Information Retrieval

Information retrieval systems are everywhere: Web search engines, library catalogs, store catalogs, cookbook indexes, and so on. *Information retrieval* (IR), also called *information storage and retrieval* (ISR or ISAR) or *information organization and retrieval*, is the art and science of retrieving from a collection of items a subset that serves the user's purpose; for example:

- Web pages useful in preparing for a trip to Europe;
- magazine articles for an assignment or good reading for that trip to Europe;
- educational materials for a learning objective;
- digital cameras for taking family photos;
- recipes that use ingredients on hand;
- facts needed for deciding on a company merger.

The main trick is to retrieve what is useful while leaving behind what is not.

The Scope of IR

IR systems are part of a family that shares many principles (Figure 1).

	Finding answers and information that already exist in a system		Creating answers and new information by analysis and inference – based on query
	Search by navigation (following links, as in a subject directory and the Web generally)	Search by query (as in Google)	
Unstructured information (text, images, sound)	**Hypermedia systems** (Many small units, such as paragraphs and single images, tied together by links)	**IR systems** (Often dealing with whole documents, such as books and journal articles)	
Structured information		**Database management systems (DBMS)**	**Data analysis systems Expert systems**

Figure 1. The IR system family

Two distinctions are of particular importance:

(1) A system for *unstructured information* deals with questions like The economic impact of the Reformation, The pros and cons of school uniforms, or Find a nice picture of my niece. It finds documents that are more or less useful; the user must then extract the data needed. In contrast, a system for *well-structured information* deals with precise questions and returns precise answers, exactly the small pieces of data needed: the salary of Mrs. Smith; the population of China; the winner of the 1997 World Series.

(2) *Finding versus creating answers*. IR and database systems merely find what is already there: for example, from a patient database, a patient's symptoms; from a disease database, the diseases these symptoms point to (or a medical textbook from which to extract this information); and from a drug database, the drugs that treat a disease. A physician must then absorb all this information, derive a diagnosis, and prescribe a drug. A medical expert system goes beyond just finding the facts – it creates new information by inference: it identifies a disease that explains the patient's symptoms and then finds a drug for the disease.

The Objects of IR

Traditionally, IR has concentrated on finding whole documents consisting of written text; much IR research focuses more specifically on *text retrieval* – the computerized retrieval of machine-readable text without human indexing. But there are many other interesting areas:

- *Speech retrieval*, which deals with speech, often transcribed manually or (with errors) by automated speech recognition (ASR).
- *Cross-language retrieval*, which uses a query in one language (say English) and finds documents in other languages (say Chinese and Russian).
- *Question-answering* IR systems, which retrieve answers from a body of text. For example, the question Who won the 1997 World Series*?* finds a 1997 headline World Series: Marlins are champions.
- *Image retrieval*, which finds images on a theme or images that contain a given shape or color.
- *Music retrieval*, which finds a piece when the user hums a melody or enters the notes of a musical theme.
- IR dealing with any kind of other entity or object: works of art, software, courses offered at a university, people (as experts, to hire, for a date), products of any kind.

Text, speech, and images, printed or digital, carry information, hence *information* retrieval. Not so for other kinds of objects, such as hardware items in a store. Yet IR methods apply to retrieving books or people or hardware items, and this article deals with IR broadly, using "document" as stand-in for any type of object. Note the difference between retrieving information about objects (as in a Web store catalog) and retrieving the actual objects from the warehouse.

Utility, Relevance, and IR System Performance

Utility and relevance underlie all IR operations. A document's utility depends on three things, topical relevance, pertinence, and novelty. A document is *topically relevant* for a topic, question, or task if it contains information that either directly answers the question or can be used, possibly in combination with other information, to derive an answer or perform the task. It is *pertinent* with respect to a user with a given purpose if, in addition, it gives just the information needed; is compatible with the user's background and cognitive style so he can apply the information gained; and is authoritative. It is *novel* if it adds to the user's knowledge. Analogously, a soccer player is topically relevant for a team if her abilities and playing style fit the team strategy, pertinent if she is compatible with the coach, and novel if the team is missing a player in her position.

Utility might be measured in monetary terms: "How much is is it worth to the user to have found this document?" "How much is this player worth to us?" "How much did we save by finding this software?" In the literature, the term "relevance" is used imprecisely; it can mean utility or topical relevance or pertinence. Many IR systems focus on finding topically relevant documents, leaving further selection to the user.

Relevance is a matter of degree; some documents are highly relevant and indispensable for the user's tasks; others contribute just a little bit and could be missed without much harm (see ranked retrieval in the section on *Matching*).

From relevance assessments we can compute measures of retrieval performance such as

recall	=	$\frac{\text{relevant items correctly retrieved}}{\text{all relevant items in the collection}}$	How good is the system at finding relevant documents?
discrimination	=	$\frac{\text{irrelevant items correctly rejected}}{\text{all irrelevant items in the collection}}$	How good is the system at rejecting irrelevant documents?
precision	=	$\frac{\text{relevant items retrieved}}{\text{all items retrieved}}$	Depends on discrimination, recall, and the # of relevant documents

Evaluation studies commonly use recall and precision or a combination; whether these are the best measures is debatable. With low precision, the user must look at several irrelevant documents for every relevant document found. More sophisticated measures consider the gain from a relevant document and the expense incurred by having to examine an irrelevant document. For ranked retrieval, performance measures are more complex. All of these measures are based on assessing each document on its own, rather than considering the usefulness of the retrieved set as a whole; for example, many relevant documents that merely duplicate the same information just waste the user's time, so retrieving fewer relevant documents would be better.

How Information Retrieval Systems Work

IR is a component of an information system. An information system must make sure that everybody it is meant to serve has the information needed to accomplish tasks, solve problems, and make decisions, no matter where that information is available. To this end, an information system must (1) actively find out what users need, (2) acquire documents (or computer programs, or products, or data items, and so on), resulting in a collection, and (3) match documents with needs. Determining user needs involves (1.1) studying user needs in general as a basis for designing responsive systems (such as determining what information students typically need for assignments), and (1.2) actively soliciting the needs of specific users, expressed as query descriptions, so that the system can provide the information (Figure 2). Figuring out what information the user really needs to solve a problem is essential for successful retrieval. Matching involves taking a query description and finding relevant documents in the collection; this is the task of the IR system (Figure 3, at end).

Query description	Document titles	Relevant
Production and uses of plastic pipes	1 The production of copper pipes	
As the examples show, simple word match is often not enough; retrieving documents and assessing relevance require knowledge: The system needs to know that polyethylene and PVC are plastics, that tube is another word for pipe, that artery in the context of 6 means a major street and in 7 a pipe in the body, usually made of plastic.	2 Cost of plastic pipe manufacture	√
	3 Polyethylene water pipes	√
	4 Steel rod manufacture	
	5 Spiral PVC tubes as cooling elements	√
	6 Innovative plastic surface for new city artery	
	7 Artificial arteries help heart bypass patients	√
	8 Plastic mouthpieces in making smoking pipes	
Bioinformatics	1 Bioinformatics	√
Bioinformatics is the application of sophisticated computer methods to studying biology. This is another illustration of the *variability of language* IR systems must deal with.	2 Computer applications in the life sciences	√
	3 Biomedical informatics	√
	4 Modeling life processes	√
	5 Modeling traffic flow	
	6 Modeling chemical reactions in the cell	√
Jewish-Gentile relations	1 We played with our non-Jewish friends.	√
This could be a question to the Shoah Foundation's collection of transcribed testimonies from Holocaust survivors. None of the stories that shed light on this question has the query phrase in it. Relevance must be inferred from the entire context.	2 We were taunted in school.	√
	3 Aryan people had many advantages.	
	4 My mother talked often to the neighbors.	√
	5 Jews were deported to concentration camps.	
	6 Jews were forbidden to attend concerts.	√

Figure 2. Query descriptions compared with document or story titles

The simplest text retrieval systems merely compare words in the query description with words in the documents (title, abstract, or full text) and rank documents by the number of matches, but results are often poor (Figure 2). A good IR system provides the access points required to respond to user needs in retrieval and selection. This means preparing user-oriented *document representations* (Figure 4) that describe a document by several statements using <*relationships*> as verbs and Entities as subjects and objects. The allowable Entity Types and <*relationship types*> define what kinds of information the system can store; they make up the *conceptual schema*.

Statement				Data field
Document	*<written by>*	Person	John Smith	Author
Document	*<has title>*	Text	Artificial arteries help heart ...	Title
Document	*<has abstract>*	Text	A clinical study ... showed that ...	Abstract
Document	*<contains word or phrase>*	Phrase	artificial arteries	Free text
Document	*<relevant for>*	Subject	Blood Vessel Prosthesis	Descriptor
Document	*<describes tool for>*	Function	Coronary Artery Bypass	Function
Document	*<has URL>*	URL	www.healtheduc.com/heart/...	URL

Figure 4. Document representation as a group of statements

For some entity types (in the example Person, Text, Phrase, and URL), values can be freely chosen; for others (Subject and Function), values come from a *controlled vocabulary* that fixes the term used for a concept. For example, pipe is used for the concept also known as tube, so the user needs to enter only one term. If the user enters tube, the system (or the user) follows the thesaurus cross-reference

tube USE ST pipe (ST = Synonymous Term)

The thesaurus also includes conceptual cross-references:

pipe BT hollow object (BT = Broader Term) and
pipe NT capillary (NT = Narrower Term)

(For the structure of thesauri, see the article on Information Organization.) The conceptual schema and the thesaurus must of course reflect user needs.

If an entity (such as a document or a data file) is sought as a source of data/information, the data about the entity are used as *metadata* (data describing data); thus, the data in Google's catalog of Web pages are used primarily as metadata.

Steps in the IR Process

An IR system prepares for retrieval by *indexing documents* (unless the system works directly on the document text) and *formulating queries*, resulting in document representations and query representations, respectively; the system then *matches* the representations and *displays* the documents found and the user *selects* the relevant items. These processes are closely intertwined and dependent on each other. The search process often goes through several iterations: Knowledge of the features that distinguish relevant from irrelevant documents is used to improve the query or the indexing (*relevance feedback*).

Indexing: Creating Document Representations

Indexing (also called cataloging, metadata assignment, or metadata extraction) is the manual or automated process of making statements about a document, lesson, person, and so on, in accordance with the conceptual schema (see Figure 4). We focus here on subject indexing – making statements about a document's subjects. Indexing can be *document-oriented* – the indexer captures what the document is about, or *request-oriented* – the indexer assesses the document's relevance to subjects and other features of interest to users; for example, indexing the testimonies in Figure 2 with Jewish-Gentile relations, marking a document as interesting for a course, or marking a photograph as publication quality. Related to indexing is *abstracting* –

creating a shorter text that describes what the full document is about (*indicative abstract*) or even includes important results (*informative abstract*, summary). *Automatic summarization* has attracted much research interest.

Automatic indexing begins with raw *feature extraction*, such as extracting all the words from a text, followed by refinements, such as eliminating stop words (and, it, of), stemming (pipes ⇒ pipe), counting (using only the most frequent words), and mapping to concepts using a thesaurus (tube and pipe map to the same concept). A program can analyze sentence structures to extract phrases, such as labor camp (a Nazi camp where Jews were forced to work, often for a company; phrases can carry much meaning). For images, extractable features include color distribution or shapes. For music, extractable features include frequency of occurrence of notes or chords, rhythm, and melodies; refinements include transposition to a different key.

Raw or refined features can be used directly for retrieval. Alternatively, they can be processed further: The system can use a *classifier* that combines the evidence from raw or refined features to assign descriptors from a pre-established index language. To give an example from Figure 2, the classifier uses the words life and model as evidence to assign bioinformatics (a descriptor in Google's directory). A classifier can be built by hand by treating each descriptor as a query description and building a query formulation for it as described in the next section. Or a classifier can be built automatically by using a training set, such as the list of documents for bioinformatics in Figure 2, for machine learning of what features predict what descriptors. Many different words and word combinations can predict the same descriptor, making it easier for users to find all documents on a topic Assigning documents to (mutually exclusive) classes of a classification is also known as *text categorization*. Absent a suitable classification, the system can produce one by *clustering* – grouping documents that are close to each other (that is, documents that share many features).

Query Formulation: Creating Query Representations

Retrieval means using the available evidence to predict the degree to which a document is relevant or useful for a given user need as described in a free-form *query description*, also called *topic description* or *query statement*. The query description is transformed, manually or automatically, into a formal *query representation* (also called *query formulation* or *query* for short) that combines features that predict a document's usefulness. The query expresses the information need in terms of the system's conceptual schema, ready to be matched with document representations. A query can specify text words or phrases the system should look for (free-text search) or any other entity feature, such as descriptors assigned from a controlled vocabulary, an author's organization, or the title of the journal where a document was published. A query can simply give features in an unstructured list (for example, a "bag of words") or combine features using Boolean operators (structured query). Examples:

	Bag of words:	(pipe tube capillary plastic polyethylene production manufacture)
	Boolean query:	(pipe OR tube OR capillary) AND (plastic OR polyethylene) AND (production OR manufacture)

The Boolean query specifies three ANDed conditions, all of which are necessary (contribute to the document score); each condition can be filled by any of the words joined by OR; one of the words is as good as two or three. If some relevant documents are known, the system can use them as a training set to build a classifier with two classes: relevant and not relevant.

Stating the information need and formulating the query often go hand-in-hand. An intermediary conducting a *reference interview* helps the user think about the information need

and find search terms that are good predictors of usefulness. An IR system can show a subject hierarchy for browsing and finding good descriptors, or it can ask the user a series of questions and from the answers construct a query. For buying a digital camera, the system might ask the following three questions:

- What kind of pictures do you take (snapshots, stills, ...)?
- What size prints do you want to make (5x7, 8x10, ...)?
- What computer do you want to transfer images to?

Without help, users may not think of all the features to consider. The system should also suggest synonyms and narrower and broader terms from its thesaurus. Throughout the search process, users further clarify their information needs as they read titles and abstracts.

Matching the query representation with entity representations

The match uses the features specified in the query to predict document relevance. In *exact match* the system finds the documents that fill all the conditions of a Boolean query (it predicts relevance as 1 or 0). To enhance recall, the system can use *synonym expansion* (if the query asks for pipe, it finds tubes as well) and *hierarchic expansion* or *inclusive searching* (it finds capillary as well). Since relevance or usefulness is a matter of degree, many IR systems (including most Web search engines) rank the results by a score of expected relevance (*ranked retrieval*). Consider the query Housing conditions in Siemens labor camps. Figure 5 illustrates a simple way to compute relevance scores: Each term's contribution is a product of three weights: The *query term weight* (the importance of the term to the user), the *term frequency (tf)* (the number of occurrences of the term in the document, synonyms count also), and the rarity of the term or *inverse document frequency (idf)* on a logarithmic scale. If document frequency = .01 (1 % or 1/**100** of all documents include the term), then idf = **100** or $10^{\mathbf{2}}$ and log(idf) = **2**. For example, in Figure 5 the contribution of housing to relevance score of Document 1 is

query weight 2 * log(idf) 4 * tf (term frequency in document) 5 = 40

(Google considers, in addition, the number of links to a Web page.) Usually (but not in the simple example), scores are normalized to a value between 0 and 1.

Query term (weight		housing (weight 2)	conditions (1)	Siemens (2)	"labor camps" (3)	Score
idf, log(idf)		10,000, log=4	100, log=2	100,000, log=5	10,000, log=4	
term(tf) (tf = frequency of the term in each document)	Doc. 1	barracks (5 times)	conditions (3)	Siemens (2)	"labor camps" (4)	
		40 +	6 +	20 +	48 =	114
	Doc. 2	housing (3 times)	conditions (2)	Siemens (2)	"labor camps" (4)	96
	Doc. 3	housing (3 times)	conditions (4)	Siemens (1)	"labor camps" (4)	90
	Doc. 4.	housing (3 times)	conditions (3)	Siemens (2)	"labor camps" (3)	86
	Doc. 5	housing (2 times)	conditions (10)		"labor camps" (1)	48

Figure 5. Computing relevance scores

Selection

The user examines the results and selects relevant items. Results can be arranged in rank order (examination can stop when enough information is found); in subject groupings, perhaps created by automatic classification or clustering (similar items can be examined side by side); or by date. Displaying title + abstract with search terms highlighted is most useful (title alone is too short, the full text too long). Users may need assistance with making the connection between an item found and the task at hand.

Relevance Feedback and Interactive Retrieval

Once the user has assessed the relevance of a few items found, the query can be improved: The system can assist the user in improving the query by showing a list of features (assigned descriptors; text words and phrases, and so on) found in many relevant items and another list from irrelevant items. Or the system can improve the query automatically by learning which features separate relevant from irrelevant items and thus are good predictors of relevance. A simple version of automatic query adjustment is this: increase the weights of features from relevant items and decrease the weights of features from irrelevant items

IR System Evaluation

IR systems are evaluated with a view to improvement (*formative evaluation*) or with view to selecting the best IR system for a given task (*summative evaluation*). IR systems can be evaluated on system characteristics and on retrieval performance. System characteristics include the following:

- the quality of the conceptual schema (Does it include all information needed for search and selection?);
- the quality of the subject access vocabulary (index language and thesaurus) (Does it include the necessary concepts? Is it well structured? Does it include all the synonyms for each concept?);
- the quality of human or automated indexing (Does it cover all aspects for which an entity is relevant at a high level of specificity, while avoiding features that do not belong?);
- the nature of the search algorithm;
- the assistance the system provides for information needs clarification and query formulation; and
- the quality of the display (Does it support selection?).

Measures for retrieval performance (recall, discrimination, precision, novelty) were discussed in the section *Relevance and IR system performance*. Requirements for recall and precision vary from query to query, and retrieval performance varies widely from search to search, making meaningful evaluation difficult. Standard practice evaluates systems through a number of test searches, computing for each a single measure of goodness that combines recall and precision, and then averaging over all the queries. This does not address a very important system ability: the ability to adapt to the specific recall and precision requirements of each individual query. The biggest problem in IR evaluation is to identify beforehand all relevant documents (the recall base); small test collections have been constructed for this purpose, but there is a question of how well the results apply to large-scale real-life collections. The most important evaluation efforts of this type today are TREC and TDT (see Further Reading).

Outlook: Beyond Retrieval

Powerful statistical and formal-syntax-based methods of *natural language processing* (NLP) extract meaning from text, speech, and images and create detailed metadata for support of more

focused searching. *Data mining and machine learning* discover patterns in large masses of data. Sophisticated database and *expert systems* search and correlate huge amounts of different types of data (often extracted from text) and answer questions by inference. New *visualization* techniques using high-resolution displays allow users to see patterns and large networks of linked information. Sophisticated user models allow *intelligent customization*. IR can be integrated into day-to-day work: A medical IR system can process a patient's chart, find several relevant articles, and prepare a tailor-made multi-document summary, or it can deduce the drugs to be prescribed. A legal IR system can take an attorney's outline of the legal issues in a case, find relevant cases or sections of cases, and arrange them according to the outline to give the attorney a running start on writing a brief. All these advances contribute to an unprecedented level of support for problem solving, decision making, and intellectual work.

Dagobert Soergel

See also Information Organization; Information Filtering; Search

Engines; Ontologies

Further Reading

Important Periodicals and Conference Proceedings

Annual Review of Information Science and Technology
Journal of the American Society for Information Science and Technology
Information Processing and Management
Journal of Documentation
Association for Computing Machinery, Special Interest Group on Information Retrieval (SIG-IR) Conference Proceedings
Joint Conference on Digital Libraries (JCDL) and European Conference on Digital Libraries (ECDL) Proceedings
Text REtrieval Conference (TREC), co-sponsored by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA). (There are also pointers to other important evaluation conferences on this website.) Retrieved January 22, 2004, from http://trec.nist.gov/

General Textbooks

Chu, H. (2003). *Information representation and retrieval in the digital age*. Medford, NJ: Information Today.
Soergel, D. (1985). *Organizing information: Principles of database and retrieval systems***.** Orlando, FL: Academic Press.
Blair, D. C. (1990). *Language and representation in information retrieval*. Amsterdam: Elsevier Science.
Baeza-Yates, R., & Rubiero-Neto, B. (1999). *Modern information retrieval*. Reading, MA: Addison Wesley.
Boiko, B. (2002). *Content management bible*. New York: Hungry Minds. Retrieved January 22, 2004, from http://metatorial.com/index.asp
Frakes, W. B., & Baeza-Yates, R. (Eds.). (1992). *Information retrieval: Data structures and algorithms*. Engelwood Cliffs, NJ: Prentice-Hall.

Negnevitsky, M. (2001). *Artificial intelligence: A guide to intelligent systems***.** Reading, MA: Addison Wesley.
Sparck Jones, K., & Willett, P. (1997). *Readings in information retrieval***.** San Francisco, CA: Morgan Kaufmann.
Witten, J., & Bainbridge, D. (2002). *How to build a digital library***.** San Francisco, CA: Morgan Kaufmann.

Other Literature on IR Systems

Berners-Lee, T., Hendler, J., Lassila, O. (2001). The semantic web. *Scientific American, 284*(5), 34<N>43, Retrieved January 22, 2004, from http://www.sciam.com
Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine**.** 7th International World Wide Web Conference (WWW7). *Computer Networks and ISDN Systems, 30*(XXX). Retrieved January 22, 2004, from www-db.stanford.edu/~backrub/google.html, www7.scu.edu.au/programme/fullpapers/1921/com1921.htm, decweb.ethz.ch/WWW7/00/
Feldman, S. (2000). The answer machine. *Searcher, 8*(1), 1<N>21, 58<N>78. Retrieved January 22, 2004, from http://www.infotoday.com/searcher/jan00/feldman.htm

Searching

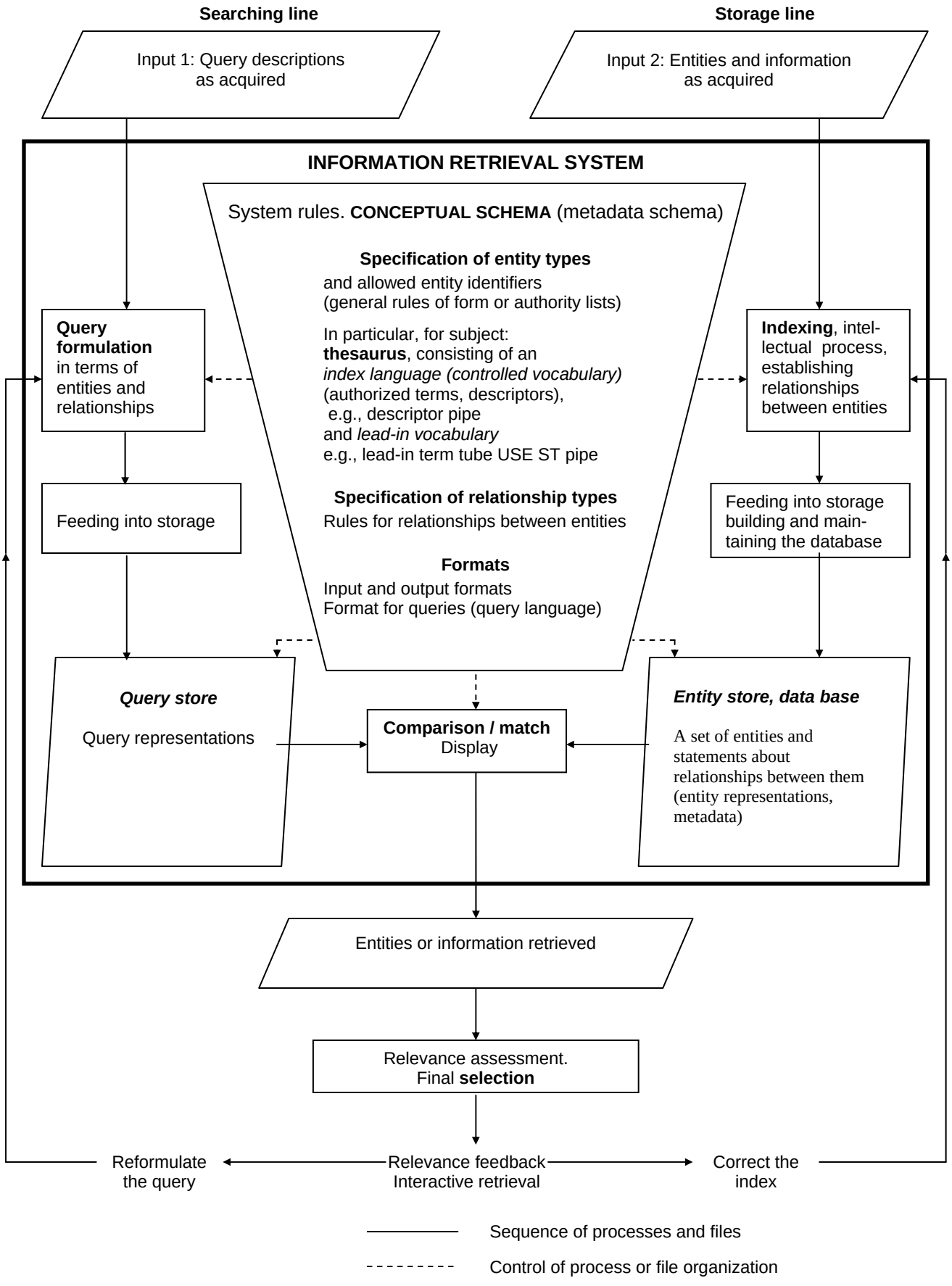
[**Tutorial on searching the Web**] Retrieved January 22, 2004, from www.lib.berkeley.edu/TeachingLib/Guides/Internet/FindInfo.html
Hert, C. A. (1997). *Understanding information retrieval interactions: Theoretical and practical applications***.** Stamford, CT: Ablex.

Natural Language Processing (NLP) in IR

Feldman, S. (1999). NLP Meets the Jabberwocky: Natural language processing in information retrieval. *ONLINE, May 1999*. Retrieved January 22, 2004, from www.onlinemag.net/OL1999/feldman5.html
Jackson, P., & Moulinier, I. (2002). *Natural language processing for online applications: Text retrieval, extraction, and categorization***.** Amsterdam: John Benjamins.

Relevance and IR System Evaluation

Wilson, P. (1973). Situational relevance. *Information Storage and Retrieval, 9*(8), 457<N>471.
Soergel, D. (1994). Indexing and retrieval performance: The logical evidence. *Journal of the American Society for Information Science, 4*(8), 589<N>599.
Also see the TREC conference, above.


Searching line
Storage line
Input 1: Query descriptions as acquired
Input 2: Entities and information as acquired
INFORMATION RETRIEVAL SYSTEM
System rules. CONCEPTUAL SCHEMA (metadata schema)
Specification of entity types
and allowed entity identifiers
(general rules of form or authority lists)
In particular, for subject:
thesaurus, consisting of an
index language (controlled vocabulary)
(authorized terms, descriptors),
e.g., descriptor pipe
and lead-in vocabulary
e.g., lead-in term tube USE ST pipe
Specification of relationship types
Rules for relationships between entities
Formats
Input and output formats
Format for queries (query language)
Query formulation in terms of entities and relationships
Indexing, intellectual process, establishing relationships between entities
Feeding into storage
Feeding into storage building and maintaining the database
Query store
Query representations
Comparison / match
Display
Entity store, data base
A set of entities and statements about relationships between them (entity representations, metadata)
Entities or information retrieved
Relevance assessment.
Final selection
Reformulate the query
Relevance feedback
Interactive retrieval
Correct the index
Sequence of processes and files
Control of process or file organization