PANIMALAR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF CSE

R-2017

ACADEMIC YEAR	:2023-2024
BATCH	:2020-2024
YEAR/SEM	:IV/VIII
SUBJECTCODE/TITLE	:CS8080/INFORMATION RETRIEVAL TECHNIQUES

UNIT-1

INTRODUCTION

UNIT I
INTRODUCTION

Information Retrieval – Early Developments – The IR Problem – The Users Task – Information versus Data Retrieval - The IR System – The Software Architecture of the IR System – The Retrieval and Ranking Processes - The Web – The e-Publishing Era – How the web changed Search – Practical Issues on the Web – How People Search – Search Interfaces Today – Visualization in Search Interfaces.

INFORMATION RETRIEVAL

Information retrieval (IR) is a broad area of Computer Science focused primarily on providing the users with easy access to information of their interest, as follows. Information retrieval deals with the representation, storage, organization of, and access to information items such as documents, Web pages, online catalogs, structured and semi-structured records, multimedia objects. The representation and organization of the information items should be such as to provide the users with easy access to information of their interest. Nowadays, research in IR includes modeling, Web search, text classification, systems architecture, user interfaces, data visualization, filtering, languages.

EARLY DEVELOPMENTS

For more than 5, 000 years, man has organized information f or later retrieval and searching. In its most usual form, this has been done by compiling, storing, organizing, and indexing clay tablets, hieroglyphics, papyrus rolls, and books. For holding the various items, special purpose buildings called libraries, from the Latin word liber for book, or bibliothekes, from the Greek word biblion for papyrus roll, are used. The oldest known library was created in Elba, in the "Fertile Crescent", currently northern Syria, some time between 3,000 and 2,500 BC. In the seventh century BC, Assyrian king Ashurbanipal created the library of Nineveh, on the Tigris River (today, north of Iraq), which contained more than 30,000 clay tablets at the time of its destruction in 612 BC. By 300 BC, Ptolemy Soter, a Macedonian general, created the Great Library in Alexandria – the

Egyptian city at the mouth of the Nile named after the Macedonian king Alexander the Great (356-323 BC). For seven centuries the Great Library, jointly with other major libraries in the city, made Alexandria the intellectual capital of the Western world.

Since then, libraries have expanded and flourished. Nowadays, they are everywhere. They constitute the collective memory of the human race and their popularity is in the rising. In 2008 alone, people in the US visited their libraries some 1.3 billion times and checked out more than 2 billion items – an increase in both yearly figures of more than 10 percent. Since the volume of information in libraries is always growing, it is necessary to build specialized data structures for fast search – the indexes. In one form or another, indexes are at the core of every modern information retrieval system. They provide fast access to the data and allow speeding up query processing.

For centuries indexes have been created manually as sets of categories. Each category in the index is typically composed of labels that identify its associated topics and of pointers to the documents that discuss those topics. While these indexes are usually designed by library and information science researchers, the advent of modern computers has allowed the construction of large indexes automatically, which has accelerated the development of the area of Information Retrieval (IR). Early developments in IR date back to research efforts conducted in the 50's by pioneers such as Hans Peter Luhn, Eugene Garfield, Philip Bagley, and Cal vi n Moores, this last one having allegedly coined the term information retrieval. In 1955, Allen Kent and colleagues published a paper describing the precision and recall metrics, which was followed by the publication in 1962 of the Cranfield studies by Cyril Cleverdon. In 1963, Joseph Becker and Robert Hayes published the first book on information retrieval [164]. Throughout the 60's, Gerard Salton and Karen Sparck Jones, among others, shaped the field by developing the fundamental concepts that led to the modern technologies of ranking in IR.In1968,thefirstIR book authored by Salton was published. In 1971, N.

Jardine and C.J.VanRijsbergen articulated the "cluster hypothesis". In 1978, the first ACM Conference on IR (ACM SIGIR) was held in Rochester, New York. In 1979, C.J. Van Rijsbergen published Information Retrieval, which focused on probabilistic models. In 1983, Salton and McGill published Introduction to Modern Information Retrieval, a classic book on IR focused on vector models. Since then, the IR community has grown to include thousands of professors, researchers, students, engineers, and practitioners throughout the world. The main conference in the area, the ACM International Conference on Information Retrieval (ACM SIGIR), now attracts hundreds of attendees and receives hundreds of submitted papers on an yearly basis.

THE IR PROBLEM

Users of modern IR systems, such as search engine users, have information needs of varying complexity. In the simplest case, they are looking for the link to the homepage of a company, government, or institution. In the more sophisticated cases, they are looking for information required to execute tasks associated with their jobs or immediate needs. An example of a more complex information need is as follows:

Find all documents that address the role of the Federal Government in financing the operation of the National Railroad Transportation Corporation (AMTRAK). This full description of the user need does not necessarily provide the best formulation for querying the IR system. Instead, the user might want to first translate this information need into a query, or sequence of queries, to be posed to the system. In its most common form, this translation yields a set of keywords, or index terms, which summarize the user information need. Given the user query, the key goal of the IR system is to retrieve information that is useful or relevant to the user. The emphasis is on the retrieval of information as opposed to the retrieval of data. To be effective in its attempt to satisfy the user information need, the IR system must somehow 'interpret' the contents of the information items i.e., the documents in a collection, and rank them according to a degree of relevance to the user query. This 'interpretation' of a document content involves extracting

syntactic and semantic information from the document text and using this information to match the user information need.

The IR Problem: the primary goal of an IR system is to retrieve all the documents that are relevant to a user query while retrieving as few non relevant documents as possible. The difficulty is knowing not only how to extract information from the documents but also knowing how to use it to decide relevance. That is, the notion of relevance is of central importance in IR. One main issue is that relevance is a personal assessment that depends on the task being solved and its context. For example, relevance can change with time (e.g., new information becomes available), with location (e.g., the most relevant answer is the closest one), or even with the device (e.g., the best answer is a short document that is easier to download and visualize). In this sense, no IR system can provide perfect answers to all users all the time.

THE USERS TASK

The user of a retrieval system has to translate their information need into a query in the language provided by the system. With an IR system, such as a search engine, this usually implies specifying a set of words that convey the semantics of the information need. We say that the user is searching or querying for information of their interest. While searching for information of interest is the main retrieval task on the Web, search can also be used for satisfying other user needs distinct from information access, such as the buying of goods and the placing of reservations. Consider now a user who has an interest that is either poorly defined or inherently broad, such that the query to specify is unclear. To illustrate, the user might be interested in documents about car racing in general and might decide to glance related documents about Formula 1 racing, Formula Indy, and the ‘24 Hours of Le Mans. We say that the user i s browsing or navigating the documents in the collection, not searching. It is still a process of retrieving information, but one whose main objectives are less clearly defined in the beginning. The task in this case is more related to exploratory search and resembles a process of quasi-

sequential search for information of interest. In this book, we make a clear distinction between the different tasks the user of the retrieval system might be engaged in. The task might be then of two distinct types: searching and browsing,asillustratedinFigure1.1.

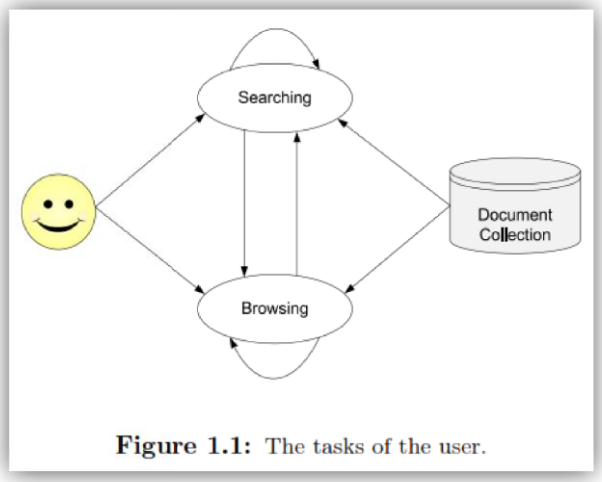

Figure 1.1: The tasks of the user.

INFORMATION VERSUS DATA RETRIEVAL

Data retrieval, in the context of an IR system, consists mainly of determining which documents of a collection contain the keywords in the user query which, most frequently, is not enough to satisfy the user information need. In fact, the user of an IR system is concerned more with retrieving information about a subject than with retrieving data that satisfies a given query. For instance, a user of an IR system is willing to accept documents that contain synonyms of the query terms in the result set, even when those documents do not contain any query terms.

That is, in an IR system the retrieved objects might be inaccurate and small errors are likely to go unnoticed. In a data retrieval system, on the contrary, a single erroneous object among a thousand retrieved objects means total failure. A data retrieval system, such as a relational database, deals with data that has a well defined structure and sem ntics, while an IR system deals with natural language text which is not well structured. Data retrieval, while providing a solution to the user of a database system, does not solve the problem of retrieving information about a subject or topic.

Sn	Data	Information
1	unorganized raw facts that need processing without which it is seemingly random and useless to humans	Information is a processed, organized data presented in a given context and is useful to humans.
2	Data is an individual unit that contains raw material which does not carry any specific meaning	Information is a group of data that collectively carry a logical meaning
3	Data doesn't depend on information.	Information depends on data.
4	It is measured in bits and bytes.	Information is measured in meaningful units like time, quantity, etc.
5	Data is never suited to the specific needs of a designer	Information is specific to the expectations and requirements because all the irrelevant facts and figures are removed, during the transformation process.
6	An example of data is a student's test score	The average score of a class is the information derived from the given data.

THE IR SYSTEM

In this section we provide a high level view of the software architecture of an IR system. We also introduce the processes of retrieval and ranking of documents in response to a user query.

THE SOFTWARE ARCHITECTURE OF THE IR SYSTEM

To describe the IR system, we use a simple and generic software architecture as shown in Figure 1.2.

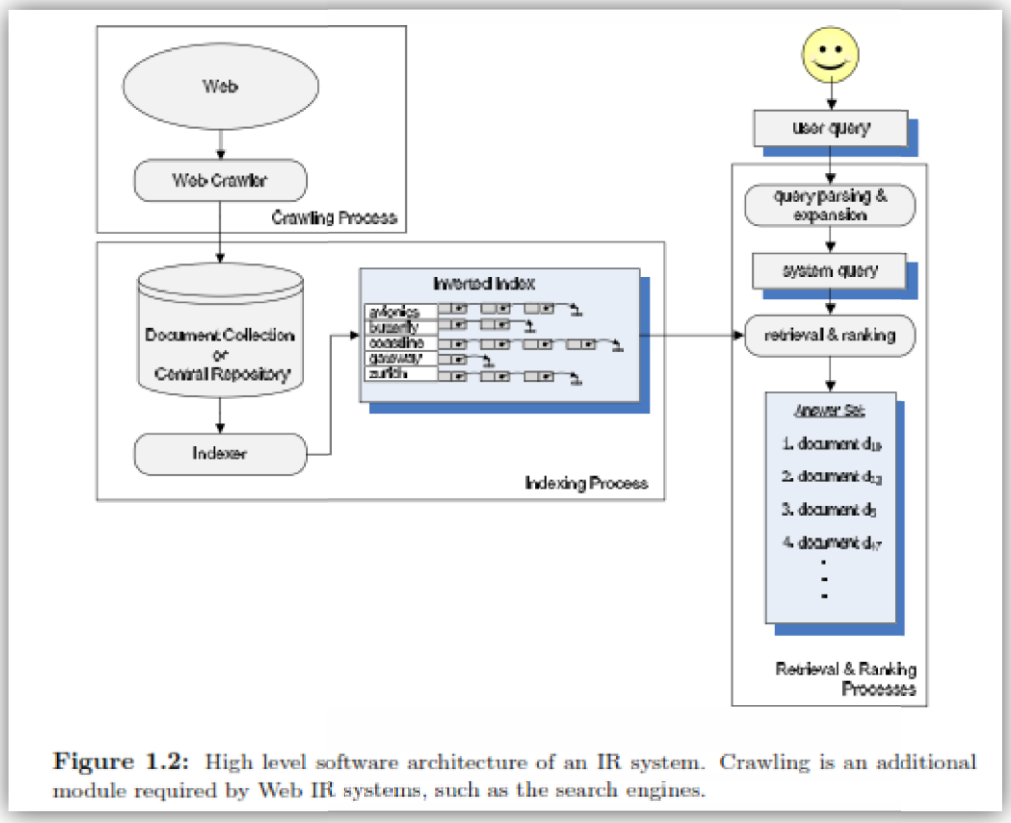

Figure 1.2: High level software architecture of an IR system. Crawling is an additional module required by Web IR systems, such as the search engines.

The first step in setting up an IR system is to assemble the document collection, which can be private or be crawled from the Web. In the second case a crawler module is responsible for collecting the documents. The document collection is stored in disk storage usually referred to as the central repository. The documents in the central repository need to be indexed for fast retrieval and ranking. The most used index structure is an inverted index composed of all the distinct words of the collection and, for each word, a list of the documents that contain it.

Given that the document collection is indexed, the retrieval process can be initiated. It consists of retrieving documents that satisfy either a user query or a click in a hyper link. In the first case, we say that the user is searching for information of interest; in the second case, we say that the user is browsing for information of interest. In the remaining of this section, we use retrieval as it applies to the searching process. For a more detailed discussion on browsing and how it compares to searching. To search, the user first specifies a query that reflects

their information need. Next, the user query is parsed and expanded with, for instance, spelling variants of a query word. The expanded query, which we refer to as the system query, is then processed against the index to retrieve a subset of all documents. Following, the retrieved documents are ranked and the top documents are returned to the user.

The purpose of ranking is to identify the documents that are most likely to be considered relevant by the user, and constitutes the most critical part of the IR system. Given the inherent subjectivity in deciding relevance, evaluating the quality of the answer set is a key step for improving the IR system. A systematic evaluation process allows fine tuning the ranking algorithm and improving the quality of the results. The most common evaluation procedure consists of comparing the set of results produced by the IR system with results suggested by human specialists.

To improve the ranking, we might collect feedback from the users and use this information to change the results. In the Web, the most abundant form of user feedback are the clicks on the documents in the results set. Another important source of information for Web ranking are the hyperlinks among pages, which allow identifying sites of high authority. There are many other concepts and technologies that bear impact on the design of a full fledged IR system, such as a modern search engine.

THE RETRIEVAL AND RANKING PROCESSES

To describe the retrieval and ranking processes, we further elaborate on our description of the modules shown in Figure 1.2, as illustrated in Figure 1.3. Given the documents of the collection, we first apply text operations to them such as eliminating stop words, stemming, and selecting a subset of all terms for use as indexing terms. The indexing terms are then used to compose document representations, which might be smaller than the documents themselves (depending on the subset of index terms selected). Given the document representations, it is

necessary to build an index of the text. Different index structures might be used, but the most popular one is an inverted index. The steps required to generate the index compose the indexing process and must be executed offline, before the system is ready to process any queries.

The resources (time and storage space) spent on the indexing process are amortized by querying the retrieval system many times. Given that the document collection is indexed, the retrieval process can be initiated. The user first specifies a query that reflects their information need. This query is then parsed and modified by operations that resemble those applied to the documents. Typical operations at this point consist of spelling corrections and elimination of terms such as stop words, whenever appropriated. Next, the transformed query is expanded and modified. For instance, the query might be modified using query suggestions made by the system and confirmed by the user.

The expanded and modified query is then processed to obtain the set of retrieved documents, which is composed of documents that contain the query terms. Fast query processing is made possible by the index structure previously built. The steps required to produce the set of retrieved documents constitute the retrieval process. Next, the retrieved documents are ranked according to a likelihood of relevance to the user. This is a most critical step because the quality of the results, as perceived by the users, is fundamentally dependent on the ranking.

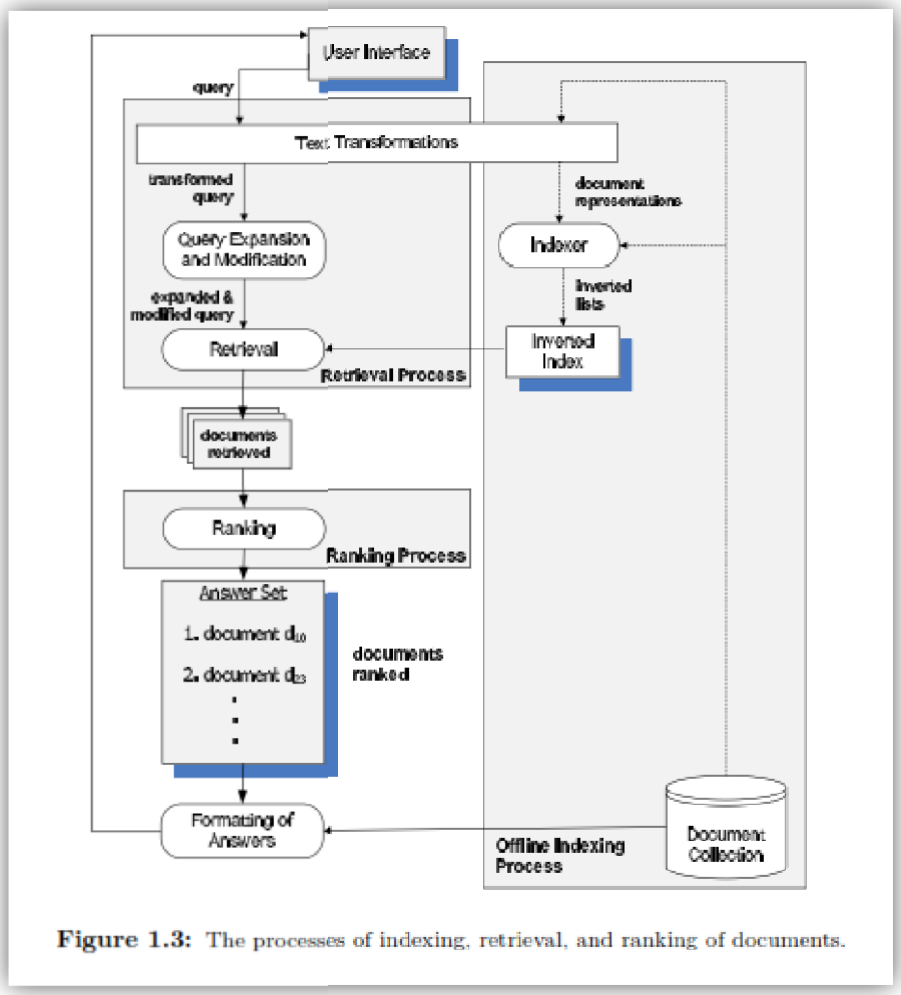

Figure 1.3: The processes of indexing, retrieval, and ranking of documents.

the ranking process in great detail. The top ranked documents are then formatted for presentation to the user. The formatting consists of retrieving the title of the documents and generating snippets for them, i.e., text excerpts that contain the query terms, which are then displayed to the user.

THE WEB

In this section we discuss the creation of the Web and its major implication–the advent of the e-publishing age. We also discuss how the Web changed search, i.e., he major impacts of the Web on the search task. At the end, we cover practical issues, such as security and copyright, which derive directly from the massive presence of millions of users on the Web.

A Brief History

"As We May Think" influenced people like Douglas Engelbart who, at the Fall Joint Computer Conference in San Francisco in December of 1968, ran a demonstration in which he introduced the first ever computer mouse, video conferencing, teleconferencing, and hypertext. It was so incredible that it became known as "the mother of all demos" [1690]. Of the innovations displayed, the one that interests us the most here is hypertext. The term was coined by Ted Nelson in his Project Xanadu.

Hypertext allows the reader to jump from one electronic document to another, which was one important property regarding the problem Tim Berners-Lee faced in 1989. At the time, Berners-Lee worked in Geneva at the CERN – Conseil Europeen pourla Recherche Nucleaire. There, researchers who wanted to share documentation with others had to reformat their documents to make them compatible with an internal publishing system. It was annoying and generated many questions, many of which ended up been directed towards Berners-Lee. He understood that a better solution was required.

It just so happened that CERN was the largest Internet node in Europe. Berners Lee reasoned that it would be nice if the solution to the problem of sharing documents were decentralized, such that the researchers could share their contributions freely. He saw that a networked hypertext, through the Internet, would be a good solution and started working on its implementation. In 1990, he wrote the HTTP protocol, defined the HTML language, wrote the first browser, which he called "World Wide Web", and the first Web server. In 1991, he made hi s browser and server software available in the Internet. The Web was born.

THE E-PUBLISHING ERA

Since its inception, the Web became a huge success. The number of Web pages now far exceeds 20 billion and the number of Web users in the world exceeds 1.7 billion. Further, it is known that there are more than one trillion distinct URLs on the Web, even if many of them are pointers to dynamic pages, not static

HTML pages. Further, a viable model of economic sustainability based on online advertising was developed.

The advent of the Web changed the world in a way that few people could have anticipated. Yet, one has to wonder on the characteristics of the Web that have made it so successful. Is there a single characteristic of the Web that was most decisive for its success? Tentative answers to this question include the simple HTML markup language, the low access costs, the wide spread reach of the Internet, the interactive browser interface, the search engines. However, while providing the fundamental infrastructure for the Web, these technologies were not the root cause of its popularity. What was it then? The fundamental shift in human relationships, introduced by the Web, was freedom to publish. Jane Austen did not have that freedom, so she had to either convince a publisher of the quality of her work or pay for the publication of an edition of it herself. Since she could not pay for it, she had to be patient and wait for the publisher to become convinced. It took 15 years.

In the world of the Web, this is no longer the case. People can now publish their ideas on the Web and reach millions of people over night, without paying anything for it and without having to convince the editorial board of a large publishing company. That is, restrictions imposed by mass communication media companies and by natural geographical barriers were almost entirely removed by the invention of the Web, which has led to a freedom to publish that marks the birth of a new era. One which we refer to as The e-Publishing Era.

HOW THE WEB CHANGED SEARCH

Web search is today the most prominent application of IR and its techniques. Indeed, the ranking and indexing components of any search engine are fundamentally IR pieces of technology. An immediate consequence is that the Web has had a major impact in the development of IR, as we now discuss. The first major impact of the Web on search is related to the characteristics of the document

collection itself. The Web collection is composed of documents (or pages) distributed over millions of sites and connected through hyperlinks , i.e., links that associate a piece of text of a page with other Web pages. The inherent distributed nature of the Web collection requires collecting all documents and storing copies of them in a central repository, prior to indexing. This new phase in the IR process, introduced by the Web, is called.

The second major impact of the Web on search is related to the size of the collection and the volume of user queries submitted on a daily basis. Given that the Web grew larger and faster than any previous known text collection, the search engines have now to handle a volume of text that far exceeds 20 billion pages, i.e., a volume of text much larger than any previous text collection. Further, the volume of user queries is also much larger than ever before, even if estimates vary widely. The combination of a very large text collection with a very high query traffic has pushed the performance and scalability of search engines to limits that largely exceed those of any previous IR system.

That is, performance and scalability have become critical characteristics of the IR system, much more than they used to be prior to the Web. While we do not discuss performance and scalability of search engines. The third major impact of the Web on search is also related to the vast size of the document collection. In a very large collection, predicting relevance is much harder than before. Basically, any query retrieves a large number of documents that match its terms, which means that there are many noisy documents in the set of retrieved documents. That is, documents that seem related to the query but are actually not relevant to it according to the judgement of a large fraction of the users are retrieved.

This problem first showed up in the early Web search engines and became more severe as the Web grew. Fortunately, the Web also includes new sources of evidence not present in standard document collections that can be used to alleviate the problem, such as hyperlinks and user clicks in documents in the answer set.

Two other major impacts of the Web on search derive from the fact that the Web is not just a repository of documents and data, but also a medium to do business. One immediate implication is that the search problem has been extended beyond the seeking of text information to also encompass other user needs such as the price of a book, the phone number of a hotel, the link for downloading a software. Providing effective answers to these types of information needs frequently requires identifying structured data associated with the object of interest such as price, location, or descriptions of some of its key characteristics.

The fifth and final impact of the Web on search derives from Web advertising and other economic incentives. The continued success of the Web as an interactive media for the masses created incentives for its economic exploration in the form of, for instance, advertising and electronic commerce. These incentives led also to the abusive availability of commercial information disguised in the form of purely informational content, which is usually referred to as Web spam.

The increasingly pervasive presence of spam on the Web has made the quest for relevance even more difficult than before, i.e., spam content is sometimes so compelling that it is confused with truly relevant content. Because of that, it is not unreasonable to think that spam makes relevance negative, i.e., the presence of spam makes the current ranking algorithms produce answers sets that are worst than they would be if the Web were spam free. This difficulty is so large that today we talk of Adversarial Web Retrieval.

PRACTICAL ISSUES ON THE WEB

Electronic commerce is a major trend on the Web nowadays and one which has benefited millions of people. In an electronic transaction, the buyer usually submits to the vendor credit information to be used for charging purposes. In its most common form, such information consists of a credit card number. For security reasons, this information is usually encrypted, as done by institutions and companies that deploy automatic authentication processes.

Besides security, another issue of major interest is privacy. Frequently, people are willing to exchange information as long as it does not become public. The reasons are many, but the most common one is to protect oneself against misuse of private information by third parties. Thus, privacy is another issue which affects the deployment of the Web and which has not been properly addressed yet.

Two other important issues are copyright and patent rights. It is far from clear how the wide spread of data on the Web affects copyright and patent laws in the various countries. This is important because it affects the business of building up and deploying large digital libraries. For instance, is a site which supervises all the information it posts acting as a publisher? And if so, is it responsible for misuse of the information it posts (even if it is not the source)? Additionally, other practical issues of interest include scanning, optical character recognition (OCR), and cross-language retrieval (in which the query is in one language but the documents retrieved are in another language).

HOW PEOPLE SEARCH

Search tasks range from the relatively simple (e.g., looking up disputed facts or finding weather information) to the rich and complex (e.g., job seeking and planning vacations). Search interfaces should support a range of tasks, while taking into account how people think about searching for information. This section summarizes theoretical models about and empirical observations of the process of online information seeking.

Information Lookup versus Exploratory Search

User interaction with search interfaces differs depending on the type of task, the amount of time and effort available to invest in the process, and the domain expertise of the information seeker. The simple interaction dialogue used in Web search engines is most appropriate for finding answers to questions or to finding Web sites or other resources that act as search starting points. But, as Marchionini

notes, the "turn-taking" interface of Web search engines is inherently limited and in many cases is being supplanted by speciality search engines – such as for travel and health information – that offer richer interaction models.

Classic versus Dynamic Model of Information Seeking

Researchers have developed numerous theoretical models of how people go about doing search tasks. The classic notion of the information seeking process model as described by Sutcliffe and Ennis is formulated as a cycle consisting of four main activities:

- problem identification,
- articulation of information need(s),
- query formulation, and
- results evaluation.

The standard model of the information seeking process contains an underlying assumption that the user's information need is static and the information seeking process is one of successively refining a query until all and only those documents relevant to the original information need have been retrieved. More recent models emphasize the dynamic nature of the search process, noting that users learn as they search, and their information needs adjust as they see retrieval results and other document surrogates. This dynamic process is sometimes referred to as the berry picking model of search.

SEARCH INTERFACES TODAY

At the heart of the typical search session is a cycle of query specification, inspection of retrieval results, and query reformulation. As the process proceeds, the searcher learns about their topic, as well as about the available information sources. This section describes several user interface components that have become standard in search interfaces and which exhibit high usability. As these components are described, the design characteristics that they support will be underscored. Ideally, these components are integrated together to support the different parts of the process, but it is useful to discuss each separately.

Query Specification

Once a search starting point has been selected, the primary methods for a searcher to express their information need are either entering words into a search entry form or selecting links from a directory or other information organization display. For Web search engines, the query is specified in textual form. Today this is usually done by typing text on a keyboard, but in future, query specification via spoken commands will most likely become increasingly common, using mobile devices as the input medium.

Typically in Web queries today, the text is very short, consisting of one to three words. Multiword queries are often meant to be construed as a phrase, but can also consist of multiple topics. Short queries reflect the standard usage scenario in which the user "tests the waters" to see what the search engine returns in response to their short query. If the results do not look relevant, then the user reformulates their query. If the results are promising, then the user navigates to the most relevant- looking Web site and pursues more fine-tuned queries on that site.

This search behavior, in which a general query is used to find a promising part of the information space, and then follow hyperlinks within relevant Web sites, is a demonstration of the orienteering strategy of Web search. There is evidence that in many cases searchers would prefer to state their information need in more detail, but past experience with search engines taught them that this method does not work well, and that keyword querying combined with orienteering works better.

Query Specification Interfaces

The standard interface for a textual query is a search box entry form, in which the user types a query, activated by hitting the return key on the keyboard or selecting a button associated with the form. Studies suggest a relationship between query length and the width of the entry form; results find that either small forms

discourage long queries or wide forms encourage longer queries. Some entry forms are divided into multiple components, allowing for a more general free text query followed by a form that filters the query in some way. For instance, at yelp.com, the user enters a general query into the first entry form and refines the search by location in the second form (see Figure 2.1).

Figure 2.1: Query form, from yelp.com, illustrating support to facilitate structured queries, and stored information about past queries.

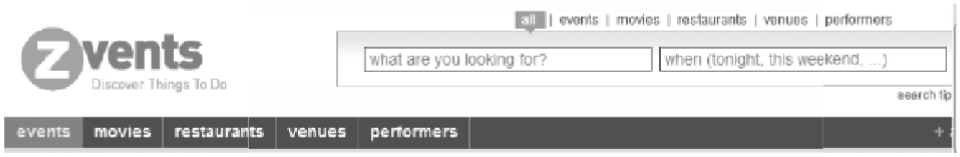

Figure 2.2: Query form, from zvents.com, illustrating greyed-out text that provides hints about what kind of information to type, directly in the form.

Forms allow for selecting information that has been used in the past; sometimes this information is structured and allows for setting parameters to be used in future. For instance, the yelp.com form shows the user's home location (if it has been indicated in the past) along with recently specified locations and the option to add additional locations.

An increasingly common strategy within the search form is to show hints about what kind of information should be entered into each form via greyed-out text. For instance, in zvents.com search (see Figure 2.2), the first box is labeled "what are you looking for?" while the second box is labeled "when (tonight, this weekend, ...)". When the user places the cursor into the entry form, the grey text disappears, and the user can type in their query terms. This example also illustrates

specialized input types that some search engines are supporting today. For instance, the zvents.com site recognizes that words like “tomorrow” are time-sensitive, and interprets them in the expected manner. It also allows flexibility in the syntax of more formal specification of dates. So searching for “comedy” on “wed” automatically computes the date for the nearest future Wednesday.

This is an example of designing the interface to reflect how people think, rather than making how the user thinks conform to the brittle, literal expectations of typical programs. (This approach to “loose” query specifi cation works better for “casual” interfaces in which getting the date right is not critical to the use of the system; casual date specification when filling out a tax form is not acceptable, as the cost of error is too high.)

An innovation that has greatly improved query specification is the inclusion of a dynamically generated list of query suggestions, shown in real time as the user types the query. This is referred to variously as auto-complete, auto-suggest, and dynamic query suggestions. A large log study found that users clicked on dynamic suggestions in the Yahoo Search Assist tool about one third of the time they were presented . This topic is covered in detail for the case of Web search engines. Often the suggestions shown are those whose prefix matches the characters typed so far, but in some cases, suggestions are shown that only have interior letters matching. If the user types a multiple word query, suggestions may be shown that are synonyms of what has been typed so far, but which do not contain lexical matches.

To exemplify, Netflix.com both describes what is wanted in gray and then shows hits via a dropdown box. In dynamic query suggestion interfaces, the display of the matches varies. Some interfaces color the suggestions according to category information. In most cases, the user must move the mouse down to the desired suggestion in order to select it, at which point the suggestion is used to fill the query box. In some cases, the query is then run immediately; in others, the user must hit the Return key or click the Search button. The suggestions can be derived from several sources. In some cases, the list is taken from the user’s own query

history, in other cases, it is based on popular queries issued by other users. The list can be derived from a set of metadata that a Web site's designer considers important, such as a list of known diseases or gene names for a search over pharmacological literature (see Figure 2.3), a list of product names when searching within an e-commerce site, or a list of known film names when searching a movie site.

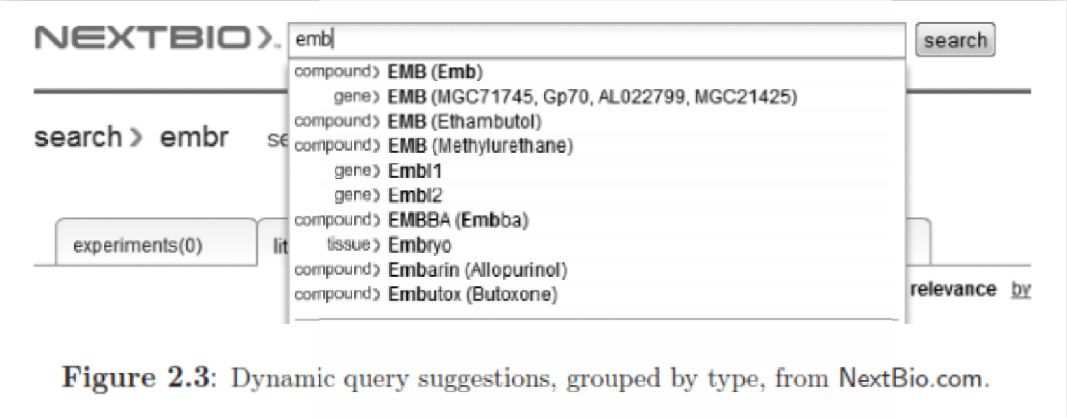

Figure 2.3: Dynamic query suggestions, grouped by type, from NextBio.com.

The suggestions can also be derived from all of the text contained within a Web site. Another form of query specification consists of choosing from a display of information, typically in the form of hyperlinks or saved bookmarks. In some cases, the action of selecting a link produces more links for further navigation, in addition to results listings. This kind of query specification is discussed in more detail in the section on organizing search results below.

Query Reformulation

After a query is specified and results have been produced, a number of tools exist to help the user reformulate their query, or take their information seeking process in a new direction. Analysis of search engine logs shows that reformulation is a common activity; one study found that more than 50% of searchers modified at least one query during a session, with nearly a third of these involving three or more qu ries.

Organizing Search Results

Searchers often express a desire for a user interface that organizes search results into meaningful groups to help understand the results and decide what to do next. A longitudinal study in which users were provided with the ability to group search results found that users changed their search habits in response to having the grouping mechanism available. Currently two methods for grouping search results are popular: category systems, especially faceted categories, and clustering. Both are described in more detail in this section, and their usability is compared.

A category system is a set of meaningful labels organized in such a way as to reflect the concepts relevant to a domain. They are usually created manually, although assignment of documents to categories can be automated to a certain degree of accuracy. Good category systems have the characteristics of being coherent and (relatively) complete, and their structure is predictable and consistent across search results for an information collection.

VISUALIZATION IN SEARCH INTERFACES.

Text as a representation is highly effective for conveying abstract information, but reading and even scanning text is a cognitively taxing activity, and must be done in a linear fashion. By contrast, images can be scanned quickly and the visual system perceives information in parallel. People are highly attuned to images and visual information, and pictures and graphics can be captivating and appealing. A visual representation can communicate some kinds of information much more rapidly and effectively than any other method. Consider the difference between a written description of a person's face and a photograph of it, or the difference between a table of numbers containing a correlation and a scatter plot showing the same information.

Experimentation with visualization for search has been primarily applied in the following ways:

- Visualizing Boolean Syntax,

- Visualizing Query Terms within Retrieval Results,
- Visualizing Relationships among Words and Documents,
- Visualization for Text Mining.

Visualizing Boolean Syntax

As noted above, Boolean query syntax is difficult for most users and is rarely used in Web search. For many years, researchers have experimented with how to visualize Boolean query specification, in order to make it more understandable. A common approach is to show Venn diagrams visually; Hertzum and Frokjaer [755] found that a simple Venn diagram representation produced more accurate results than Boolean syntax. A more flexible version of this idea was seen in the VQuery system (see Figure 2.15). Each query term is represented by a circle or oval, and the intersection among circles indicates ANDing (conjoining) of terms. VQuery represented disjunction by sets of circles within an active area of the canvas, and negation by deselecting a circle with in the active area.

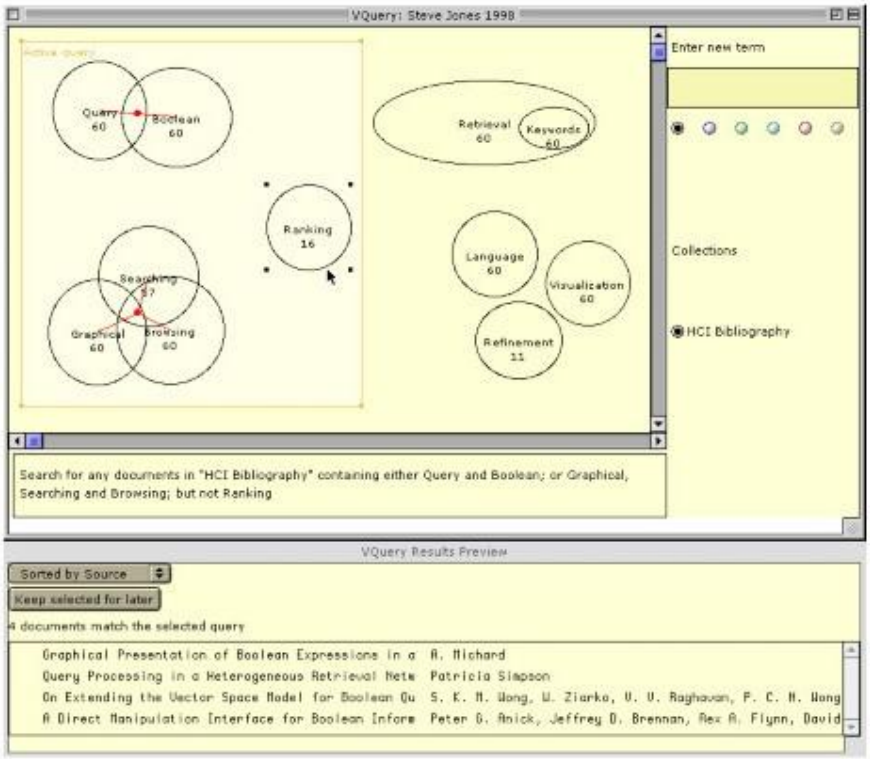

Figure 2.15: The VQuery [851] Venn Diagram interface for Boolean query specification.

One problem with Boolean queries is that they can easily end up with empty results or too many results. To remedy this, the filter-flow visualization allows users to lay out the different components of the query, and show via a graphical flow how many hits would result after each operator is applied . Other visual representations of Boolean queries include lining up blocks vertically and horizontally and representing components of queries as overlapping "magic" lenses.

Visualizing Query Terms within Retrieval Results

As discussed above, understanding the role played by the query terms within the retrieved documents can help with the assessment of relevance. In standard search results listings, summary sentences are often selected that contain query terms, and the occurrence of these terms are highlighted or boldfaced where they appear in the title, summary, and URL. Highlighting of this kind has been shown to be effective from a usability perspective.

Experimental visualizations have been designed that make this relationship more explicit. One of the best known is the TileBars interface, in which documents are shown as horizontal glyphs with the locations of the query term hits marked along the glyph (see Figure 2.16).

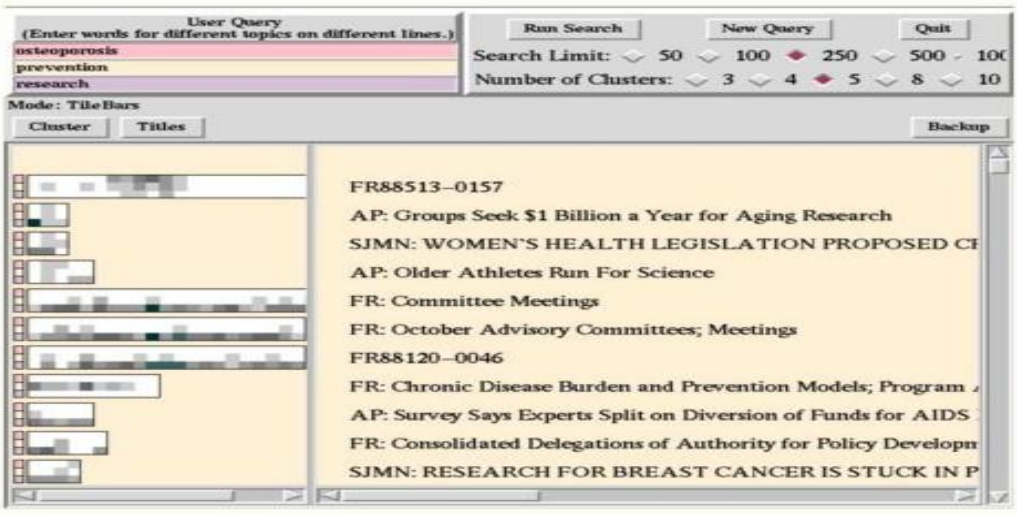

Figure 2.16: The TileBars visualization of query term hits within retrieved documents, from [732].

The user is encouraged to break the query into its different facets, with one concept per line, and then the horizontal rows within each document's

representation show the frequency of occurrence of query terms within each topic. Longer documents are divided into subtopic segments, either using paragraph or section breaks, or an automated discourse segmentation technique called TextTiling. Grayscale implies the frequency of the query term occurrences. The visualization shows where the discussion of the different query topics overlaps within the document.

Visualizing Relationships Among Words and Documents

Numerous visualization developers have proposed variations on the idea of placing words and documents on a two-dimensional canvas, where proximity of glyphs represents semantic relationships among the terms or documents. An early version of this idea is seen in the VIBE interface, where queries are laidout on a plane, and documents that contain combinations of the queries are placed midway between the icons representing those terms (see Figure 2.19). A more modern version of this idea is seen in the Aduna Autofocus product, and the Lyberworld project presented a 3D version of the ideas behind VIBE.

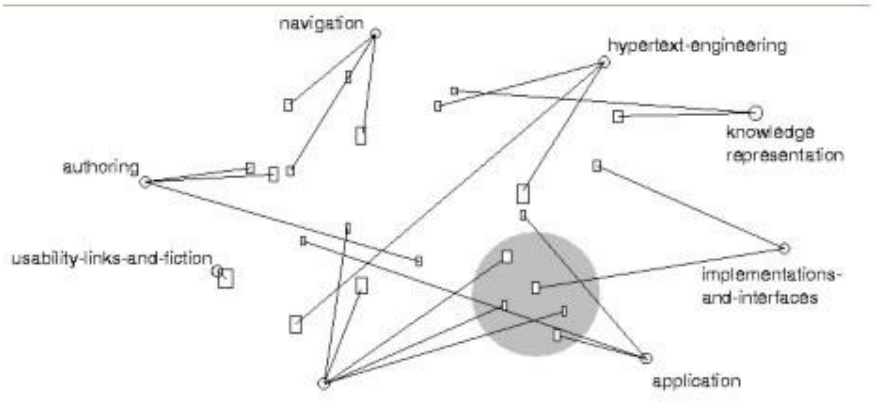

Figure 2.19: The VIBE display, in which query terms are laid out in a 2D space, and documents are arranged according to which subset of the text they share, from [1228].

Another variation of this idea is to map documents or words from a very high dimensional term space down into a two-dimensional plane, and show where the documents or words fall within that plane, using 2D or 3D. This variation on clustering can be done to documents retrieved as a result of a query, or documents that match a query can be highlighted within a preprocessed set of documents.

Visualization for Text Mining

The subsections above show that usability results for visualization in search are not particularly strong. It seems in fact that visualization is better used for purposes of analysis and exploration of textual data. Most users of search systems are not interested in seeing how words are distributed across documents, or in viewing the most common words within a collection, but these are interesting activities for computational linguists, analysts, and curious word enthusiasts. Visualizations such as the Word Tree show a piece of a text concordance, allowing the user to view which words and phrases commonly precede or follow a given word (see Figure 2.22), or the Name Voyager explorer, which shows frequencies of baby names for U.S. children across time (see Figure 2.23 on page 51).

Figure 2.22: The Word Tree visualization, on Martin Luther King's *I have a dream* speech, from The word tree, an interactive visual concordance, *IEEE Transactions on Visualization and Computer Graphics*, 14(6), pp. 1221-8 (Wattenberg, M. and Fernanda, B., 2008), ©2008 IEEE [1669].

Figure 2.23: A visualization of the relative popularity of baby names over time, with names beginning with the letters JA, from babynamewizard.com.

PART-A

1 Define information retrieval
2 Identify the need of Information Retrieval
3 List and explain the components of IR block diagram.
4 List the fundamental concepts in IR
5 Express the need of tiered indexes
6 Interpret the role of Artificial Intelligence (AI) in IR
7 Differentiate data retrieval and information retrieval
8 Give the components of Search Engine and the performance measures.
9 What is an extractor?
10 Show the issues that affects IR
11 Give the purpose of Query Interface
12 Summarize the queries of IR
13 Design the IR architecture diagram
14 State the impact of WEB on IR
15 Show the type of natural language technology used in information retrieval.
16 Compare Information vs Data Retrieval
17 What is search engine?
18 Compare IR vs Web Search
19 Construct the function of Information Retrieval System
20 Summarize on text acquisition.

PART-B

1 i) Summarize the history of IR. (7)
ii) Explain the purpose of Information Retrieval System.(6)
2 Describe the various components of Information Retrieval System with neat diagram. (13)
3 i) Define Information Retrieval system and its features. (4)
ii) Formulate the working of Search Engine. (9)
4. i) Identify the various issues in IR system. (7)
ii) Examine the various impact of WEB on IR (6)
5 Demonstrate the framework of Open Source Search engine with necessary diagrams. (13)
6 i) Compare in detail Information Retrieval and Web Search with examples. (8)
ii) Analyze the fundamental concepts involved in IR system. (5)
BTL 4 Analyze
7 Develop the role of Artificial Intelligence in Information Retrieval Systems. (13)
8 i) Describe the various components of a Search Engine. (8)
ii) Summarize the functions and features of Information Retrieval Systems (5)
9 i) Describe the different stages of IR system. (8)
ii) Estimate the various Search Engine available in current world. (5)
10 i) Demonstrate the working of IR architecture with a diagram. (6)
ii) Infer How Designing Parsing and Scoring functions works in detail.(7)
11 i)Define Information Retrieval. (2)
ii) Describe in detail the IR system, Fundamental concepts, need and purpose of the system.(4+4+3)

12 Explain how to characterize the web in detail. (13) BTL 4 Analyze
13 Explain the different types of computer software used in computer architecture.(13)
14 i) Demonstrate database and Information Retrieval with example (4)
ii) Generalize the Process of Search Engine in detail.(9)

PART-C

1 Create an open source search engine like Google with suitable functionalities. (15)
2 Evaluate the best search engines other than Google and explain any five of them in detail. (15)
3 Justify how the AI impact Search and Search Engine optimization. (15)
4 Generalize the Deep Learning and Human Learning capabilities in Future of Search Engine Optimization. (15)