ELSEVIER International Journal of Medical Informatics 47 (1997) 5–26

International Journal of Medical Informatics

Information retrieval: an overview of system characteristics

F. Wiesman [a,*], Arie Hasman [a], H.J. van den Herik [b]

[a] *Department of Medical Informatics, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands*
[b] *Department of Computer Science, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands*

Abstract

The paper gives an overview of characteristics of information retrieval (IR) systems. The characteristics are identified from the descriptions of 23 IR systems. Four IR models are discussed: the Boolean model, the vector model, the probabilistic model and the connectionistic model. Twelve other characteristics of IR models are identified: search intermediary, domain knowledge, relevance feedback, natural language interface, graphical query language, conceptual queries, full-text IR, field searching, fuzzy queries, hyptertext integration, machine learning, and ranked output. Finally, the relevance of IR systems for the World Wide Web is established. © 1997 Elsevier Science B.V.

Keywords: Boolean model; Information retrieval; Vector model

1. Introduction

This paper provides an overview of the characteristics distinguishing information retrieval (IR) systems from one another. The overview deals with 23 IR systems described in sufficient detail for an adequate comparison. We do not aim at making quantitative statements about the IR systems characteristics but at enumerating characteristics for comparing all possible IR systems. The choice of the 23 systems does not fully reflect the IR market, because some emphasis is put on systems that apply artificial-intelligence techniques. Nevertheless, this paper provides a clear picture of the current techniques and developments in IR.

Section 2 briefly introduces terminology used in IR. Section 3 describes the most important IR models. Twelve characteristics of IR systems are discussed in Section 4. Section 5 discusses an important area for the application of IR: the World Wide Web. Finally, Section 6 presents the conclusions.

* Corresponding author. Tel.: +31 43 3882295/2240; fax: +31 43 3671052; e-mail: weisman@mi.unimaas.nl

2. Background

The *function of an IR system* is to lead its users to those documents that will best enable them to satisfy their information need [1]. This information can be of various types: text, sound, images, video, and so on. 'Real-world documents' can be stored in a computer at different levels of detail: as reference only (for instance, published as: F. Wiesman, Graphical information retrieval by browsing meta-information, Comput. Methods Progr. Biomed. 53 (3) (1997) 135–152), by reference together with an abstract or the entire contents. All IR systems have some type of description of the document contents: the document representation. This may be a list of key terms[1], a conceptual graph, a frame, or any other representational form.

The user communicates his information need to the IR system by formulating a *query*. Queries usually exist of a list of key terms, which may be combined with connectives, such as AND and OR. Other possibilities are queries formulated in a natural language or in graphically-oriented query languages. These topics are discussed in Sections 4.1.3 and 4.1.4. Once a query is formulated, the IR system determines which document representations meet the query (the matching procedure) and reports this to the user. The user may decide to reformulate the query if the results are not satisfactory.

The following two quantities together form a measure for the effectiveness of an IR system [2]:

Recall: The part of the relevant documents that is actually retrieved

Precision: The part of the retrieved documents that is actually relevant

It is trivial to get a high recall at the cost of precision. Likewise, a high precision can be attained at the cost of recall. The problem is to find a good balance between recall and precision. Buckland and Gey analyze this problem theoretically [3]. Tague-Sutcliffe gives an overview of the problems that arise with the evaluation of IR systems [4].

The generation of document representations is called indexing. The quality of the indexing process is defined as the degree to which the representation reflects the content of the document. A high quality is vital for the proper operation of an IR system. It depends on the document representation form whether the indexing can be performed automatically. Simply assigning key terms can be done automatically when using statistical techniques. Computers do so with satisfactory results [5,6]. For each word occurring in the document the frequency is determined. Common words (or *stop words*), such as *the*, *a*, *in*, *of*, *from*, are eliminated. On the basis of its frequency each word is assigned a weight. For the weighting process different formulas are in use.

More complex representations that truly reflect the content of a document are still to be constructed *manually*. These encompass advanced formalisms such as frames and first-order logic, in which documents are not represented at a global level but at paragraph or sentence level. The advantage is the greater expressive power of the query language. Additionally, by representing facts from documents a query system can answer questions about the contents of documents. This is called a *question answering system*. The manual construction of document representations required by an advanced representation formalism imposes restrictions on the

[1] Throughout this paper, we use 'key term' instead of the more commonly used 'key word', because it indicates better that it may consist of multiple words. For instance, Graves' disease therapy can be considered as one key term.

size of the system. An advanced system that uses complex representations is TOPIC [7].

Text-only IR systems are further developed than multimedia IR systems. The problem with non-textual media is the difficulty of making clear their semantics to the computer; this requires image and speech recognition techniques. Most current systems do not automatically extract content representations, but merely features (e.g. the colours used in an image). Aigrain and Longueville [8] and Shneier and Abdel-Mottaleb [9] report on IR with images and Smoliar and Zhang [10] conducted experiments on IR with video. For this paper, we assume that information consists of documents containing text.

3. Models for information retrieval

When looking at an IR system the question arises: what IR model has been used? An IR model specifies a document representation, a query representation and a matching procedure. The majority of IR systems is based on the Boolean model. The vector model is the most frequently used in experimental environments [11]. Both will be discussed here. Other models are: the probabilistic model; the connectionistic model; the cluster model; the rule-based model; the fuzzy-logic or fuzzy-set model and the semantic model. From these the first two will be discussed since they have an extensive record.

The topics in this paper will be illustrated by the following case: a user with only a little knowledge of IR is looking for information on gall bladder surgery. She is especially interested in scientific publications on complications that may occur as a result of one particular operation: choledochojejunostomy. In this operation a direct connection (stoma) is made between the common bile duct (ductus choledochus) and the small gut (jejunum).

3.1. The Boolean model

The most popular model for building IR systems—and especially commercial ones—is the *Boolean model*. In this model a document is represented by a set of key terms: chosen from a fixed set of key terms, or, possibly automatically, from the documents themselves. Queries and documents can be matched by checking for each document whether the associated key terms satisfy the query. A query consists of a list of key terms connected by logic operators, such as AND, OR and NOT.

For our sample user the query may read:

(ductus choledochus OR common bile duct) AND surgery AND complication

This query can be formalized as A AND B AND C. Compared with the query A AND B it may be expected that the former query selects fewer documents. This touches upon a well-known problem of the Boolean model: the more key terms are involved with AND operators, the fewer documents are retrieved. The decrease of documents retrieved can be very drastic. The cause is that the AND operation imposes severe restrictions on the presence of the key terms. Vice versa, leaving out one or two key terms may result in a (too) large set of retrieved documents. An analogous but converse reasoning holds for the OR operator: the more key terms are added, the larger the set of retrieved documents.

Next to this problem the Boolean model has three drawbacks [12,13]:

- A good query is difficult to formulate, especially for IR novices, since the impact of complex combinations of the operators is hard to grasp.
- The relative importance of the key terms for a query cannot be specified.

- Arranging the retrieved documents in order of relevance is difficult. This is further discussed in Section 4.8.

Of course, the Boolean model also has advantages:

- The Boolean model can be easily implemented and has low computational cost [14].
- Its query language is more expressive than that of other models [15].
- The model is fit for users who know exactly what they are looking for [16].

The following seven systems use the Boolean model: WinSpirs[2], [17–22].

3.2. The vector model

In a *vector-model* IR system containing n key terms, an n-dimensional space is defined such that each axis is associated with a different key term[3]. Every key term has a weight. Documents and queries are represented by vectors in that space. A *document vector* is an n-tuple in which every coordinate is represented by the weight of the corresponding key term. The higher the weight, the more important the key term is to the document associated with the vector. A *query vector* is defined analogously, only the weights express the importance of the key terms for the user. Usually a 1 is used for desired key terms and a 0 for undesired key terms.

[2] WinSpirs is a commercially available IR system from Silverplatter. No literature is available, but we refer to Silverplatter's home page: http://www.silverplatter.com.

[3] Instead of key terms, Cavnar associates with each axis a four-letter string [23]. The document is considered as a sequence of four-letter strings (e.g. retieval is stored as reti, etie, tiev, ieva, eval). Thus only parts of words are indexed. The cost of redunancy is made up firstly because the system is robust against spelling-errors in the query (as in the example). Secondly, the system is language-independent because there is no need for a stop-word list.

The weights are assigned on the basis of word frequency. There are many ways of doing this [24]; we present a well-known example [13]. The weight of a key term w in document i is defined as:

$$\text{weight}(w, i) = \frac{f_{wi} \log(N/n_w)}{\sqrt{\sum_{k=1}^{W_i} (f_{ki})^2 (\log(N/n_k))^2}} \qquad (1)$$

where f_{wi} is the frequency that key term w occurs in document i, N the number of documents, n_w the number of documents in which key term w occurs and W_i the total number of key terms in document i.

The weight is proportional to the number of times the key term occurs in the document. Key terms that also occur in other documents are relatively unimportant, hence the weight is diminished by a factor $\log(N/n_w)$. The length of a vector is thus dependent on the occurrence and frequency of the key terms involved. To avoid the influence of their length the vectors are normalized (in the denominator).

There are several ways of calculating the similarity between a document vector and a query vector. One of them is the normalized dot product of the two vectors. As a case in point we take a universe of three key terms (see Fig. 1):

⟨gallstone, angina, complication⟩

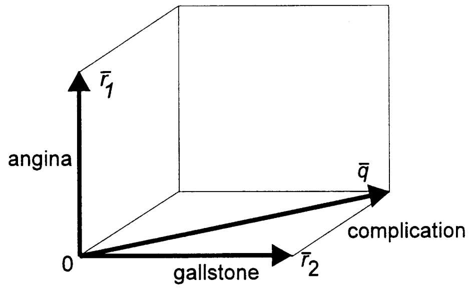

Fig. 1. Vector space with three key terms.

We assume that only two documents are present. Document 1 is on complications with angina and document 2 is on complications with gallstones. Since the term complication occurs in both (all) documents the weight of complication is 0 in both documents. This implies, after normalization, the following document representations for document 1 and document 2, respectively:

$$\bar{r}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \qquad \bar{r}_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

We now want to retrieve documents with the key terms gallstone and complication by the query vector

$$\bar{q} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

For a query vector $\bar{q}$ and a document vector $\bar{r}$ in a system with n key terms the similarity is:

$$\text{similarity}(\bar{q}, \bar{r}) = \frac{\sum_{i=1}^{n} q_i r_i}{\sqrt{\sum_{i=1}^{n} (q_i)^2 \sum_{i=1}^{n} (r_i)^2}} \tag{2}$$

This formula applied on the example results in:

$$\text{similarity}(\bar{q}, \bar{r}_1) = 0; \qquad \text{similarity}(\bar{q}, \bar{r}_2) = 0.7$$

Document 2 scores better than document 1 and therefore matches better with the query than document 1, hence document 2 is retrieved.

In practice a universe of n key terms does not mean that each document is described by n weights; the vast majority of the weights will be zero because most documents treat a limited number of topics. Therefore only combinations of key terms with weights are stored for weights that are unequal to zero. Thus, the vector model does not require much more memory than the Boolean model, and the response time is also reasonable.

Advantages of the vector model are:

- It is possible to assign weights to key terms in a query (simply by replacing the ones and zeros in the query vector).
- The similarity measure can be used to present the results in order of relevance.
- Many researchers assume that the retrieval results obtained with the vector model are better than those obtained with the Boolean model. This is supported by various experiments [25,26]. However, in the study of Lu et al. between Boolean and vector systems no differences in effectiveness were found [27]. An explanation may be that in this study a much larger document collection was used.

The vector model does not have the disadvantages of the Boolean model, but it has others [12]:

- Key terms are supposed to be independent.
- In a query no logic relations (such as AND, OR and NOT) between key terms can be used.

Shaw Jr. et al. surveyed over 100 IR papers published in 1985–1994, and concluded that from all models the vector model is by far the most frequently used [11]. Of the systems we have studied six use the vector model: [28–33].

3.3. The probabilistic model

The basis for the *probabilistic model* is the *probability ranking principle* [34]:

The best possible retrieval results are achieved when documents are shown in the order of their probable relevance to the query.

The task of the IR system is to compute the relevance of a document to a query. This section follows the account given by Salton and McGill [14]. A query consists of a set of key terms chosen from a fixed universe of key terms. A document contains a set of terms from the same universe:

$$Doc = \langle term_1, \ldots, term_t \rangle$$

A document is retrieved if the following condition is satisfied:

$$p(Rel|Doc) \geq p(Notrel|Doc)$$

where *Rel* means the document is relevant and *Notrel* that the document is not relevant to the query. With Bayes' rule [35] this can be rewritten as:

$$\frac{p(Doc|Rel)p(Rel)}{p(Doc|Notrel)p(Notrel)} \geq 1$$

The retrieved documents are ordered using the left-hand side of the inequality. A problem is how to determine the necessary probabilities. This can be done using document samples of which the relevance is known. If queries consist of only one key term or key terms are assumed independent the relevance of each query term that matches a document term can be expressed as the following odds-ratio:

$$w = \frac{r/(R-r)}{(n-r)/(N-n-(R-r))} \tag{3}$$

where N is the number of documents in the collection, n is the number of documents that contains the term, R is the number of relevant documents and r is the number of relevant documents that contain the term. Given a query $\langle w_1, w_2, \ldots, w_t \rangle$ where w_i is the weight as determined in Eq. (3), and a document $\langle x_1, x_2, \ldots, x_t \rangle$ with $x_i = 1$ if the key term is present in the document and $x_i = 0$ if it is not, we can define the similarity function:

$$\text{similarity}(query, document) = \sum_{k=1}^{t} x_k w_k$$

Kwok [36] and Robertson et al. [37], the two systems in this survey that use the probabilistic model, modified Eq. (3) to include the frequency of the key term in the document. Kwok also included the frequency of the key term in the entire document set.

According to Savoy [26] the effectiveness of the probabilistic model is clearly better than the Boolean model, and slightly worse than the vector model. The probabilistic model has the following disadvantages:

- Key terms are supposed to be independent of each other (as in the vector model).
- There is no method for estimating term relevance at the beginning, when no relevant documents are known.

A special form of the probabilistic model is the *Bayesian network*. Such a network defines in terms of probabilities how the relevance of a particular key term changes as soon as the relevance of a related key term changes. Because of its suitability for integration with hypertext systems this model is becoming increasingly popular. The computational complexity is high, which suggests that the model is not fit for larger networks. Refs. [16,38,39] use this model.

3.4. The connectionistic model

Neural networks create a form of *connectionism* which is also applicable in IR. A neural network consists of a number of simple processing units called neurons. They communicate with each other by sending signals over a large number of weighted network connections. Each neuron does nothing else than receiving signals and computing an output signal that is propagated to other neurons. Normally the neurons are divided into several layers. The *input layer* is used to offer a problem to the network by activating

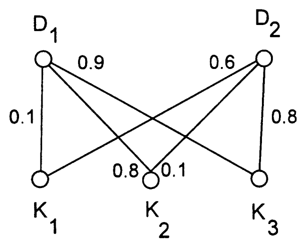

Fig. 2. Simple neural network for two documents and three key terms.

the appropriate nodes. The resulting activation of the *output layer* nodes determines the solution. Between these layers there may be other layers, the *hidden layers*. A network can be trained by adjusting its weights, for instance, as a reaction to the difference between the desired output and the actual one.

For IR purposes each key term can be associated with an input neuron and each document with an output neuron. A query is presented to the network by activating the neurons which are associated with the desired key terms. The network then calculates the output signals. The activated output neurons correspond with the desired documents. Recall and precision can be improved by training the network. Training is an essential ingredient of neural networks. In principle all weights initially are assigned random values. After the user has evaluated the results of a query, the system can adjust the weights according to a particular *learning rule*. Several learning rules are known, varying from very simple to complex.

Fig. 2 shows a network with three input neurons, K_1, K_2 and K_3, which represent the key terms gallstone, angina and complication, and two output neurons, D_1 and D_2, which represent two documents. The *activation* of an output neuron D_i is a function of its total input. In this example it is defined as:

$$\text{activation}(D_i) = \text{sgn}\left(\sum_j w_{ij}a_j - 1\right)$$

where w_{ij} is the weight between nodes i and j, a_j is the activation of neuron K_j and sgn is defined as:

$$\text{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

A document is selected only if the activation of the corresponding output neuron is greater than zero. If a query activates the neurons of gallstone (K_1) and complication (K_3) then the activations of the output neurons D_1 and D_2 are:

$$\text{activation}(D_1) =$$

$$\text{sgn}(1 \cdot 0.1 + 0 \cdot 0.8 + 1 \cdot 0.9 - 1) = \text{sgn}(0) = 0$$

$$\text{activation}(D_2) =$$

$$\text{sgn}(1 \cdot 0.6 + 0 \cdot 0.1 + 1 \cdot 0.8 - 1) = \text{sgn}(0.4) = 1$$

Output neuron D_2 is activated because its value is greater than zero; therefore document 2 is retrieved. If both activations were greater than zero, both documents would have been retrieved.

The system can learn by taking into account relevance feedback. Normally, relevance feedback is used to reformulate a query, but in the connectionistic model it is also used for the improvement of the document representation. In the example of Fig. 2 the user may decide that D_2 is not relevant. In that case the weights between D_2 and the activated input neurons, K_1 and K_3 are decreased. Weights are increased for documents that are judged relevant. The initial weights can be determined by taking random weights and training the network on a set of documents with known relevances. Another approach is to determine the weights in a way similar to vector model indexing.

It is striking that a network with only an input layer and an output layer strongly resembles the vector model. This resemblance disappears when hidden layers or connections between input neurons are added. Layaida and Caron do the latter: a connection between two input neurons is made if the corresponding key terms occur in at least one document [40]. The idea is that thus neurons of synonymous key terms are also activated.

Advantages of the connectionistic model are:

- Learning is a part of the model.
- Hidden layers make the constraint of independent key terms superfluous.

No results of large-scale projects are available yet. This may be due to the number of neurons that is required to represent large numbers of documents and key terms and the resulting number of computations, specially in the learning phase. Refs. [12,40–42] use this model.

4. Characteristics of information retrieval systems

By choosing an IR model the IR system is not completely determined; the IR model is only a basis. An IR-system designer has to decide on many characteristics whether they should be included, and how they should be realized. This section discusses twelve characteristics of the systems we examined.

Table 1 summarizes the characteristics (vertical) of the examined systems (horizontal). The *model* row indicates whether the *Bo*olean, *ve*ctor, *pr*obabilistic model, *Ba*yesian, or *co*nnectionistic IR model is used. For the other characteristics (search intermediary, Section 4.1; domain knowledge, Section 4.1.1; relevance feedback, Section 4.1.2; natural language interface, Section 4.1.3; graphical query language, Section 4.1.4; conceptual queries, Section 4.2; full-text IR, Section 4.3; field searching, Section 4.4; fuzzy queries, Section 4.5; hypertext integration, Section 4.6; machine learning, Section 4.7; ranked output, Section 4.8) it is indicated whether or not they are used in the system. Every system is denoted by the first author of the reference used.

4.1. Search intermediaries

The more powerful the query language of an IR system, the more accurate the user can formulate the query. However, formulating a query becomes more difficult for the users. It is therefore not surprising that especially Boolean IR systems aid users with the formulation and reformulation of queries. Such an aid, which acts as an intermediary between user and IR system, is called a *search intermediary*.

There are several reasons for providing an IR system with a search intermediary. We already mentioned hiding a complex query language. Another purpose is providing active help during formulation and reformulation. Finally, a search intermediary can be used to limit the cost of searching if connection time is charged. The user formulates the query off-line with the search intermediary and only then connects to the IR mechanism. The system of StPierre et al. makes extensive use of this approach; it is even possible to make the search intermediary connect to multiple IR mechanisms for one query [31].

An *intelligent search intermediary* is a search intermediary that helps the user with the aid of knowledge of the domain of the documents and/or search tactics. An obvious way to realize an intelligent search intermediary is through an expert system. The user formulates a problem (how do I find exactly all the documents I need) and a model of the knowledge of a human expert who can solve

Table 1
Characteristics of IR systems

	Bodner	Callan	Can	Charoenkitkam	Chen	Consens	Crouch	Findler	Gauch	Gordon	Humphrey	Kwok	Layaida	Motro	Robertson	Rose 91	Rose 97	Savoy	Smith	StPierre	Turtle	Yoon	Win-Spirs
Model	ve	ba	ve	bo	co	bo	co	?	bo	ve	bo	pr	co	ve	pr	co	ve	ba	bo	ve	ba	bo	bo
Intermediary	+	+	−	−	−	−?	−	?	+	−	+	−	+	+	−	−	−	−	+	+	−	+?	−
Dom. knowledge	+	−	−	+	+	−	−	+	+	−	+	−	−	−	−	+	−	−	+	−	−	+	+
Relev. feedback	−	−	−	−	−	−	−	−	−	−[a]	−	+	+	−	+	+	+	+	−	+	+	−	−
Natural language	−	+	−	−	−	−	+	+	−	−	−	−	+	−	−	−	−	−	+[b]	+	−	−	−
Graphical	−	−	−	+	−	+	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−	−	−
Conceptual	−	+	−	−	+	−	−	?	+	−	−	−	−	−	−	+	−	−	+	−	−	−	+
Full-text	?	−	?	+?	−	−	+	−	+	−	−	−	+	−	+	−	+	−	−	+	−	−	+[c]
Field searching	−	−	−	−	+	+	−	−	−	−	+?	−	−	+	−	+	−	−	−	−	−	−	+
Fuzzy	−	+[d]	−	−	−	−	−	+	−	?	−	−	−	+	−	−	−	−	−	+[d]	−	−	−
Hypertext	−	−	−	−	−	+	+	−	−	−	−	−	−	−	−	−	−	+	−	−	−	−	−
Learning	−	−	−	−	−	−	−	−	−	+	−	−	+	−	−	+	−	−	−	−	−	−	−
Ranking	+	+	?	+	+?	?	+	+	+	+	?	+	+	+	+	−?	+	+	−	+	+	+	−

'+' indicates that the system concerned has this characteristic; '−' indicates that this is not the case and '?' indicates that the literature does not give a definite answer. The letter after 'Model' indicates which model is used: *bo*olean, *ve*ctor, *pr*obabilistic, *ba*yesian, or *co*nnectionistic.

[a] The user can give feedback on retrieval results, but this influences the document representations, and does not trigger query reformulation.

[b] The user is presented a natural language interpretation of the query that consists of a key term list.

[c] Full-text on abstract and title.

[d] Fuzzy queries cannot be used with the natural language interface.

the problem (e.g. a librarian) is used to solve the problem.

As a case in point we refer to the Automatic Query Reformulation system of Gauch and Smith [19]. The underlying system contains the complete contents of the documents, and can retrieve relevant text fragments given a Boolean query. A search intermediary acts as an interface between this system and the user. It consists of a domain knowledge representation and an expert system that can reformulate queries and present retrieval results in order of estimated relevance to the user. Relevance of a text is judged on the basis of the relative frequency of the query key terms in the text, whether the key terms were proposed by the user or the system, and the contextual distance between the key terms. At the reformulation stage different techniques are applied:

- Expansion of queries by adding related terms
- Adaptation of the context (e.g. how close two key terms must be in the text)
- Manipulation of Boolean operators (e.g., replacing an AND by an OR)

We have looked at nine systems with a search intermediary: [18–21,29,31,32,39,40].

Below we discuss characteristics of IR systems that are often realised through a search intermediary: domain knowledge, relevance feedback, natural language interface and graphical query language.

4.1.1. Domain knowledge

Knowledge of the domain of the documents may be used in the IR process. This may be especially useful when reformulating a query; the query may be broadened or narrowed by replacing a key term by a more general or a more specific key term. According to Smeaton and Rijsbergen reformulation should be done using domain knowledge instead of statistical deliberations [43]. Adding key terms to a query on the basis of statistical relations with terms in the query may have negative consequences for the results. For example: adding the key term connectionism to a query containing the key term neural may be the consequence of the co-occurrence of both key terms in many documents. As an undesired result also documents about neural disorders will be found.

Domain knowledge can be represented in several ways. One way is by a *thesaurus*: a list of terms that states for each term its synonyms, more specific terms and a more general term [44]. If a query with the key term choledochojejunostomy results in too few hits, the system may, after consulting the thesaurus, replace choledochojejunostomy by the more general term biliary tract surgery.

An example of a large thesaurus (some 70 000 concepts with some 96 000 key terms) containing general knowledge is Wordnet [45,46]. It is used by Bodner and Song [32] and Charoenkitkarn et al. [22]. Bodner and Song use two thesauri to automatically expand queries: a domain-specific thesaurus and WordNet. Their experiments showed improved effectiveness. Charoenkilkarn et al. only let the user browse the thesaurus. This facility was not very useful because the concepts in WordNet were too general. Voorhees also reports negative results with using WordNet: queries and key terms were disambiguated by using the is-a relations in WordNet, but it had a negative effect on the effectiveness of the system [47]. Krovetz and Kroft give an overview of the subject of ambiguity in IR [48].

The construction of a thesaurus is a difficult and tedious task; it requires a human domain expert. Automatic construction has been investigated by various researchers [32,42,49–51].

The resulting thesauri are not as detailed and accurate as manually constructed thesauri because these approaches are all based on statistical methods. It is not possible to determine automatically the exact nature of the relationship between two key terms, only *whether* they are related (or at best, that one key term is narrower than the other).

Another but similar way of representing domain knowledge is by a semantic network. Semantic networks were originally developed to model the human representation of knowledge. Major components are concepts and relations between concepts. A relation between two concepts constitutes a fact. Example facts that can be represented are 'choledochojejunostomy is a type of biliary tract surgery' and 'gallstone has complication cholecystitis'. A semantic network uses a broader range of relations than a thesaurus, hence a query mechanism that uses it can perform more sophisticated reasoning about the improvement of queries. For the medical domain a CD-ROM exists with a semantic network containing a large number of medical key terms (over 300 000) and their relationships. This Unified Medical Language System from the American National Library of Medicine [52] also encompasses the well-known MeSH thesaurus (Medical Subject Headings) which is used in many medical IR systems [53].

WinSpirs, [22,42] make passive use of domain knowledge: the system supplies it to the user. Active use of domain knowledge, that is, the system itself uses the domain knowledge during query formulation, is made by [18–21,32,41,54].

4.1.2. Relevance feedback

A system that can reformulate a query on the basis of the user's evaluation of query results makes use of *relevance feedback*. The idea is that the new query will retrieve more documents like those that were relevant (according to the user) and less documents that are like those that were irrelevant. In the vector model this can be realized simply by assigning a higher value to key terms from the query that occur in only relevant documents, and to those that occur only in irrelevant documents a lower value. This works best when relevant documents are tightly clustered (in terms of key term space), irrelevant documents are also tightly clustered and the distance between relevant and irrelevant documents is relatively large [14]. Techniques have been devised to realize relevance feedback in the Boolean model, but they are not commonly applied. In general, relevance feedback does improve retrieval effectiveness greatly [55]. A limitation is the dependence of the user's willingness to supply feedback.

Relevance feedback is applied by: [31,33,36–38,40,41,56].

4.1.3. Natural language interface

A reason for unsatisfactory performance may be the language used to formulate queries: a formal language is an obstacle for many people. A natural-language interface enables the user to formulate queries in a natural language. For example:

> I am looking for scientific publications on complications that may occur with choledochojejunostomy.

There are two approaches to dealing with such a query: one can either parse the query and try to 'understand' its meaning, or one can try to sift out the important key terms. The former approach is realised as a special kind of search intermediary because it translates a natural-language query to a query in the language of the underlying IR model.

Turtle compared the effectiveness of natural language queries (on a number of systems) with Boolean queries (on one system) [25]. Recall and precision were clearly higher for the natural language queries. There are, however, some disadvantages:

Automatic processing: Understanding natural language is still a difficult task for computers. Within a specific domain, i.e. a well-defined limited domain, the results are nowadays satisfactory. This disadvantage does not pertain to the approach where key terms are only sifted from the query.

Ambiguity: When using natural language there is a greater chance that a query is ambiguous. This holds especially for short queries. If the system identifies an ambiguity, it can ask the user which of the possible meanings is intended.

Laboriousness: Formulating and typing entire sentences as queries takes more time than entering a few key terms.

[12,31,39,40,54] allow their users to formulate their queries in natural language. From these four only Crouch et al. [12] really try to parse the sentences; the other systems simply eliminate uninteresting words and pass the remainder to the query mechanism. EP-X has a completely different approach: it accepts a list of key terms as a query and then formulates possible interpretations of the query in natural language leaving the choice to the user [18].

4.1.4. Graphical query language

An alternative way of formulating queries is by manipulating graphical objects. This looks like a simple and intuitive means of describing an information need to an IR system.

The *graphical query language* GraphLog has a great expressive power [17]. Fig. 3 is an example of a query in which documents are requested that were created three years ago by others than the user (Me). However, in our opinion this graphical query does not look all that simple.

The expressive power of the system described by Rose and Belew is limited and—therefore—easier to handle than a textual query language [41]. The system is based on the connectionistic model. The formulation of a query does remind of 'query-by-example' as used in some database systems. The user selects some key terms or documents and the system shows in a graphical way documents that are similar to the selected ones. Since pairs of documents may be linked, indicating that a relationship exists between the two documents, these links are also shown. The user then indicates which of the retrieved documents look interesting and which do not. With this information the system can retrieve and show other documents, which in their turn can be evaluated as to find more documents.

Charoenkitkarn et al. describe an even simpler graphical query language: queries are constructed by clicking on terms in the document that is shown on the screen [22]. By drawing a line between two terms they are connected by the AND-operator.

4.2. Conceptual queries

Most of the IR systems do not impose any restriction to the user in choosing key terms for a query. This has a number of disadvantages [57]:

Synonym problem: User and indexer may use different key terms for the same notion, causing too few documents to be retrieved.

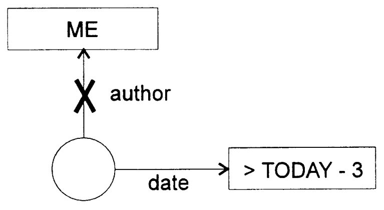

Fig. 3. A query in GraphLog (drawn after [17]).

Spelling problem: User and indexer may spell a key term differently causing too few documents to be retrieved.

Homonym problem: Using a key term which has multiple meanings will result in the retrieval of documents that concern all of these meanings, although the user is interested in only one.

Other systems limit the user to a fixed set of key terms (i.e. fixed as long as the document set remains unchanged). Retrieval is not limited to the literal key word, but also comprises the associated synonyms and spelling variations. This type of query is called a *conceptual query*. Thus, with a conceptual query one does not search for terms but for concepts. Medical IR systems often use a large thesaurus with medical terms: the Medical Subject Headings (MESH) of the National Library of Medicine [53].

Systems that allow conceptual queries are: WinSpirs, [18,19,39,41,42].

4.3. Full-text IR

Traditional library systems can search for key terms that are—often manually—assigned to a document by an indexer or for key terms that occur in the title. *Full-text* IR systems search the entire document contents for a key term.

A document is represented by the words that occur in it; the indexing does not require human intervention. Because a full-text IR system has access to the entire text and not only to the key terms, these systems provide operators that select on proximity of key terms in the text, whether key terms are in the same sentence, or in the same section, chapter or document. Because the user is not limited to a fixed set of key terms full-text IR leads to a higher recall and, in general, to a lower precision.

Full-text IR systems are: WinSpirs[4], [12,19,22,31,33,36,37,40].

4.4. Field searching

We may have raised the impression that queries always consist of key terms that are about the *content* of documents. However, a user may want to constrain the search to documents of a particular author. In some IR systems this is possible by supplying a key term that is a *field*. In these IR systems the documents are like records in a database. One of the fields is the key term field, which is the default field to be searched by a query. Other fields are, for instance, author, year, publication type and journal. An example of a query with a field is: gall bladder surgery AND year = 1997. Or for our sample user, who wants to indicate that only scientific publications are to be retrieved: choledochojejunostomy AND type = scientific publication.

Since documents rarely state whether they are scientific or not, scientific publication cannot be supplied as ordinary key term.

IR systems that allow field searching are: WinSpirs, [17,20,29,41,42]. Many systems that are actually used in practice include field searching. It is an obvious characteristic. Presumably it is not often used in experimental systems because it is well-understood and not interesting to experiment with.

4.5. Fuzzy queries

Fuzzy queries or *vague queries* allow the user to indicate for each key term in the query how important it is relative to the other key terms in the query. This is very useful in the following case: suppose the

[4] Only tital and abstract, not the actual document content, are indexed full-text.

query choledochojejunostomy results in too few documents and the query gall bladder surgery in too many. The query choledochojejunostomy AND gall bladder surgery will of course retrieve even fewer documents (or the same documents). But by indicating that the last key term is less important this can be alleviated: for example

choledochojejunostomy:0.9 OR gall bladder surgery:0.3

A document must be about choledochojejunostomy for an important part and only to some extent about gall bladder surgery to be retrieved.

The importance of key terms in a query may be expressed absolutely, for example by specifying a number between 0 and 1 for each key term, or relatively, by arranging the key terms in order of importance. Users find it easier to indicate relative importance than absolute importance [58].

The best model for fuzzy queries is the vector model because the relevance can easily be incorporated into the query vector. Our sample user from Section 3 will replace the ones and zeroes in the query

$$\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \quad \text{by} \quad \begin{pmatrix} 1 \\ 0 \\ 0.5 \end{pmatrix}$$

if the key term complication is less important. The Boolean model does not allow fuzzy queries, but several extensions of the Boolean model do. These are the fuzzy set model [59], the P-Norm model [60] and the Infinite-One model [61]. Properties of these systems are discussed by Lee [58].

Systems that allow fuzzy queries are described by: [29,31,39,54]. In the latter two systems, fuzzy queries cannot be used in combination with the natural language interface.

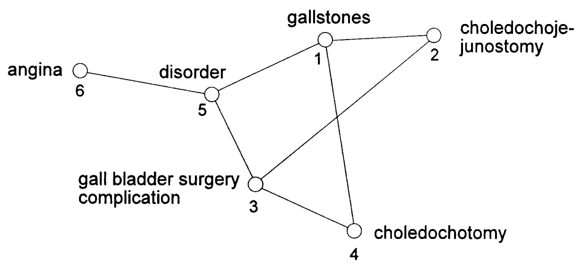

Fig. 4. A hypertext network.

4.6. Hypertext integration

Hypertext systems play an important role in documentary information systems. A hypertext system contains a large amount of information, usually text, divided into small parts called hypertext nodes. The user can travel from node to node by following links between the nodes. The user is supposed to find his information when he browses along. This uncontrolled way of searching for information can be augmented by IR techniques. The user can be brought to ‘interesting places’ from where he may start browsing.

Fig. 4 depicts a hypertext network of a (fictitious) medical handbook. A query system that is integrated with the hypertext system will, after a query concerning complications with gall bladder surgery, show hypertext node 3. It is up to the user whether he will look at adjacent nodes. The information in the nodes visited can be a reason for the user to formulate another query which may lead to another part of the hypertext network. Therefore, one can say IR systems and hypertext systems are complementary if they are integrated.

[12,17,38] describe such systems.

4.7. Machine learning

An IR system *learns* if it improves its performance in some way as a consequence of former queries. The following aspects of an IR system are suitable for improvement by learning:

Document representation: The representation of a document may be improved if the user indicates that the document shown is or is not relevant.

Presentation: The order in which query results are presented to the user can be determined by taking into account the user's interests and preferences. For this a *user model* is required.

Thesaurus: On basis of co-occurrence of terms, a full-text system IR system can conclude that terms are synonymous.

There are various ways in which machine learning for an improved document representation is realized: Rose en Belew [41] and Layaida and Caron [40] use the connectionistic model, where—obviously—weights are changed on the basis of the user's rating for the query results.

Gordon has a completely different approach: he uses a *genetic algorithm* [28]. Starting point is the observation that different users search for the same document in different ways. For each document various document representations are made: each document is represented by various document vectors. A query is matched with every representation of a document; the average of the matching scores determines whether a document will be retrieved or not. After the retrieval results have been displayed, the user indicates which documents are relevant and which are irrelevant. Now for each document it can be determined which representations fit the document best. The worst representations are discarded. New, hopefully better, representations are created by 'breeding' the best ones; that is, by concatenating randomly selected pieces of existing representations to a new representation.

The advantage of this technique lies in the increasingly better document representations that are obtained by adaptation to user preferences. Disadvantages are:

- The system must be used extensively in order to produce significantly better results.
- Storing multiple document representations takes extra disk space.
- Users may get confused when identical queries at different occasions yield different results.

4.8. Ranked output

If a query results in a large number of retrieved documents, it is not feasible for the user to scan through the entire set. However, if the results are sorted in order of estimated relevance the user may safely limit himself to the top of the list. A system that sorts its results in such a way is said to produce *ranked output* (or: to apply *relevance ranking*).

In the evaluation of IR systems with ranked output, it is customary to determine the recall and precision at various cut-off points. For instance, recall and precision are determined for the 10, 20 and 30 highest ranked documents. Another practice is to determine the recall at various precision value cut-offs (or vice versa). Thus, the ranking algorithm strongly influences the IR system's effectiveness.

Many IR systems provide ranked output. Harman gives an overview of existing ranking algorithms [62]. She distinguishes two types: ranking of individual documents and ranking of sets of related documents.

We first discuss the first type. In the vector model, obviously, the similarity measure can be used as a basis for ranking. In the probabilistic model the document relevance probability and in the connectionistic model the output neuron activation are used. In the Boolean model things are more complicated. The obvious measure is the number of matching key terms. This only makes sense when at least one OR-operator is used and gives hardly an ordering; it is more of a division in groups. Somewhat better is to take into account the inverse document frequency of each key term, which depends on the frequency a key term occurs in the entire document collection. This measure in fact ranks key terms. Charoenkitkarn et al. use for each matching document the within-document frequency (i.e. the number of times a key term occurs in a document) normalized by the document length [22]. Another, hybrid, approach is to use the Boolean model to retrieve the documents and the vector model to rank them.

Rose and Stevens remark that the ranking should be consistent with the users' expectations [33]. For instance, users expect a document with two matching key terms to be ranked higher than a document with only one matching key term, even if the key term frequencies indicate that the latter document is more important. Rose and Stevens found that their method of adjusting the similarity measure by taking into account the number of matching key terms did not have a negative influence on retrieval effectiveness.

Other systems that use the first type of ranking are described by: [16,19,21,28,29,31, 32,36–40,42].

The best-known example of the second type of ranking is *clustering*. Generally speaking, this is arranging a number of objects in such a way that clusters emerge that group together objects that are similar according to some criterion. This principle is applied to IR in different ways. A possible criterion is the measure of similarity of documents as defined in the vector model. Documents that have many key terms in common form a cluster; different clusters denote different areas of interest. Fig. 5 shows documents that are clustered according to this criterion. All documents are assigned the key term surgery, but three different operations are concerned: balloon angioplasty, choledochotomy and choledochojejunostomy. This results in three clusters. The balloon angioplasty cluster is somewhat distant from the other two because it is a different type of operation. In the system of Crouch et al. the result of a query is a hierarchy of clusters of possibly relevant documents, where documents that seem to be similar are in the same cluster [12]. Can focuses on the problem of updating clusters dynamically, as determining the clusters anew every time documents are added can be very costly [30]. Findler et al. use clustering in a different way: they divide the domain knowledge, which is represented in a semantic network, in clusters using the user's suggestions [54]. Thus the search for documents can be limited.

5. Information retrieval and the World Wide Web

The World Wide Web (www, or Web) is because of its diversity and dimensions a challenge for IR systems. First we explain what the Web is and why IR is important for it, then we discuss several IR systems that are used for the Web.

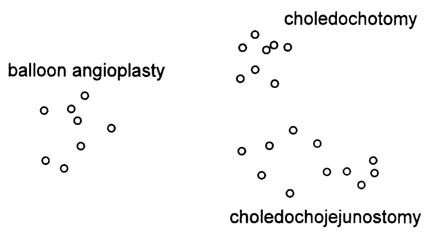

Fig. 5. Clusters of documents concerning surgery.

5.1. The Word Wide Web

The Web is a distributed client/server hypermedia system [63]. A hypermedia system is a hypertext system that contains multimedial information, i.e. a combination of text, images, sounds and movies. It is a client/server system: the hypermedia information resides on the server-side and the user needs a client program to view the information. There are tens of thousands of Web servers world-wide, all connected by hyperlinks. That is why the Web is a *distributed* system. No central authority exists to coordinate the servers, or even register them. This makes it difficult for users to locate the information they need.

In the beginning of the Web, in 1990–1992, it was possible to browse through all information on all servers. When this was not possible anymore, subject directories started to emerge. In its simplest form, a subject directory is a list of Web sites that are important to a particular subject. Every entry has a hyperlink that takes the reader directly to the corresponding site. More sophisticated subject directories give short descriptions of the sites and break down the list into categories. These lists were never complete because they were compiled manually and the Web grew exponentionally. This problem was tackled with the introduction of *Web robots* (or *Spiders*). These are programs that can automatically traverse the Web by recursively following all hyperlinks starting from one arbitrary server. The information gathered can be indexed and serves as the basis for an IR system. Examples of Web robots are Harvest [64] and MOMspider [65]. The first search engine—as a Web IR system is called—that was really usable because it covered a substantial part of the Web was Lycos. Other search engines followed. All Web search engines have in common that they use a Web robot to index text documents on the Web, ignoring non-textual information. Nowadays, a Web without search engines would be unthinkable. Despite the maintenance problem, subject directories continue to exist. Some of them are still maintained by hand, others use Web robots to gather information and humans for classification. The differences between search engines and subject directories have grown smaller: some search engines also offer subject directories, and many subject directories now have a search facility.

5.2. Search engines

We now discuss several Web search engines in terms of the IR system characteristics that were introduced in Section 4.

For the search engines discussed here it is not clear which IR model is used. They are commercial products, and it is therefore difficult to get information about their inner workings. The query language suggests that all use some form of extended Boolean model. All have both a simple and an advanced user interface, allowing simple key term searches and Boolean queries, respectively. The simple query interface may be viewed as an intermediary to the complex query interface. The systems do not provide for conceptual queries and do not have machine learning capabilities. All engines provide ranked output.

AltaVista[5] is a full-text IR system that also allows field searching. For instance, one can search in a title or a hyper link. AltaVista can dynamically generate a graphical thesaurus, called LiveTopics [66], based on a query and its results. Related terms are determined using statistics on co-occurrence of terms in the retrieved documents. The related key terms are grouped in a topic; related topics are depicted with a link between them. The user

[5] URL is: http://www.altavista.digital.com/

can click on key terms and thus include or exclude them in the query. For instance, the query gallstone resulted in 886 documents. Topics that were extracted included: gallstones, complications, abdominal, liver. The first topic contained the key terms gallstones, cholecystectomy, laporoscopic, surgery, surgical, abdomen (which is also a topic, surprisingly), incision, incisions. This facility works well when many documents are retrieved. For smaller numbers the resulting topic graph can be very odd.

Excite[6] is a full-text IR system that not only takes into account the key terms in the query, but automatically broadens the search to the synonyms of the key terms. The synonyms are automatically extracted from the documents[7]. Excite shows together with the query results a list of terms that are related with the query key terms. For instance, the query gallstone resulted in 554 documents. Related key terms shown were: gallstones, gall bladder, bile, pancreatitis, biliary, gastroenterol, thijs (this is the name of the author of an article about gallstones), laparoscopic, calorie, extracorporeal. The idea of this facility is similar to AltaVista's topic graph, but it is more limited. Another facility is relevance feedback: when the user has identified an interesting document, similar documents can be retrieved.

HotBot[8] is a full-text IR system that has extensive field search options. For instance: date, location, media type.

Infoseek[9] accepts, apart from Boolean queries, also natural language queries. Field searches are also possible. Like AltaVista and Excite, query results are accompanied with related topics; these can be selected to narrow the query. For instance, the query gallstone resulted in 2344 documents. Related topics shown were: dieting, medical specialties, gastroenterology. By selecting a topic the original query is narrowed to documents concerning the selected topic. As opposed to AltaVista and Excite, these topics are not determined dynamically, but are selected from a subject directory that is part of Infoseek.

Lycos[10] offers the user various interfaces. There is a Boolean search interface with many operators, a natural language interface and Lycos Pro. With Lycos Pro it is possible for the user to influence the relevance ranking. For each factor that is taken into account during relevance ranking, the user can specify the importance that should be attached to it. Some factors are: frequency of words, words appear early in the text, words appear close together.

Yahoo[11] is actually a subject directory, but is also has a query interface. It differs from most subject directories in that categories are hierarchically organized. The user can either browse through the hierarchy of categories or compose a query. Because Yahoo is updated manually, searches typically have a high precision and low recall. Low recall is not necessarily a problem, because often the most important documents for a subject are present. For example, the query gallstone resulted in two documents. Their titles were shown together with their place in the Yahoo hierarchy: Health: Diseases and Conditions: Digestion and Nutrition Disorders and Regional: US States: New York: Cities: New York City: Business: Health:

[6] URL is: http://www.excite.com/
[7] The mechanism is not clear
[8] URL is http://www.hotbot.com/
[9] URL is: http://www.infoseek.com
[10] URL is http://www.lycos.com/
[11] URL is: http://www.yahoo.com/

Providers: Clinics and Practices. Both documents were indeed very relevant to the query, yet only to laymen. If a query returns no documents at all, the query is automatically passed on to AltaVista.

It is not clear how many documents the search engines have indexed. They provide different figures and use different methods of counting them. However, it is safe to say that the engines discussed here contain some tens of millions of documents. The exception is Yahoo, which contains some hundreds of thousands of documents. Because of the importance of search engines for the Web, they are very frequently used. It is therefore not surprising that these search engines require considerable computing power. For instance, AltaVista uses a machine with ten processors, 6 GB memory, 210 GB hard disk to run its search engine on, a 1.5 GB memory, 30 GB hard disk machine to run its Web robot, and four other machines for handling Web traffic, indexing and Usenet[12].

Chu and Rosenthal propose evaluation criteria for search engines [67]. These include: indexing (coverage, update frequency, portion of document that is indexed), search capabilities (a subset of the characteristics described in Section 4), retrieval performance (recall, precision and response time), output (options and content) and user effort (documentation and user interface). They compared AltaVista, Excite and Lycos. No differences in the response times were found. AltaVista scored the highest precision, considering the ten highest ranked documents only. Recall was not determined. Notess discusses Yahoo and other subject directories [68]. He concludes that despite their incompleteness and inaccuracy, subject directories are effective for searching specific information on the Web.

[12] Usenet is an Internet service that is strictly not part of the Web, but is indexed by AltaVista and several other search engines. It can be viewed as a collection of thousands of electronic notice boards, all dedicated to different subjects, free for everyone to read and contribute to.

6. Conclusions

The construction of an IR system requires a number of decisions to be made. The most important ones have been discussed here, and where possible advantages and disadvantages were given. However, it is not possible to name the 'best choice' because it depends on factors as type and size of the application domain, type of users, and so on.

There are many techniques in IR but only a few have been used in largescale systems: only the Boolean model is widely commercially available. The vector model too is used in commercial systems, but on a small scale. Both the Boolean model and vector model have proven themselves with large document bases, but other retrieval models have only been tested on small document bases; the connectionistic model and the Bayesian network model are possibly too complex.

New developments in IR are not to be expected so much from novel IR models but rather from characteristics as discussed in Section 4. Especially in the field of search intermediaries much research is being carried out. The Web is a challenging testbed for IR systems because of its size, diversity and popularity.

References

[1] S.E. Robertson, The methodology of information retrieval experiment, in: K. Sparck Jones (Ed.), Information Retrieval Experiment, ch. 2, Butterworths, London, 1981.

[2] A. Kent, M. Berry, F.U. Leuhrs, J.W. Perry, Machine literature searching VIII. Operational criteria for designing information retrieval systems, Am. Doc. 6 (2) (1955) 93–101.

[3] M. Buckland, F. Gey, The relationship between recall and precision, J. Am. Soc. Inf. Sci. 45 (January) (1994) 12–19.

[4] J.M. Tague-Sutcliffe, Some perspectives on the evaluation of information retrieval systems, J. Am. Soc. Inf. Sci. 47 (1) (1996) 1–3.

[5] D. Harman, Overview of the second text retrieval conference (TREC-2), in: D.K. Harman (Ed.), The Second Text Retrieval Conf. (TREC-2), National Institute of Standards and Technology, Gaithersburg, MD, 1994, pp. 1–20.

[6] D. Harman, Overview of the third text retrieval conference (TREC-3), in: D.K. Harman (Ed.), The Third Text Retrieval Conf. (TREC-3), National Institute of Standards and Technology, Gaithersburg, MD, 1995, pp. 1–19.

[7] U. Hahn, U. Reimer, Automatic generation of hypertext knowledge bases, ACM SIGOIS Bull. 9 (2–3) (1988) 182–188.

[8] P. Aigrain, V. Longueville, A connection graph for user navigation in a large image bank, in: A. Lichnerowicz (Ed.), Proc. Conf. Intelligent Text and Image Handling 'RIAO 91', Elsevier, Amsterdam, 1991, pp. 67–85.

[9] M. Shneier, M. Abdel-Mottaleb, Exploiting the JPEG compression scheme for image retrieval, IEEE Trans. Pattern Anal. Machine Intell. 18 (8) (1996) 849–853.

[10] S. Smoliar, H.J. Zhang, Content-based video indexing and retrieval, IEEE Multimedia 1 (2) (1994) 62–72.

[11] W.M. Shaw Jr., R. Burgin, P. Howell, Performance standards and evaluations in IR test collections: Vector-space and other retrieval models, Inf. Process. Manag. 33 (1) (1997) 15–36.

[12] D. Crouch, C. Crouch, G. Andreas, The use of cluster hierarchies in hypertext information retrieval, in: Hypertext '89 Proc., 5–8 November, 1989, Pittsburgh, PA, pp. 225–237.

[13] G. Salton, Developments in automatic text retrieval, Science 253 (August) (1991) 974–980.

[14] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.

[15] W.B. Croft, R.H. Thompson, I^3R: A new approach to the design of document retrieval systems, J. Am. Soc. Inf. Sci. 38 (6) (1987) 389–404.

[16] H. Turtle, Text retrieval in the legal world. Tutorial notes presented at the 4th Int. Conf. on AI and Law, 1993.

[17] M. Consens, A. Medezon, Expressing structural hypertext queries in Graphlog, in: Hypertext '89 Proc., 5–8 November, 1989, Pittsburgh, PA, pp. 269–292.

[18] P.J. Smith, S.J. Shute, D. Galdes, M.H. Chignell, Knowledge-based search tactics for an intelligent intermediary system, ACM Trans. Inf. Syst. 7 (3) (1989) 247–270.

[19] S. Gauch, J.B. Smith, Search improvement via automatic query reformulation, ACM Trans. Inf. Syst. 9 (3) (1991) 249–280.

[20] S. Humphrey, Indexing biomedical documents: From thesaural to knowledge-based retrieval systems, Artif. Intell. Med. 4 (5) (1992) 343–371.

[21] Y. Yoon, K. Choi, G. Kim, D. Shin, A hybrid knowledge-based approach to information retrieval, Microprocess. and Microprog. 35 (1) (1992) 329–336.

[22] N. Charoenkitkarn, M. Chignell, G. Golovchinsky, Interactive exploration as a formal text retrieval method: How well can interactivity compensate for unsophisticated retrieval algorithms, in: D.K. Harman (Ed.), Overview of the Third Text Retrieval Conf. (TREC-3), National Institute of Standards and Technology, Gaithersburg, MD, 1995.

[23] W.B. Cavnar, Using an N-gram-based document representation with a vector processing retrieval model, in: D.K. Harman (Ed.), Overview of the Third Text Retrieval Conf. (TREC-3), National Institute of Standards and Technology, Gaithersburg, MD, 1995.

[24] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Inf. Process. Manag. 24 (5) (1988) 513–523.

[25] H. Turtle, Natural language vs. Boolean query evaluation: A comparison of retrieval performance, in: W.B. Croft, C.J. van Rijsbergen (Eds.), Proc. 17th Annu. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, Springer-Verlag, London, 1994, pp. 212–220.

[26] J. Savoy, Searching information in legal hypertext systems, Artif. Intell. Law 2 (1994) 205–232.

[27] X.A. Lu, J.D. Holt, D.J. Miller, Boolean systems revisited: Its performance and its behavior, in: D.K. Harman (Ed.), The 4th Text Retrieval Conf. (TREC-4), National Institute of Standards and Technology, Gaithersburg, MD, 1996.

[28] M. Gordon, Probalistic and genetic algorithms for document retrieval, Commun. ACM 31 (10) (1988) 1208–1218.

[29] A. Motro, VAGUE: A user interface to relational databases that permits vague queries, ACM Trans. Inf. Syst. 6 (3) (1988) 187–214.

[30] F. Can, Incremental clustering for dynamic information processing, ACM Trans. Inf. Syst. 11 (2) (1993) 143–164.

[31] M. StPierre, J. Fullton, K. Gamiel, J. Goldman, B. Kahle, J. Kunze, H. Morris, F. Schiettecatte, WAIS server, WAIS workstation, WAIS forwarder for UNIX, release 1.1, Technical Report, Wais, 1993.

[32] R.C. Bodner, F. Song, Knowledge-based approaches to query expansion in information retrieval, Lecture Notes in Computer Science 1081 (1996) 146–158.

[33] D.E. Rose, C. Stevens, V-Twin: A lightweight engine for interactive use, in: D.K. Harman (Ed.), TREC Proc.—TREC-5, National Institute of Standards and Technology, Gaithersburg, MD, 1997. To appear; URL is http://www-nlpir.nist.gov/TREC/.

[34] S.E. Robertson, The probability ranking principle in IR, J. Doc. 33 (4) (1977) 294–304.

[35] T. Bayes, An essay toward solving a problem in the doctrine of chance, Philos. Trans. R. Soc. Lond. 53 (1763) 370–418.

[36] K.L. Kwok, Experiments with a component theory of probabilistic information retrieval based on single terms as document components, ACM Trans. Inf. Syst. 8 (4) (1990) 363–386.

[37] S.E. Robertson, S. Walker, M.M. Beaulieu, M. Gatford, A. Payne, Okapi at TREC-4, in: D.K. Harman (Ed.), The 4th Text Retrieval Conf. (TREC-4), National Institute of Standards and Technology, Gaithersburg, MD, 1996.

[38] J. Savoy, D. Desbois, Information retrieval in hypertext systems: an approach using Bayesian networks, Electron. Publ. 4 (2) (1991) 87–108.

[39] J.P. Callan, W.B. Croft, S.M. Harding, The INQUERY retrieval system, in: Proc. 3rd Int. Conf. on Database and Expert Systems Applications, Valencia, Spain, 1992, pp. 78–83.

[40] R. Layaida, A. Caron, Application of the backpropagation algorithm to an information retrieval system, in: RIAO 94 Conference Proceedings—Intelligent Multimedia Information Retrieval Systems and Management, Centre de Hautes Etudes Internationales d'Informatique Documentaire, Paris, 1994, pp. 161–171.

[41] D.E. Rose, R.K. Belew, A connectionist and symbolic hybrid for improving legal research, Int. J. Man–Machine Studies 35 (1) (1991) 1–33.

[42] H. Chen, K.J. Lynch, K. Basu, T. Ng, Generating, integrating, and activating thesauri for concept-based document retrieval, IEEE Expert 8 (2) (1993) 25–34.

[43] A. Smeaton, C. Rijsbergen, The retrieval effects on query expansion on a feedback document retrieval system, Comput. J. 26 (3) (1983) 239–246.

[44] ISO, Documentation—guidelines for the establishment and development of monolingual thesauri, Tech. Rep. ISO 2788, International Organisation for Standardisation, Geneva, 1986.

[45] G.A. Miller, WordNet: An on-line lexical database, Int. J. Lexicography 3 (4) (1990) 235–244.

[46] G. Miller, WordNet: A lexical database for english, Commun. ACM 38 (November) (1995) 39–41.

[47] E.M. Voorhees, Using WordNet to disambiguate word senses for text retrieval, in: Proc. 16th Annu. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, Linguistic Analysis, 27 June–1 July, 1993, Pittsburgh, PA, pp. 171–180.

[48] R. Krovetz, W.B. Kroft, Lexical ambiguity in information retrieval, ACM Trans. Inf. Syst. 10 (2) (1992) 115–141.

[49] C.J. Crouch, B. Yang, Experiments in automatic statistical thesaurus construction, in: Proc. 15th Annu. Int. ACM/SIGIR Conf. on Research and Development in Information Retrieval, 21–24 June, 1992, Copenhagen, Denmark, pp. 77–88.

[50] Y. Jing, W.B. Croft, An association thesaurus for information retrieval, Technical Report UMass 94-17, University of Massachusetts, Amherst, 1994.

[51] H. Chen, B. Schatz, T. Ng, J. Martinez, A. Kirchhoff, C. Lin, A parallel computing approach to creating engineering concept spaces for semantic retrieval: The Illinois Digital Library Initiative project, IEEE Trans. Pattern Anal. Machine Intell. 18 (8) (1996) 771–782.

[52] D. Lindberg, B.L. Humphreys, A.T. McCray, The unified medical language system, in: Yearbook of Medical Informatics 1993, Schattauer, Stuttgart, 1993, pp. 41–51.

[53] H.J. Lowe, G.O. Barnett, Understanding and using the medical subject headings (MeSH) vocabulary to perform literature searches, J. Am. Med. Assoc. 271 (14) (1994) 1103–1108.

[54] N. Findler, S. Maini, A.F.M. Yuen, SHRIF, a general-purpose system for heuristic retrieval of information and facts, applied to medical knowledge processing, Inf. Process. Manag. 28 (2) (1992) 219–240.

[55] G. Salton, C. Buckley, Improving retrieval performance by relevance feedback, J. Am. Soc. Inf. Sci. 41 (4) (1990) 288–297.

[56] H. Turtle, W.B. Croft, Evaluation of an inference network-based retrieval model, ACM Trans. Inf. Syst. 9 (3) (1991) 187–222.

[57] B. Sudarshan, Development of reference retrieval system with simultaneous building of thesaurus, Lib. Sc. 16 (3) (1979) 77–83.

[58] J.H. Lee, Properties of extended Boolean models in information retrieval, in: W.B. Croft, C.J. Rijsbergen (Eds.), Proc. 17th Annu. Int. ACM-SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '94), Springer-Verlag, London, 1994, pp. 182–190.

[59] W.M. Sachs, An approach to associative retrieval through the theory of fuzzy sets, J. Am. Soc. Inf. Sci. 27 (1976) 85–87.

[60] G. Salton, E.A. Fox, H. Wu, Extended Boolean information retrieval, Commun. ACM 26 (11) (1983) 1022–1036.

[61] M.E. Smith, Aspects of the P-norm model of information retrieval: Syntactic query generation, efficiency, and theoretical properties, Technical Report CORNELLCS//TR90-1128, Cornell University, Computer Science Department, May 1990.

[62] D. Harman, Ranking algorithms, in: W.B. Frakes, R. Baeza-Yates (Eds.), Information Retrieval: Data Structures & Algorithms, ch. 14, Englewood Cliffs, Prentice Hall, NJ, 1992.

[63] T. Berners-Lee, R. Cailliau, J.F. Groff, B. Pollermann, World-Wide Web: The information universe, Electronic Networking: Research, Applications and Policy 1 (2) 1992, 52–58.

[64] C.M. Bowman, P.B. Danzig, D.R. Hardy, U. Manber, M.F. Schwartz, Harvest: A scalable, customizable discovery and access system, Technical Report CU-CS-732-94, University of Colorado, Department of Computer Science, 1994.

[65] R.T. Fielding, Maintaining distributed hypertext infostructures: welcome to MOMspider's Web, Comput. Networks ISDN Syst. 27 (November) (1994) 193–204.

[66] F. Bourdoncle, LiveTopics: recherche visuelle d'information sur l'Internet, in: Computer-Assisted Information Searching on Internet (RIAO'97), Montreal, 1997, pp. 651–654.

[67] H. Chu, M. Rosenthal, Search engines for the World Wide Web: a comparative study and evaluation, in: Methodology in Global complexity: Information, Chaos and Control: ASIS Annu. Meeting, 21–24 October, 1996, Baltimore, MD.

[68] G.R. Notess, On the nets, Database 20 (1) (1997) 61–64.