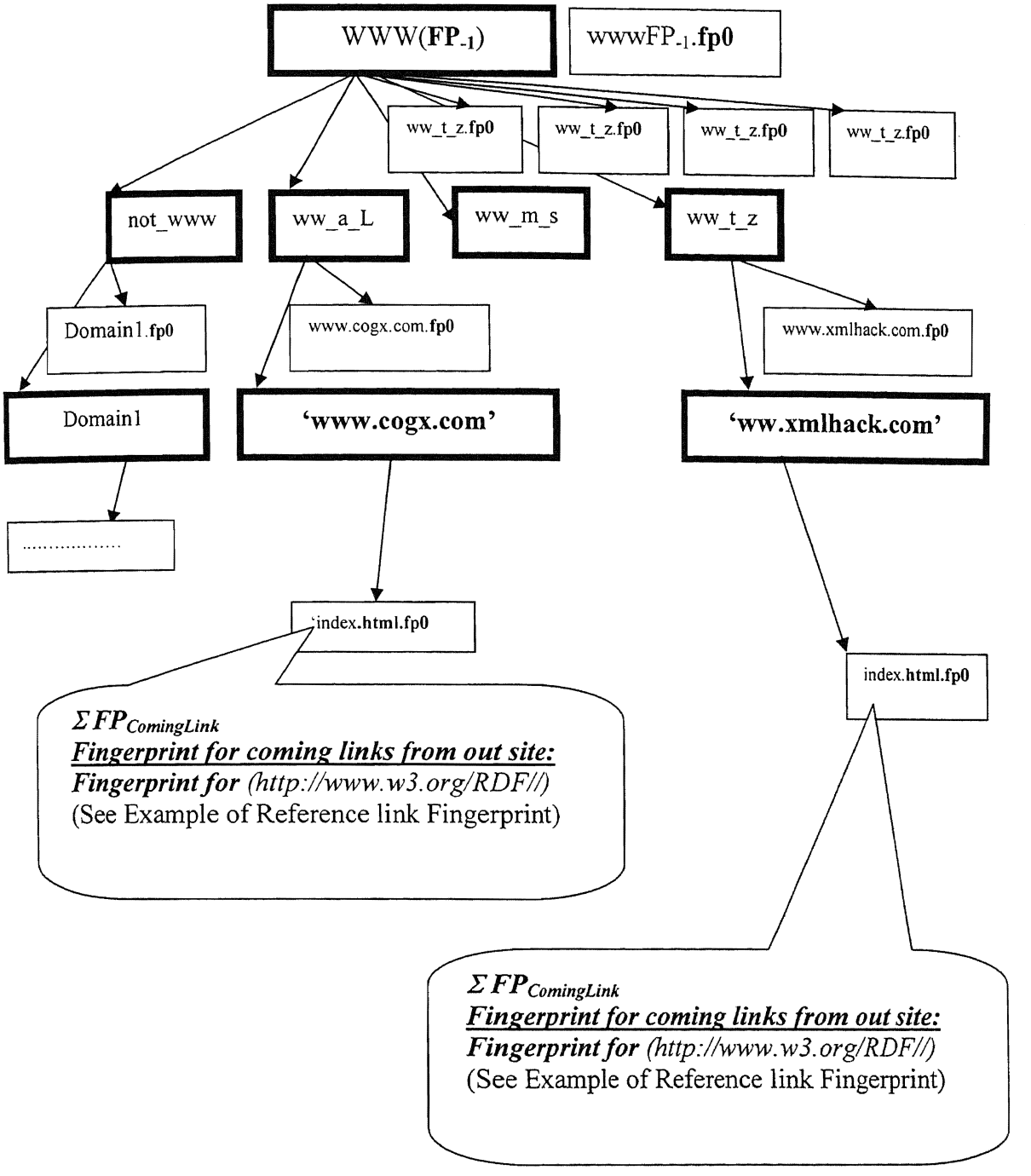


Figure A. 6 Tree structure for FP_{-1} files in the *Sample Web*.

Figure A.6 shows the FP_{-1} files in the *Sample Web* are organised in a tree structure with the FP_{-1} files as leaves of the tree. The FP_{-1} file contains word counts of the web page and the sum of the histograms of its immediate **predecessors**.

Appendix (B)

Plots for Search Term *TRIZ* and *lean*

This section presents a summary of the search progress; a typical search was carried out for the term '*TRIZ*' and '*lean*', by means of a graph showing cumulative score (word count in the examples) versus files opened (or effort expended). These graphs are called 'type A' in this thesis. The next graph is a graph describing the way those target words are distributed throughout the search. This type of graph is referred to as 'type B'. Graphs of effort required versus cumulative score "best results" (typ C) show the effort required (in terms of the percentage of html files) that must be searched to achieve a given percentage score when this score is required in the *best results order.*

Plots of the corresponding results for *TRIZ* (Figure B.4) are with the above in general the search is 'quicker' for FP_{+1} than for FP_{-1} , which is in turn 'quicker' than FP_0 . This is evidence that the FP_{-1} and FP_{+1} Fingerprints can indeed aid in directing a search towards areas, which may prove relevant of the *Sample Web* answering the main research question in the affirmative. In the case of *lean*, however the curves for FP_{-1} and FP_0 cross a number of times and so no generalisation can be made. These and other results reported later indicate that the results for lean suffer from it having a relatively low frequency in the *Sample Web.*

Figures Lean for All FPs(A)

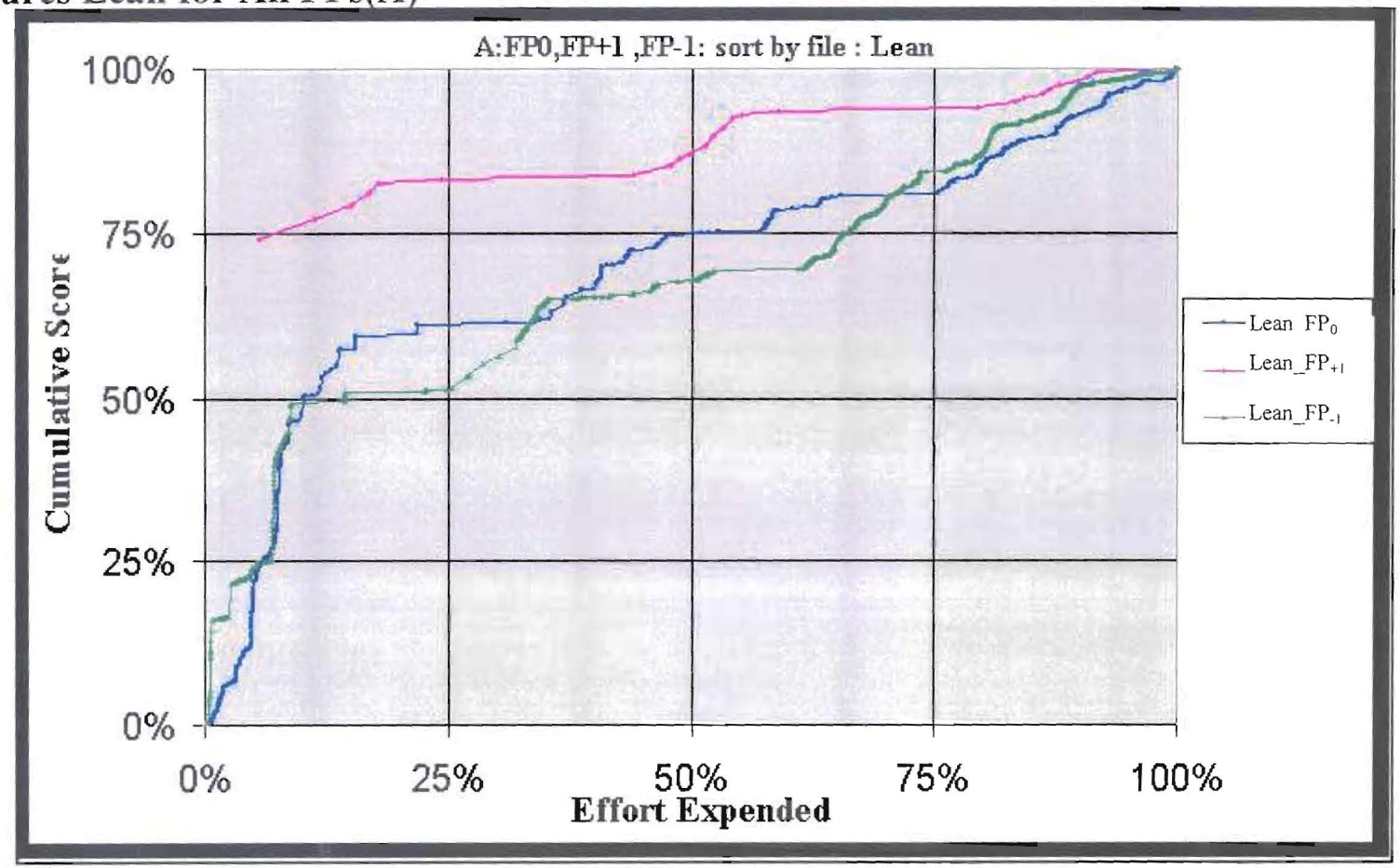


Figure B. 1 FP_0, FP_{-1} and FP_{+1}, score accumulating as file open of *lean*. Combined graph of ISA FP_0, FP_{-1} and FP_{+1} searches, showing score accumulating as a function of html files opened in the case of *lean* ("A").

95

Figures Triz for All FPs(A)

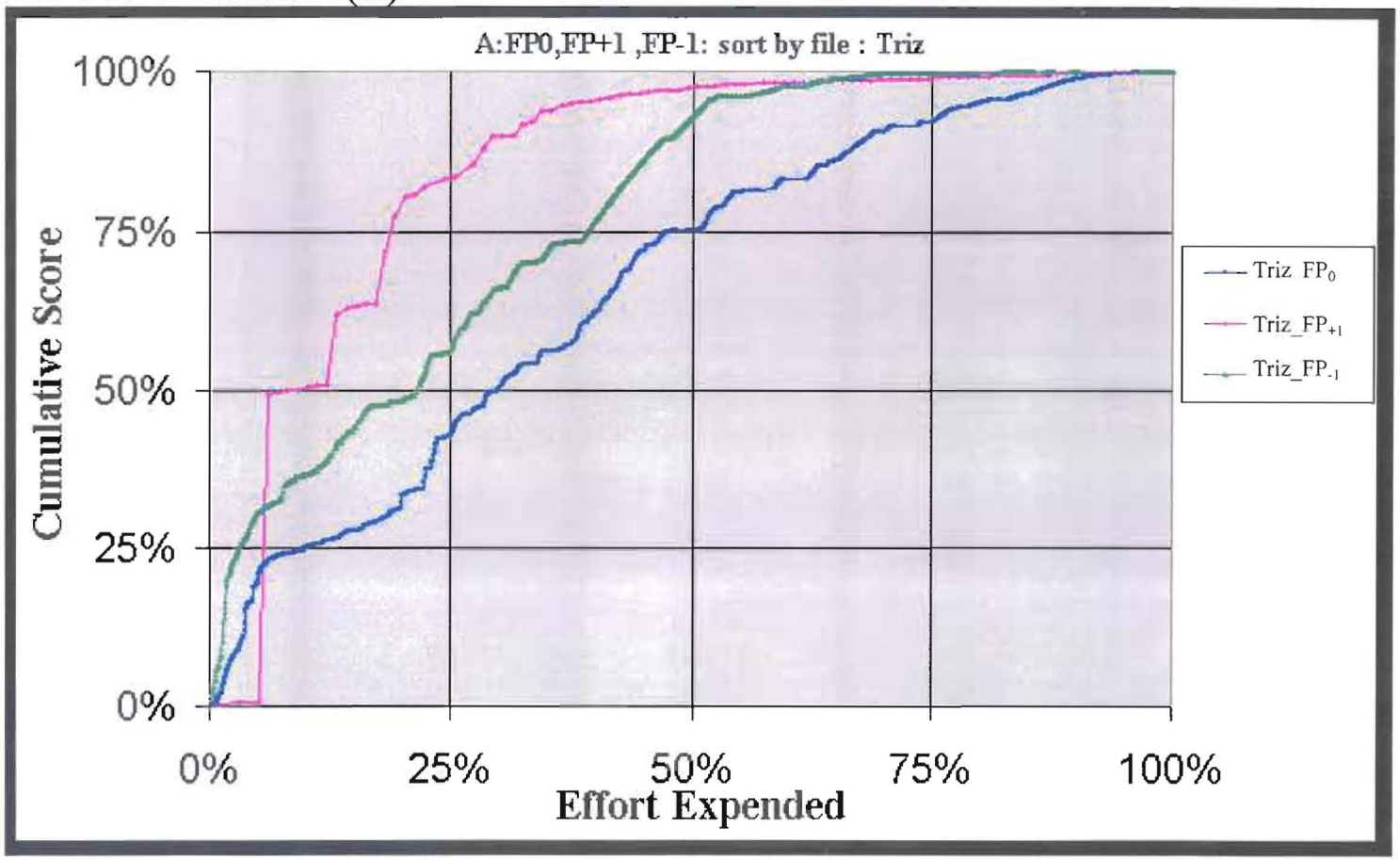


Figure B. 2 FP_0 , FP_{-1} and FP_{+1}, score accumulating as files opened of *TRIZ*. Combined graph of ISA FP_0, FP_{-1} and FP_{+1} searches, showing score accumulating as a function of html files opened in the case of *TRIZ* ("A").

Figures Lean for All FPs(B)

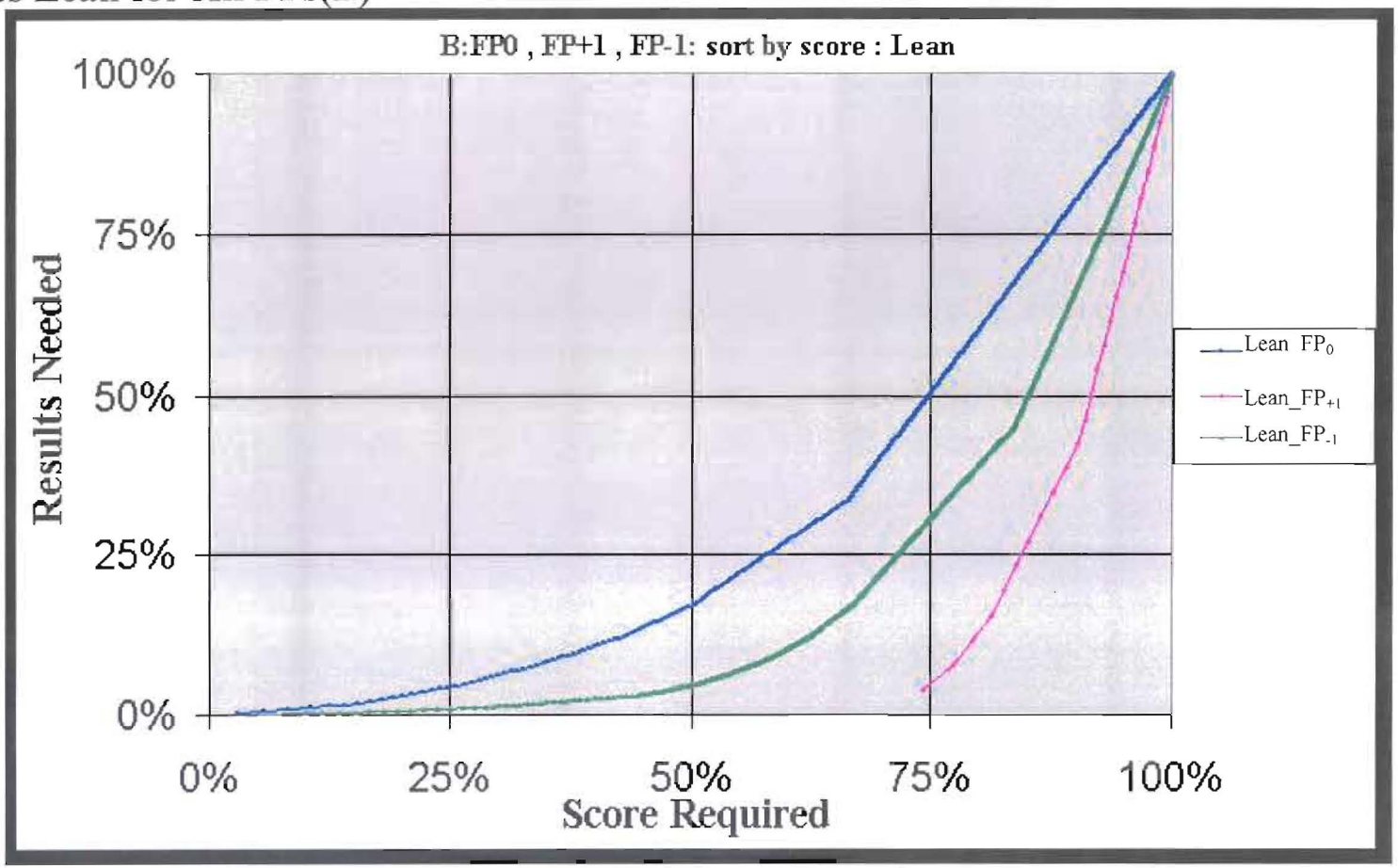


Figure B. 3 FP_0, FP_{-1} and FP_{+1} , for Results needed vs score required of *lean* . Combined graph of ISA for FP_0, FP_{-1} and FP_{+1} searches, showing many result files must be taken in order to have a particular percentage score in the case of *lean* ("B").

Figures Triz for All FPs(B)

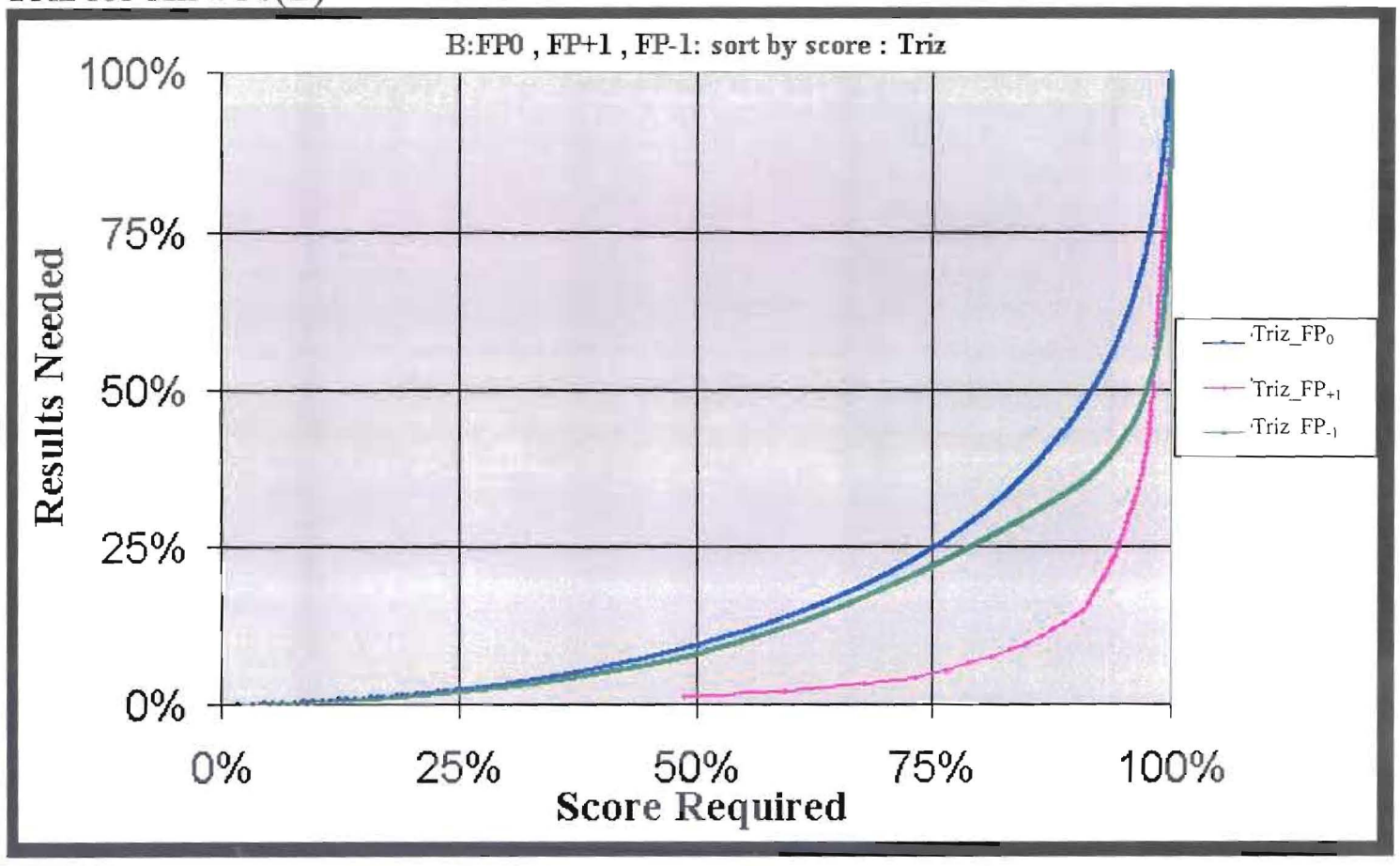


Figure B. 4 FP_0, FP_{-1} and FP_{+1} for Results needed vs score required of *TRIZ*. Combined graph of ISA for FP_0, FP_{-1} and FP_{+1} searches, showing many result files must be taken in order to have a particular percentage score in the case of *TRIZ* ("B").

Figures Lean for All FPs(C)

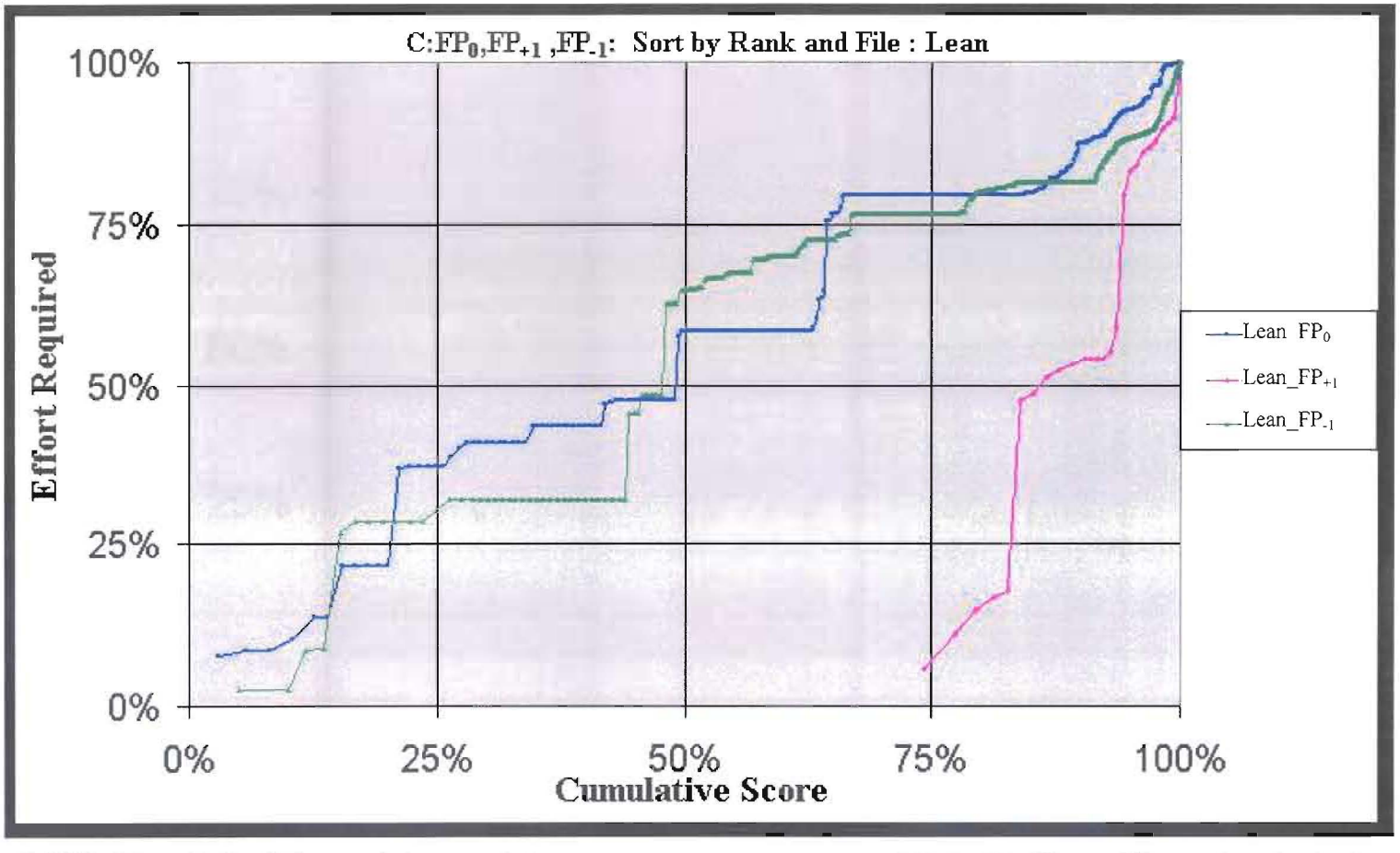


Figure B. 5 FP_0, FP_{-1} and FP_{+1} , Effort required vs cumulative score of *lean*. Combined graph of ISA for FP_0, FP_{-1} and FP_{+1} searches, showing how many html files need to be visited in order to accumulate a particular score in the case of *lean* ("C").

Figures Triz for All FPs(C)

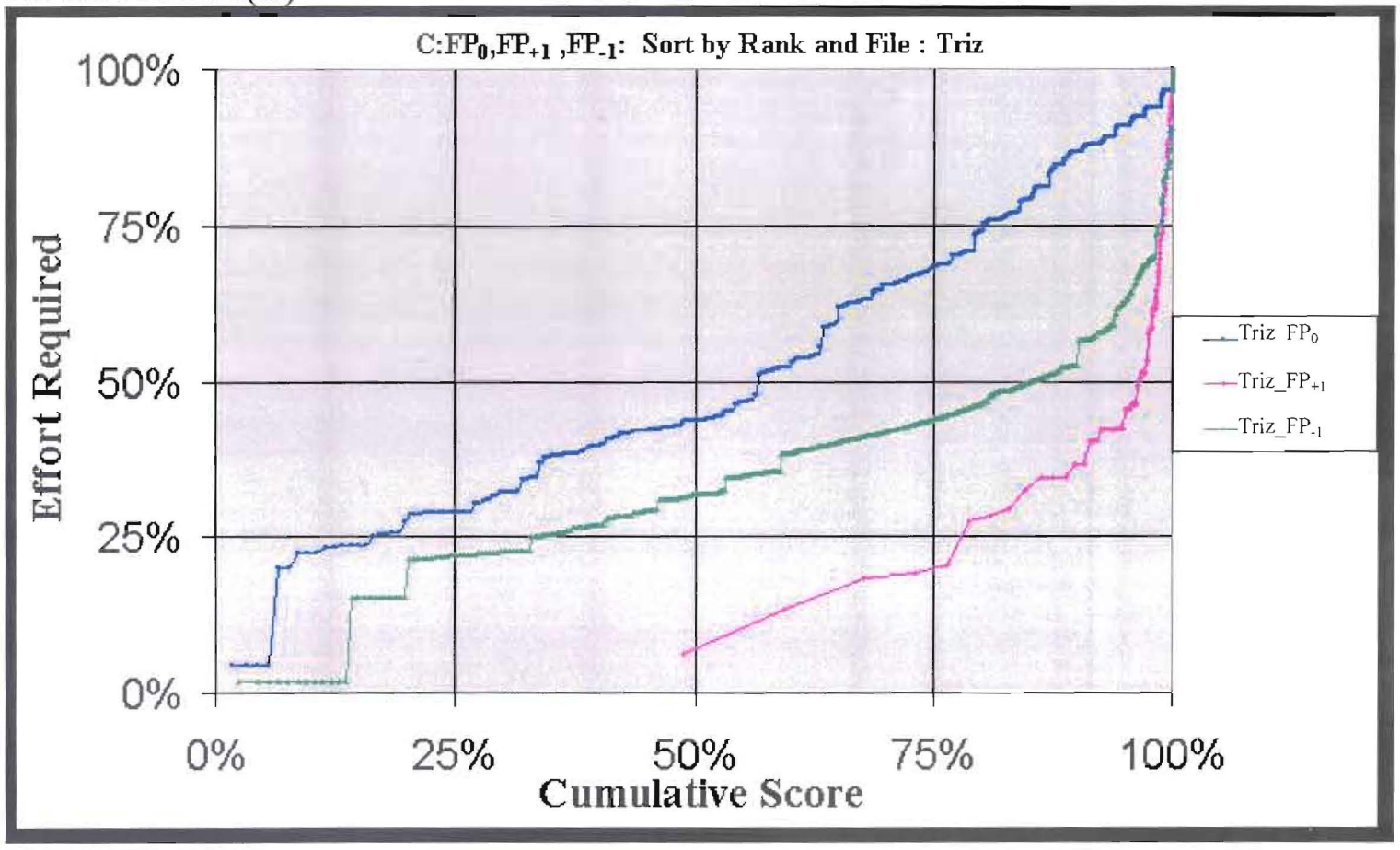


Figure B. 6 FP_0, FP_{-1} and FP_{+1} , Effort required vs cumulative score of *TRIZ*. Combined graph of ISA FP_0 , FP_{-1} and FP_{+1} searches, showing how many html files need to be visited in order to accumulate a particular score in the case of *TRIZ* ("C").

		Word Frequency		
		FP_0	FP_{+1}	FP_{-1}
Words in *Sample Web*	**Duran**	28306	23474	449074
	TRIZ	22031	14039	388287
	Lean	757	155	4154

Table B. 1 Frequency number for three main search terms, in *Sample Web*.

Frequency number for three main search terms, which occur with frequency in *Sample Web* for FP_0 , FP_{+1} and FP_{-1} .

Appendix (C)

Sample of ISA Source Code

ISA is written in Microsoft's Visual Basic version 6. It consists of eight modules (Controller, Fingerprint, FP_search, Internet_tools, Parse_HTML, Strings_tools, System_tool, and Link_harvester) with some nine thousand lines of code between them. Here we list some of the code from the module called *Fingerprint* that deals with the fingerprint aspect of the program is listed.

```
************************************
'Windows API/Global Declarations for :Sy
'    stem
'************************************
Option Explicit

Private Type SHFILEOPSTRUCT
   hWnd As Long
   wFunc As Long
   pFrom As String
   pTo As String
   fFlags As Integer
   fAborted As Boolean
   hNameMaps As Long
   sProgress As String
   End Type

Private Type BrowseInfo
   hwndOwner As Long
   pIDLRoot As Long
   pszDisplayName As Long
   lpszTitle As Long
   ulFlags As Long
   lpfnCallback As Long
   lParam As Long
   iImage As Long
   End Type
   Private FileDestination As String
   Private Const BIF_RETURNONLYFSDIRS = 1
   Private Const MAX_PATH = 260

Private Declare Sub CoTaskMemFree Lib "ole32.dll" (ByVal hMem As Long)

Public Declare Function lstrcat Lib "kernel32" Alias "lstrcatA" _
   (ByVal lpString1 As String, ByVal lpString2 As String) As Long

Private Declare Function SHBrowseForFolder Lib "shell32" _
   (lpbi As BrowseInfo) As Long

Private Declare Function SHGetPathFromIDList Lib "shell32" _
   (ByVal pidList As Long, ByVal lpBuffer As String) As Long
   Private Const FO_COPY = &H2
   Private Const FO_DELETE = &H3
   Private Const FO_MOVE = &H1
   Private Const FO_RENAME = &H4
   Private Const FOF_ALLOWUNDO = &H40
   Private Const FOF_CONFIRMMOUSE = &H2
   Private Const FOF_FILESONLY = &H80 ' on *.*, Do only files
   Private Const FOF_MULTIDESTFILES = &H1
   Private Const FOF_NOCONFIRMATION = &H10 ' Don't prompt the user.
   Private Const FOF_NOCONFIRMMKDIR = &H200 ' don't confirm making any needed dirs
   Private Const FOF_RENAMEONCOLLISION = &H8
   Private Const FOF_SILENT = &H4 ' don't create progress/report
   Private Const FOF_SIMPLEPROGRESS = &H100 ' means don't show names of files
   Private Const FOF_WANTMAPPINGHANDLE = &H20 ' Fill in SHFILEOPSTRUCT.hNameMappings
Option Explicit
 Public WebLocalStr As String
 Public FP_LocalStr As String
 Public FolderSelctorFlag As String
 Public Const MaxCharecter = 20
 Public Const MaxSameChar = 5

 Private FF_FP0 As Integer 'FF_FP0 - Used to hold Freefile number
```

```
Private FF_FP1 As Integer 'FF_FP1 - Used to hold Freefile number
Private FF_FPm1 As Integer 'FF_FPm1 - Used to hold Freefile number


Private Const MainName = "MainFP"
Private Const WordFolderName = "Wordfp"
Private Const HtmlFolderName = "MainHtmls"
Private Const FPfolderName = "fpf"
Private Const Minus1_FPFfolderName = "m1_fpf"
Private Const NameFP_m1 = "fp_m1"
Private Const NameFPF1 = "fpf1"
Private Const NameFP1 = "fp1"
Private Const NameFP0 = "fp0"
Private Const LinkFolderName = "Linkfp"
Private Const LogPath = "MetaLink"
Private Const LogName = "isa"
Private Const ExprName = "Exp_Engines"
Private Const FP1ExtName = ".fp1"


'Public listfolder() As String
'Public smt As Long

Function Get_Minus1_FPF() As String
   Get_Minus1_FPF = Minus1_FPFfolderName
End Function
Function Get_FpMinus1() As String
   Get_FpMinus1 = NameFP_m1
End Function
Function GetExprPath() As String
   GetExprPath = ExprName
End Function
Function GetLogPath() As String
   GetLogPath = LogPath
End Function
Function GetLogName() As String
   GetLogName = LogName
End Function
Function GetMainfp() As String
   GetMainfp = MainName
End Function
Function GetWebFolder() As String
   GetWebFolder = HtmlFolderName
End Function
Function GetLinkFolder() As String
   GetLinkFolder = LinkFolderName
End Function
Function GetWordFolder() As String
   GetWordFolder = WordFolderName
End Function
Function GetFP_Folder() As String
   GetFP_Folder = FPfolderName
End Function
Function FP1FolderGet() As String
   FP1FolderGet = NameFP1
End Function
Function Get_FPF1Folder() As String
   Get_FPF1Folder = NameFPF1
End Function
Function FP0FolderGet() As String
   FP0FolderGet = NameFP0
End Function
Function SearchWebLocal(Prog As Boolean) As String
  Dim colFiles As New Collection 'Note 'New' keyword!!

    FindFolder "c:\", HtmlFolderName, colFiles, Prog
    If colFiles.Count > 0 Then
```

```
      SearchWebLocal = colFiles.Item(1)
    Else
      SearchWebLocal = App.Path
    End If

End Function
Function FindLocalFP(Prog As Boolean) As String
  Dim colFiles As New Collection 'Note 'New' keyword!!
  Dim TempStr As String

    FindFolder "c:\", MainName, colFiles, Prog
    If colFiles.Count > 0 Then
      TempStr = PerviousFolder(colFiles.Item(1))
      TempStr = CleanPath(TempStr)
      FindLocalFP = TempStr
    Else
      FindLocalFP = App.Path
    End If

End Function

Function VerifyLocalWeb(AddStr As String) As Boolean
  Dim HtmLeng As Long

    HtmLeng = Len(HtmlFolderName) + 1
    If Right(AddStr, HtmLeng) = (HtmlFolderName & "\") Then
      VerifyLocalWeb = True
    Else
      VerifyLocalWeb = False
    End If
End Function
Function FPVerifyFolder(AddStr As String, fpLevel As String) As Boolean
  Dim FPposi As Long

    FPposi = InStr(1, AddStr, "\" & fpLevel & "\")
    If FPposi > 0 Then
      FPVerifyFolder = True
    Else
      FPVerifyFolder = False
    End If
End Function
Sub DeletScreenRest()
 Dim myform As Form
   Set myform = Screen.ActiveForm
   myform.LabError.Visible = False
   myform.TxtError.Visible = False
   myform.TxtError.Text = 0
   myform.CheFP1maker.Enabled = False
   myform.CheFP1maker.Value = 0
   myform.CheFolderFinder.Value = 1
   myform.CheConverToHtm.Value = 0
   myform.CheConverToHtm.Enabled = True
   myform.CheDeletAllEmpty.Value = 0
   myform.CheWebTxt.Value = 0
   myform.CheWebTxt.Enabled = True
   myform.CheFPGenerator.Value = 1
   myform.CheFPGenerator.Enabled = True
   myform.CheLinkFile.Value = 1
   myform.CheLinkFile.Enabled = True
   myform.FraFpSelect.Enabled = True
  Set myform = Nothing
End Sub
Sub ScreenDeletFolderFile()
 Dim myform As Form
   Set myform = Screen.ActiveForm
    myform.CheConverToHtm.Value = 0
    myform.CheConverToHtm.Enabled = False
```

```
    myform.CheFPGenerator.Value = 0
    myform.CheFPGenerator.Enabled = False
    myform.CheLinkFile.Value = 0
    myform.CheLinkFile.Enabled = False
    myform.CheWebTxt.Value = 0
    myform.CheWebTxt.Enabled = False
    myform.CheFP1maker.Value = 0
    myform.CheFP1maker.Enabled = False
    myform.FraFpSelect.Enabled = False
End Sub
Sub FP1ScreenActive(ErrOK As Boolean, NoErr As Long)
 Dim myform As Form
  Set myform = Screen.ActiveForm
    If ErrOK Then
      myform.LabError.Visible = True
      myform.TxtError.Visible = True
      myform.TxtError.Text = NoErr
    End If
    myform.CheFP1maker.Enabled = True
    myform.CheFP_m1maker.Enabled = True
    myform.CheFolderFinder.Value = 0
    myform.CheDeletAllEmpty.Value = 0
    myform.CheConverToHtm.Value = 0
    myform.CheWebTxt.Value = 0
    myform.CheFPGenerator.Value = 0
    myform.CheLinkFile.Value = 0
  Set myform = Nothing
End Sub
Sub FP1ScreenRest()
 Dim myform As Form
    Set myform = Screen.ActiveForm
    myform.LabError.Visible = False
    myform.TxtError.Visible = False
    myform.TxtError.Text = 0
    myform.CheFP1maker.Enabled = False
    myform.CheFP1maker.Value = 0
    myform.CheFolderFinder.Value = 1
    myform.CheConverToHtm.Value = 1
    myform.CheDeletAllEmpty.Value = 1
    myform.CheWebTxt.Value = 1
    myform.CheFPGenerator.Value = 1
    myform.CheLinkFile.Value = 1
  Set myform = Nothing
End Sub
Function GetLevelOfFP() As Integer
 Dim myform As Form
    Set myform = Screen.ActiveForm
    If myform.Opt_FPm1.Value Then
       GetLevelOfFP = -1
    ElseIf myform.Opt_FP00.Value Then
       GetLevelOfFP = 0
    ElseIf myform.Opt_FP1.Value Then
       GetLevelOfFP = 1
    End If
  Set myform = Nothing
End Function
```

Appendix (D)

Publications during PhD

Some of the work in thesis was presented at a number of conferences:

Minaji, M. (2001). 'Information Technology today: Interactivity and mechanisms for E-Commerce.', *presented at IT summer conference at Kish University*, August. 2001, Kish Island Iran.
Minaji, M. and Vella, A. (1998) 'Why is searching the Internet such a hassle?', *presented at* the *Operational Research Conference OR40*, Lancaster UK.
Minaji, M. and Vella, A. (1999) 'Using a Fingerprint strategy for searching the Internet.', *presented at the Operational Research Conference OR41*, Edinburgh UK.
Moghaddam, N., Vella, A., Minaji, M. (1999). 'Improving genetic search algorithms.', *presented at the Operational Research Conference OR41*, Edinburgh UK.
Vella, A., Minaji, M., Ulla A. (1998). 'Using Artificial Intelligence to aid Management Information Systems.', *presented at the Operational Research Conference OR40*, Lancaster UK.

References

Arlene, G.T. (2003) *The Organisation of Information: Library and Information Science Text Series*, Second Edition, ISBN: 1563089696, pp. 68-93.

Attardi, G., Gullì, A., and Sebastiani, F. (1999) 'Automatic Web Page Categorisation by Link and Context Analysis', in *Proceedings of THAI-99 European Symposium on Telematics*, Hypermedia and Artificial Intelligence.

Baeza-Yates, R., Castillo, C., Marin, M., Rodriguez, A. (2005). 'Crawling a Country: Better Strategies than Breadth-First for Web Page Ordering ', in Proceedings of the Industrial and Practical Experience track of the 14th conference on World Wide Web, Chiba, Japan. ACM Press, ISBN:1595930515, pp.864-872.

Balling, H., Madsen, A.K. (2003). From Homer to Hypertext: Studies in Narrative, Literature and Media , Univ Pr of Southern Denmark, ISBN: 8778386799, pp. 83-86.

Belkin, N. J., and Croft, W. B. (1992). 'Information Filtering and Information Retrieval: Two sides of the same coin?', *Communications of the ACM*, 35(12), pp. 29-38.

Belkin, N.J., Cool, C., Kelly, D., Lee, H.-J., Muresan, G., Tang, M.-C., and Yuan, X.-J. (2003). 'Query length in interactive information retrieval', *in Proceedings of the 26th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR '03)*, Toronto, Canada.

Belkin, N.J., Marchetti, P.G. and Cool., C. (1993) 'BRAQUE: Design of an Interface to Support User Interaction in Information Retrieval', *Information Processing and Management*, 29(3), pp. 325-344.

Bent, L., Rabinovich, L., Voelker, G.M., and Xiao, Z. (2004) 'Characterisation of a Large Web Site Population with Implications for Content Delivery ', *in Proceedings of the thirteenth international World Wide Web Conference*, NY USA, www.www2004.org/proceedings/docs/1p522.pdf, pp. 522-533.

Bhowmick, S.S, Madria, s.k., Ng, W.K. (2003) Web Data Management a warehouse approach, Springer, ISBN:0387001751, pp. 234-241.

Bookstein, A., Kulyukin, V., Raita, T., and Nicholson, J. (2003) 'Adapting measures of clumping strength to assess term-term similarity', *Journal of the American Society forIinformation Science and Technology*, 54(7), pp.611-620.

Bowman, C.M., Peter, B.D., Darren, R.H., Manber, U., and Michael F.S. (1994) 'The Harvest information discovery and access system', in *Proceedings of the 2nd International World Wid Web Conference*, October 1994, pp. 763-771.

Boydell, O., Gurrin, C., Smeaton, A.F., Smyth, B. (2005). 'Manipulating the Relevance Models of Existing Search Engines.', in Proceedings of 27th European Conference on IR Research, ECIR 2005, Spain, Volume 3408/2005,ISBN:3540252959, pp.540.

Bray, T. (2003 a) *A series of essays on search engine techniques* [online], tbray, AntarctiSystems, Available from: http://www.tbray.org/ongoing/When/200x/2003/07/30/OnSearchTOC (Accessed October 2005).

Bray, T. (2003 b) *First article of search engine techniques A series of essays on search engine techniques: extensible Web crawler* [online], tbray, AntarctiSystems, Available from: http://www.tbray.org/ongoing/When/200x/2003/06/15/OnSearch (Accessed October 2005).

Bray, T. (2003 c) *Second article of search engine techniques A series of essays on search engine techniques* [online], tbray, AntarctiSystems, Available from: http://www.tbray.org/ongoing/When/200x/2003/06/17/OnSearch (Accessed October 2005).

Bray, T. (2003d) *Fifth article of search engine techniques A series of essays on search engine techniques* [online], tbray, AntarctiSystems, Available from: http://www.tbray.org/ongoing/When/200x/2003/06/24/IntelligentSearch (Accessed October 2005).

Bray, T. (2003 e) *Ninth article of search engine techniques A series of essays on search engine techniques* [online], tbray, AntarctiSystems, Available from: http://www.tbray.org/ongoing/When/200x/2003/07/29/SearchMeta (Accessed October 2005).

Bröder, A. and Bharat, K. (1998) 'A technique for measuring the relative size and overlap of public Web search engines', in *Proceeding of the 7th World Wide Web Conference*, 1998, (www7.scu.edu.au/programme/fullpapers/1937/com1937.htm; also see update at www.research digital.com /SRC/whatsnew/sem.html).

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer,R., and Rossin, P. (1988), 'A Statistical Approach to Language Translation', in *Proceeding of the 12th International Conference on Computational Linguistics*, Budapest.

Callan, J.P., Croft, W.B., and Harding, S.M. (1992) 'The INQUERY Retrieval System', in *Proceedings of the 3rd International Conference on Database and Expert Systems Applications*, pp. 78-83.

Cameron, R.D. (2001) *Information Retrieval and Search* [online], Simon Fraser University School of Computing Science, Available from:http://www.cs.sfu.ca/~cameron/Teaching/D-Lib/IR.html (Accessed October 2005).

Chakrabarti, S. (2003) *Mining the Web discovering knowledge from Hypertext Data*, ISBN: 1-55860-754-4, pp. 25-73

Chang, C.-H. and Ding, Z.-K. (2004) 'Categorical Data Visualisation and Clustering using Subjective Factors', in *Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery (DaWaK04)*, Zaragoza, Spain, 2004. LNCS 3181 (SCI expanded), pp. 229-238.

Charoenkitkarn, N., Chignell, M.H., and Golovchinsky, G. (1995) 'Interactive Exploration as a Formal Text Retrieval Method: How Well can Interactivity Compensate for Unsophisticated Retrieval Algorithms?', in *Proceedings of the Third Text Retrieval Conference (TREC-3). Harman*, D.K. (Ed.). NIST Special Publication 500-225. Gaithersburg, Maryland. pp. 179-199.

Chignell, M., Bodner, R., Charoenkitkarn, N., Golovchinsky, G., and Kopak, R.W. (2001) 'The Impact of Text Browsing on Text Retrieval Performance', *Information Processing and Management*, 37(3), pp.507-520.

Chowdhury,G.G. (2004).Introduction to Modern Information Retrieval, Second Edition,Facet Publishing, ISBN:1856044807, pp.321-326.

Christopher, L. (2001) *Where Have all the Gophers Gone? Why the Web beat Gopher in the Battle for Protocol Mind Share* [online], University of Michigan School of Information, Available from: http://www.personal.si.umich.edu/~calz/gopherpaper.htm (Accessed October 2005).

Cooper, W.S. (1988) 'Getting beyond Boole', *Information Processing and Management*, 24, pp. 243-225.

De Bra, P.M. E. and Post, R. D. J. (1994) 'Searching for arbitrary information in the WWW: The fish search for Mosaic', in *Proceedings Of the 2nd International. World Wide Web Conference*, Chicago, October 1994. http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/www-fall94.html.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., and Harshman, R. (1990) 'Indexing by latent semantic analysis', *Journal of the American Society for Information Science*, 41 (6), pp. 391–407.

der Linden, P.V. (1997) *Just Java the sunsoft press Java series*, Second Edition, ISBN:0-13-272303-4, pp. 67-83.

Drabenstott, K.M. and Weller, M.S. (1996) 'The exact-display approach for online catalogue subject searching', *Information Processing and Management*, 32(6), pp. 719-745.

Edwards, J., McCurley, K.S., Tomlin, J.A. (2001). 'An adaptive model for optimizing performance of an incremental web crawler.' in Proceedings of the Tenth Conference on World Wide Web, Hong Kong, Elsevier Science, pp.106–113.

Fieldsend, J. E., Fisher, M. J., and Everson, R. M.(2004). 'Precision and recall optimisation for information access tasks.' in Proceedings of the ROCAI Workshop at the European Conference on Artificial Intelligence (ECAI'04), Valencia, pp.45-54.

Fisher, M.J., and Everson, R. M. (2003). 'When are links useful? Experiments in text classification.' in Proceedings ofthe Twenty Fifth European Conference on IR Research (ECIR'03), Pisa, pp.41-56.

Frakes, W. B. and Yates, R.-B. (1992) *Information retrieval: Data structures & algorithms*, Englewood Cliffs, NJ: Prentice Hall, ISBN: 0134638379, pp. 143-162.

Greg, R. (2003a) *Google Inconsistencies* [online], Search Engine Showdown, Available from: http://www.searchengineshowdown.com/features/google/inconsistent.shtml (Accessed October 2005).

Greg, R. (2003b) *Search Engine Statistics: Database Total Size Estimates* [online], Search Engine Showdown, Available from: http://searchengineshowdown.com/stats/sizeest.shtml (Accessed October 2005).

Greg, R. (2004a) *Review of Google* [online],Search Engine Showdown Jun.05.2004, Available from: http://searchengineshowdown.com/features/google/review.html (Accessed October 2005).

Greg, R. (2004b) *Search Engine Showdown Reviews* [online],Search Engine Showdown Apr.07.2004, Available from: http://searchengineshowdown.com/reviews/ (Accessed October 2005).

Greisdorf, H. and Spink, A. (2001) 'Median measure: an approach to IR systems evaluation', *Information Processing & Management*, 37(6), pp.843-857.

Haines, D. and Croft, W.B. (1993) 'Relevance Feedback and Inference Networks', in *Proceedings of SIGIR 93, Pittsburgh*, PA: ACM Press, pp. 2-11.

Hearst, M.A. (1995) 'Tile Bars: Visualisation of Term Distribution Information in Full Text Information Access', in *Proceedings of CHI 95*, Denver, Colorado, May 1995.

Henzinger, M., Motwani, R., Silverstein,C., and Brin, S. (2002) 'Challenges in Web Search Engines', in *Proceedings of the 11th International World Wide Web Conference*, September 3,2002, pp. 1-10.

Heydon, A. and Najork, M. (2001) *High Performance Web Crawling*, SRC Research Compaq Systems Research Centre (September 2001), Report 173.

Houben, G. J., De Bra, P., Kornatzky,Y., and Post, R. (1994) 'Information Retrieval in Distributed Hypertexts' in *Proceedings Of the RIAO 94: Intelligent Multimedia Retrieval Systems and Management*, New York, NY, October 1994.

Jansen, B.J., Spink, A. (2005). 'How are we searching the World Wide Web? A comparison of nine search engine transaction logs', Information Processing and Management, vol. 42, pp.248–263.

Jones, K.S. and Willet, P. (1997) *Readings in Information Retrieval: Morgan Kaufmann Series in Multimedia Information and Systems*, Morgan Kaufmann, ISBN: 1558604545, pp. 231-258.

Jong-Gyun L. (1994) 'Using Coollists to Index HTML Documents in the Web', *in Proceedings of the 2nd World Wide Web Conference*, Chicago, IL USA, October 1994.http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/lim/-coollist.html.

Jung, D. and Boutquin, P. (2000) *Visual Basic 6 super Bible*, Second Edition, ISBN:0-672-31413-4, pp. 105-196.

Kazai, G., Massod, S., Lalmas, M.(2004). 'A study of the assessment of relevance for the INEX'02 test collection (2004)', in Proceedings of 26th European Conference on IR Research, ECIR 2004, Sunderland, UK,Volume 2997/2004 , ISBN: 3540213821,pp. 296-310.

Kehoe, C., Pitkow, J., Sutton, K., Aggarwal, G., & Rogers, J. D. (1999) *Results of GVU's Tenth WWW User Survey* [Online], Georgia Institute of Technology Georgi-Tech Research, Available from: http://www.gvu.gatech.edu/user_surveys/survey-1998-10/tenthreport.html (Accessed October 2005).

Kowalski, G.J. (2000) *Information Storage and Retrieval Systems: Theory and Implementation (Kluwer International Series on Information Retrieval)*, Kluwer l Academic Publishers, Boston, ISBN: 0792379241, pp. 67-82.

Kules, B., Shneiderman, B., and Plaisant, C. (2003) 'Data Exploration with Paired Hierarchical Visualisations: Initial Designs of Pair-Trees', in *Proceedings of the 2003 National Conference on Digital Government Research*, Multimedia pioneer Nicholas Negroponte, co-founder of the MIT Media Lab, ,Boston, MA, USA.

LaMacchia, B.A. (1997) 'The Internet Fish Construction Kit' in *Proceedings of the 6th World Wide Web Conference*, Santa Clara, CA, April 1997, pp. 277-288.

Langville, A.N., Meyer, C.D. (2005). 'A Survey of Eigenvector Methods for Web Information Retrieval', SIAM REVIEW, Society for Industrial and Applied Mathematics, 47(1), pp.135–161.

Lawrence, S. and Giles, C. L. (1998) *Searching the World Wide Web*, (in reports). Science, 280(5360):98, April 3 1998.

Lee, D. L., Chuang, H., and Seamons, K. (1997) 'Document ranking and the vector-space model', *IEEE Software*, Vol.14 (2), pp. 67-75.

Leigh, W.H. (1998) *Library Systems: Current Developments and Future Directions*, Council on Library and Information Resources, Washington DC, pub72, ISBN 1-887334-58-0, pp. 34-48.

Leroy, G., Lally, A.M., and Chen, H. (2003) 'The Use of Dynamic Contexts to Improve Casual Internet Searching' ACM Transactions on Information Systems, 21(3), pp.229-253.

Lieberman, H. (1995) 'Letizia: An agent that assists web browsing', in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 924-929.

Liere,R. and Tadepalli, P. (1996), 'The Use of Active Learning in Text Categorization', *AAAI Spring Symposium on Machine Learning in Information Access*, Stanford, USA.

Liu, X., Brody,T., Harnad,S., Carr,L., Maly, K., Zubair, M., Nelson, M.L. (2002).'A Scalable Architecture for Harvest-Based Digital Libraries' , D-Lib Magazine, 8(11),ISSN 1082-9873.

Ljungberg, F. and Sørensen, C. (1998), 'Are you pulling the plug or pushing up the daisies?', in *Proceedings the Thirty-First Hawaii International Conference on System Sciences*, Hawaii, IEEE Computer Society Press.

Losee, R.M., and Church, L. (2004). 'Information Retrieval with Distributed Databases: Analytic Models of Performance', IEEE Transactions on Parallel and Distributed Systems, 15(1), pp.18-27.

Luhn, H.P.(1957) 'A statistical approach to mechanized encoding and searching of literary information.' *IBM Journal*, 1(4), pp. 309-317.

Lyman, P., Varian, H.R., Swearingen, K., and JPal, J. (2003), *How Much Information? 2003* [online], the School of Information Management and Systems at the University of California at Berkeley, Available from: http\\:www.sims.berkeley.edu/research/projects/how-much-info-2003/ (Accessed October 2005).

Michael K. (2003) *The Deep Web: Surfacing Hidden Value* [online], Bright Planet White Paper, University of Michigan Available from: http://www.brightplanet.com/technology/deepweb.asp (Accessed October 2005).

Minaji, M. and Vella, A. (1999) 'Using Fingerprint strategy for searching the Internet.', presented at the Operational Research seminar OR41, Edinburgh UK.

Modjeska, D. and Waterworth, J. (2000) 'Effects of Desktop 3D World Design on User Navigation and Search Performance', in *Proceedings of Information Visualisation IEEE 2000*, pp.215-220.

Mostafa, J. (2005). 'Seeking Better Web Searches', Scientific American, 292(2).

Motro,A. (1998) 'Vague: A user interface to relational databases that permits vague queries', *ACM Transactions on O#ce Information Systems*, 6(3), pp.187-214.

Naisbitt, J. (1999) *Megatrends 2000* , AVON Books, ISBN: 0-380-7-437-4, pp 32-54

Naughton, J. (1999) *A brief History of the Future*, Second Edition, phoenix/paperback, ISBN: 0-75381-093-X, pp. 124-157

Ng, A.Y., Zheng, A.X., Jordan, M.I. (2001). 'Stable algorithms for link analysis', in Proceedings of the 24th annual international ACM SIGIR conference on Research and development in Information Retrieval archive, New Orleans Louisiana USA, ISBN:1581133316, pp.258-266.

Oard, D. W. and Marchionini, G. (1996) *A Conceptual Framework for Text Filtering*, College Park, MD, University of Maryland, CAR-TR-830 CLIS-TR-96-02 CS-TR-3643 EE-TR-96-25.

O'Neill, E.T, Lavoie, B.F, and Bennett, R. (2003) 'Trends in the Evolution of the Public Web 1998 – 2002', *D-Lib Magazine*, 9 (4), ISSN 1082-9873.

Plaisant, C., Grosjean, J., and Bederson, B.B. (2002) 'Space Tree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation', in *Proceedings of the IEEE Symposium on Information Visualisation* (InfoVis'02), p.57-59.

Park, S.T., Pennock, D.M., Giles, C.L., Krovetz, R. (2002). 'Analysis of lexical signatures for finding lost or related documents', in Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, ISBN:1-58113-561-0, pp.11-18.

Rijsbergen, C. J. V. (1979) *Information Retrieval*, London, Butterworths, 2nd edition. pp. 26-47

Robertson, J., Merkus, E., and Ginige, A. (1994) 'The Hypermedia Authoring Research Toolkit (HART)', in *Proceedings of ECHT '94*, Edinburgh, UK. ACM Press. pp. 177-185.

Salton, G. and Buckley, C. (1990) 'Improving retrieval performance by relevance feedback', *Journal of the American Society for Information Science*, 41 (4), pp. 288-297.

Salton, G.(1989) *Automatic Text Processing*, New York, Addison-Wesley.

Schneider, A. (2004) 'Lightweight Knowledge Aggregation Using Semantic Web Technology' (doctoral thesis, University of Zurich, 2004), pp. 32–40.

Shapiro, C. and Varian, H. (1998) *Information Rules: A Strategic Guide to the Network Economy*, Harvard Business School Press, Cambridge, pp 4-18.

Stata, R., Bharat, K., and Maghoul, F. (2000) 'The Term Vector Database: fast access to indexing terms for Web pages, in *Proceeding 9th International World Wide Web Conference*, ISBN/1-930792-00-X$159.

Strzalkowski, T., Tzukermann, E., and Klavans, J. (2002) 'Information Retrieval and Natural Language Processing', *R. Mitkov (ed.), Handbook of Computational Linguistics*, Oxford University Press, 2002.

Sullivan, D. (2003) How Search Engines Rank Web Pages [online], Available from: http://searchenginewatch.com/webmasters/article.php/216761, (Accessed October 2005).

Tekla S.P. (2001), 'Service takes over in the networked world', *IEEE Spectrum*, 38(1), pp.102-106.

Turtle, H. (1994) 'Natural Language vs. Boolean query evaluation: A comparison of retrieval performance', in *Proceedings of SIGIR 94*, London: Springer Verlag, pp. 212--220.

Turtle, H. and Crof, W.B. (1992) 'Evaluation of an inference network-based retrival mode', *ACM Transactions on Information Systems*, 9(3), pp. 187-222.

Vechtomova, O. (2005). 'The Role of Multi-word Units in Interactive Information Retrieval.', ECIR, Springer-Verlag GmbH, Volume 3408 / 2005, ISSN: 0302-9743, pp. 29-38.

Wang, P., Berry, M. W., Yang, Y. (2003). 'Mining longitudinal Web queries: Trends and patterns', American Society of Information Science and Technology, vol. 54, pp.743–758.

Wheatley, H.B. (2002) *What is an Index: A Few Notes on Indexes and Indexers*, Society of Indexers, ISBN: 1871577233, pp. 22-31.

Wilks, Y. (1998), 'Language processing and the thesaurus', in *Proceedings National language Research Institute*, Tokyo, Japan, 1998.

Witten, I.H., Moffat, A., and Bell, T.C. (1994) *Managing Gigabytes: Compressing and Indexing Documents and Images*, Kluwer Academic Publishers, ISBN: 0442018630, pp. 208-226.

Yates, R.-B. and Neto, B.-R. (1999), *Modern Information Retrieval*, Second Edition, ISBN: 0-2010-39829-X, pp. 257-395

Ye, S., Lu, G., Li, X. (2004). 'workload-Aware Web Crawling and Server Workload Detection.', in Proceedings of the 2nd Asia-Pacific Advanced Network Research Workshop, pp.263-269.