Contents

X-XXXX-XXXX-X/XX/$X.XX+$X.XX

1

Information Retrieval

Vijay V. Raghavan, *University of Louisiana*

Venkat N. Gudivada, *Marshall University*

Zonghuan Wu, *University of Louisiana*

William I. Grosky, *University of Michigan - Dearborn*

CONTENTS

X-XXXX-XXXX-X/XX/$X.XX+$X.XX

Abstract This chapter provides a succinct yet comprehensive introduction to Information Retrieval (IR) by tracing the evolution of the field from classical retrieval models to the ones employed by the Web search engines. Various approaches to document indexing are presented followed by a discussion of retrieval models. The models are categorized based on whether or not they rank output, use relevance feedback to modify initial query to improve retrieval effectiveness, consider links between the Web documents in assessing their importance to a query. IR in the context of multimedia and XML data is discussed. The chapter also provides a brief description of metasearch engines, which provide unified access to multiple Web search engines. A terse discussion of IR products and resources is provided. The chapter is concluded by indicating research directions in IR. The intended audience for this chapter are graduate students desiring to pursue research in IR area and those who want to get an overview of the field.

1.1 Introduction

Information retrieval (IR) problem is characterized by a collection of documents, possibly distributed and hyperlinked, and a set of users who perform queries on the collection to find a right subset of the documents. In this chapter, we trace the evolution of IR models and discuss their strengths and weaknesses in the contexts of both unstructured text collections and the Web.

An IR system is typically comprised of four components:(1) document indexing — representing the information content of the documents; (2) query indexing — representing user queries; (3) similarity computation — assessing the relevance of documents in the collection to an user query request; and (4) query output ranking — ranking the retrieved documents in the order of their relevance to the user query.

Each of sections 1.2 to 1.4 discusses important issues associated with one of the above components. Various approaches to document indexing are discussed in section 1.2; In respect of the query output ranking component, section 1.3 describes retrieval models that are based on exact representations for documents and user queries, employ similarity computation that is based on exact match which results in a binary value (i.e., a document is either relevant or non-relevant), and typically don't rank query output. This section also introduces the next generation retrieval models, which are more general in their query and document representations and rank the query output. In section 1.4, recent work on a class of IR models based on, the so called, *Language modeling approach*, is overviewed. Instead of separating the modeling of retrieval and indexing, these models unify approaches for indexing of documents and queries with similarity computation and output ranking.

Another class of retrieval models recognize the fact that document representations are inherently subjective, imprecise, and incomplete. To overcome these issues, they employ *learning* techniques, which involve user feedback, and *automated query expansion*. Relevance feedback techniques elicit user assessment on the set of documents initially retrieved, uses this assessment to modify the query or document representations, and re-execute the query. This process is iteratively carried out until the user is satisfied with the retrieved documents. IR systems that employ query expansion either involve statistical analysis of documents and user actions or perform modifications to the user query based on *rules*. The rules are either hand crafted or automatically generated using a dictionary or thesaurus. In both relevance feedback and query expansion based approaches, typically weights are associated with the query and document terms to indicate their relative importance in manifesting the query and document information content. The query expansion based approaches recognize that the terms in the user query or not just literal strings, but denote domain concepts. These models are referred to as Concept-based or Semantic retrieval models. These issues are discussed in section 1.5.

In Sections 1.6 and 1.8, we discuss web-based IR systems. Section 1.6 examines recent retrieval models introduced specifically for information retrieval on the Web. These models employ informa-

tion content in the documents as well as citations or links to other documents in determining their relevance to a query. IR in the context of multimedia and XML data is discussed in section 1.7. There exists a multitude of engines for searching documents on the Web including AltaVista, InfoSeek, Google, and Inktomi. They differ in terms of the scope of the Web they cover and the retrieval models employed. Therefore, a user may want to employ multiple search engines to reap their collective capability and effectiveness. Steps typically employed by a *metasearch* engine are described in section 1.8. Section 1.9 provides a brief overview of a few IR products and resources. Finally, conclusions and research directions are indicated in section 1.10.

1.2 Indexing Documents

Indexing is the process of developing a document representation by assigning content descriptors or terms to the document. These terms are used in assessing the relevance of a document to a user query and directly contribute to the retrieval effectiveness of an IR system. Terms are of two types: objective and non-objective. *Objective terms* apply integrally to the document, and in general there is no disagreement about how to assign them. Examples of objective terms include author name, document URL, and date of publication. In contrast, there is no agreement about the choice or the degree of applicability of *non-objective terms* to the document. These are intended to relate to the information content manifested in the document. Optionally, a weight may be assigned to a non-objective term to indicate the extent to which it represents or reflects the information content manifested in the document.

The effectiveness of an indexing system is controlled by two main parameters: indexing exhaustivity and term specificity [Salton, 1989]. *Indexing exhaustivity* reflects the degree to which all the subject matter or domain concepts manifested in a document are actually recognized by the indexing system. When indexing is exhaustive, it results in a large number of terms assigned to reflect all aspects of the subject matter present in the document. In contrast, when the indexing is *non-exhaustive*, the indexing system assigns fewer terms which correspond to the major subject aspects which the document embodies. *Term specificity* refers to the degree of breadth or narrowness of the terms. The use of broad terms for indexing entails retrieving many useful documents along with a significant number of non-relevant ones. Narrow terms, on the other hand, retrieve relatively fewer documents and many relevant items may be missed.

The effect of indexing exhaustivity and term specificity on retrieval effectiveness is explained in terms of recall and precision – two parameters of retrieval effectiveness used over the years in the IR area. *Recall* (R) is defined as the ratio of the number of relevant documents retrieved to the total number of relevant documents in the collection. The ratio of the number of relevant documents retrieved to the total number of documents retrieved is referred to as *precision* (P). Ideally, one would like to achieve both high recall and high precision. However, in reality, it is not possible to simultaneously maximize both recall and precision. Therefore, a compromise should be made between the conflicting requirements. Indexing terms that are narrow and specific (i.e., high term specificity) result in higher precision at the expense of recall. In contrast, indexing terms that are broad and non-specific result in higher recall at the cost of precision. For this reason, an IR system's effectiveness is measured by the precision parameter at various recall levels.

Indexing can be carried out either manually or automatically. *Manual indexing* is performed by trained indexers or human experts in the subject area of the document by using a *controlled vocabulary controlled vocabulary* made available in the form of terminology lists and *scope notes* along with instructions for the use of the terms. Because of the sheer size of many realistic document collections (e.g. the Web) and the diversity of subject material present in these collections, manual indexing is not practical. Automatic indexing relies on a less tightly controlled vocabulary and entails representing many more aspects of a document than is possible under manual indexing. This helps to retrieve a document with respect to a great diversity of user queries.

In these methods, a document is first scanned to obtain a set of terms and their frequency of occurrence. We refer to this set of terms as *term set of the document*. Grammatical function words such as *and*, *or*, and *not* occur with high-frequency in all the documents and are not useful in representing their information content. A precompiled list of such words is referred to as *stopword list*. Words in the stopword list are removed from the term set of the document. Further, stemming may be performed on the terms. *Stemming* is the process of removing the suffix or tail end of a word to broaden its scope. For example, the word *effectiveness* is first reduced to *effective* by removing *ness*, and then to *effect* by dropping *ive*.

1.2.1 Single-term Indexing

Indexing, in general, is concerned with assigning non-objective terms to documents. Indexing can be based on single or multiple terms (or words). In this section, we consider indexing based on single terms and describe three approaches for it: statistical, information-theoretic, and probabilistic.

1.2.1.1 Statistical Methods

Assume that we have N documents in a collection. Let tf_{ij} denote the frequency of the term T_j in document D_i. The term frequency information can be used to assign weights to the terms to indicate their degree of applicability or importance as index terms.

Indexing based on term frequency measure fulfills only one of the indexing aims – recall. Terms that occur rarely in individual documents of a collection are not captured as index terms by the term frequency measure. However, such terms are highly useful in distinguishing documents in which they occur from those in which they do not occur, and help to improve precision. We define the document frequency of the term T_j, denoted by df_j, as the number of times T_j occurs in a collection of N documents. Then, the inverse document frequency (idf), given by $\log \frac{N}{df_j}$, is an appropriate indicator of T_j as a document discriminator.

Both the term frequency and the inverse document frequency measures can be combined into a single frequency-based indexing model. Such a model should help to realize both the recall and precision aims of indexing, since it generates indexing terms that occur frequently in individual documents and rarely in the remainder of the collection. To reflect this reasoning in the indexing process, we assign an importance or *weight* to terms based on both term frequency (tf) and inverse document frequency (idf). The weight of a term T_j in document D_i, denoted w_{ij}, is given by $w_{ij} = tf_{ij} \log \frac{N}{df_j}$. The available experimental evidence indicates that the use of combined term frequency and document frequency factors (i.e., $tfidf$) provides a high level of retrieval effectiveness [Salton, 1989].

Some important variations of the above weighting scheme have been reported and evaluated. In particular, highly effective versions of $tf \cdot idf$ weighting approaches can be found in [Croft, 1983; Salton and Buckley, 1988]. More recently, a weighting scheme known as Okapi has been demonstrated to work very well with some very large test collections [Jones et al., 1995].

Another statistical approach to indexing is based on the notion of *term discrimination value*. Given that we have a collection of N documents and each document is characterized by a set of terms, we can think of each document as a point in the document space. Then the distance between two points in the document space is inversely proportional to the similarity between the documents corresponding to the points. When two documents are assigned very similar term sets, the corresponding points in the document space will be closer (that is, the density of the document space is increased); and the points are farther apart if their term sets are different (that is, the density of the document space is decreased).

Under this scheme, we can approximate the value of a term as a document discriminator based on the type of change that occurs in the document space when a term is assigned to the documents of the collection. This change can be quantified based on the increase or decrease in the average distance between the documents in the collection. A term has a good discrimination value if it

increases the average distance between the documents. In other words, terms with good discrimination value decrease the density of the document space. Typically, high document frequency terms increase the density; medium document frequency terms decrease the density; and low document frequency terms produce no change in the document density. The term discrimination value of a term T_j, denoted dv_j, is then computed as the difference of the document space densities before and after the assignment of term T_j to the documents in the collection. Methods for computing document space densities are discussed in [Salton, 1989].

Medium-frequency terms that appear neither too infrequently nor too frequently will have positive discrimination values; high-frequency terms, on the other hand, will have negative discrimination values. Finally, very low-frequency terms tend to have discrimination values closer to zero. A term weighting scheme such as $w_{ij} = tf_{ij} \cdot dv_j$, which combines term frequency and discrimination value produces a somewhat different ranking of term usefulness than the $tf \cdot idf$ scheme.

1.2.1.2 Information-theoretic Method

In information theory, the least predictable terms carry the greatest information value [Shannon, 1951]. Least predictable terms are those that occur with smallest probabilities. Information value of a term with occurrence probability p is given as $-\log_2 p$. The average information value per term for t distinct terms occurring with probabilities $p_1, p_2, \cdots, p_t$, respectively, is given by:

$$\vec{H} = -\sum_{i=1}^{t} p_i \log_2 p_i \tag{1.1}$$

The average information value given by equation 1.1 has been used to derive a measure of term usefulness for indexing — *signal-noise ratio*. The signal-noise ratio favors terms that are concentrated in particular documents (i.e., low document frequency terms). Therefore, its properties are similar to those of the inverse document frequency. The available data shows that substituting signal-noise ratio measure for inverse document frequency (idf) in $tf \cdot idf$ scheme or for discrimination value in $tf \cdot dv$ scheme did not produce any significant change or improvement in the retrieval effectiveness [Salton, 1989].

1.2.1.3 Probabilistic Method

Term weighting based on the probabilistic approach assumes that relevance judgments are available with respect to the user query for a training set of documents. The training set might result from the top ranked documents by processing the user query using a retrieval model such as the vector space model. The relevance judgments are provided by the user.

An initial query is specified as a collection of terms. Certain number of top ranking documents, with respect to the initial query, is used to form a training set. To compute the term weight, the following conditional probabilities are estimated using the training set: *document relevant to the query, given that the term appears in the document*, and *document non-relevant to the query, given that the term appears in the document* are estimated using the training set [Yu and Salton, 1976; Robertson and Sparck-Jones, 1976].

Assume that we have a collection of N documents of which R are relevant to the user query; that R_t of the relevant documents contain term t; that t occurs in f_t documents. Various conditional probabilities are estimated as follows:

$Pr[t \text{ is present in the document} \mid \text{document is relevant}] = R_t/R$

$Pr[t \text{ is present in the document} \mid \text{document is non-relevant}] = (f_t - R_t)/N - R$

$Pr[t \text{ is absent in the document} \mid \text{document is relevant}] = R - R_t/R$

$Pr[t \text{ is absent in the document} \mid \text{document is non-relevant}] = ((N - R) - (f_t - R_t))/(N - R)$

From these estimates, the weight of term t, denoted w_t, is derived using Bayes's theorem as:

$$w_t = \log \frac{R_t/(R - R_t)}{(f_t - R_t)/(N - f_t - (R - R_t))} \tag{1.2}$$

The numerator (denominator) expresses the odds of term t occurring in a relevant (non-relevant) document. Term weights greater than 0 indicate that the term's occurrence in the document provides evidence that the document is relevant to the query; values less than 0 indicate to the contrary. While the discussion of the above weight may be considered inappropriate in the context of determining term weights in the absence of relevance information, it is useful to consider how methods of computing a term's importance can be derived by proposing reasonable approximations of the above weight under such a situation [Croft and Harper, 1979].

1.2.2 Multi-term or Phrase Indexing

The indexing schemes described above are based on assigning single-term elements to documents. Assigning single terms to documents is not ideal for two reasons. First, single terms used out of context often carry ambiguous meaning. Second, many single terms are either too specific or too broad to be useful in indexing. Term phrases, on the other hand, carry more specific meaning and thus have more discriminating power than the individual terms. For example, the terms *joint* and *venture* do not carry any indexing value in financial and trade document collections. However, the phrase *joint venture* is a highly useful index term. For this reason, when indexing in performed manually, indexing units are composed of groups of terms, such as noun phrases that permit unambiguous interpretation. To generate complex index terms or term phrases automatically, three methods are used: statistical, probabilistic, and linguistic.

1.2.2.1 Statistical Methods

These methods employ *term grouping* or *term clustering* methods that generate groups of related words by observing word co-occurrence patterns in the documents of a collection. Term-document matrix is a two-dimensional array consisting of n rows and t columns. The rows are labeled $D_1, D_2, \cdots, D_n$ and correspond to the documents in the collection; columns are labeled $T_1, T_2, \cdots, T_t$ and correspond to the term set of the document collection. The matrix element corresponding to row D_i and column T_j represents the importance or weight of the term T_j assigned to document D_i. Using this matrix, term groupings or classes are generated in two ways. In the first method, columns of the matrix are compared to each other to assess whether the terms are jointly assigned to many documents in the collection. If so, the terms are assumed to be related and are grouped into the same class. In the second method, the term-document matrix is processed row-wise. Two documents are grouped into the same class if they have similar term assignments. The terms that co-occur frequently in the various document classes form a term class.

1.2.2.2 Probabilistic Methods

Probabilistic methods generate complex index terms based on term–dependence information. This requires considering an exponential number of term combinations, and for each combination an estimate of joint co-occurrence probabilities in relevant and non-relevant documents. However, in reality, it is extremely difficult to obtain information about occurrences of term groups in the documents of a collection. Therefore, only certain dependent term pairs are considered in deriving term classes [Van Rijsbergen, 1977; Yu et al., 1983]. In both the statistical and probabilistic approaches, co-occurring terms are not necessarily related semantically. Therefore, these approaches are not likely to lead to high-quality indexing units.

1.2.2.3 Linguistic Methods

There are two approaches to determining term relationships using *linguistic methods*: term-phrase formation and thesaurus-group generation. A *term phrase* consists of the phrase head which is the principal phrase component, and other components. A term with document frequency exceeding a stated threshold (e.g., $df > 2$) is designated as phrase head. Other components of the phrase should be medium- or low-frequency terms with stated co-occurrence relationships with the phrase head. Co-occurrence relationships are those such as the phrase components should co-occur in the same sentence with the phrase head within a stated number of words of each other. Words in the stopword list are not used in the phrase formation process. The use of only word co-occurrences and document frequencies do not produce high quality phrases, however. In addition to the above steps, the following two syntactic considerations can also be used. Syntactic class indicators (e.g., adjective, noun, verb) are assigned to terms, and phrase formation is then limited to sequences of specified syntactic indicators (e.g., noun-noun, adjective-noun). A simple syntactic analysis process can be used to identify syntactic units such as subject, noun, and verb phrases. The phrase elements may then be chosen from within the same syntactic unit.

While phrase generation is intended to improve precision, thesaurus-group generation is expected to improve recall. A thesaurus assembles groups of related specific terms under more general, higher-level class indicators. The thesaurus transformation process is used to broaden index terms whose scope is too narrow to be useful in retrieval. It takes low-frequency, overly specific terms and replaces them with thesaurus class indicators which are less specific, medium-frequency terms. Manual thesaurus construction is possible by human experts, provided that the subject domain is narrow. Though various automatic methods for thesaurus construction have been proposed, their effectiveness is questionable outside of the special environments in which they are generated.

Others considerations in index generation include case sensitivity of terms (especially for recognizing proper nouns), and transforming dates expressed in various diverse forms into a canonical form.

1.3 Retrieval Models

In this section we first present retrieval models that don't rank output followed by those that rank the output.

1.3.1 Retrieval Models Without Ranking of Output

Boolean Retrieval Model

Boolean retrieval model is a representative of this category. Under this model, documents are represented by a set of index terms. Each index term is viewed as a Boolean variable and has the value *true* if the term is present in the document. No term weighting is allowed and all the terms are considered to be equally important in representing the document content. Queries are specified as arbitrary Boolean expressions formed by linking the terms using the standard Boolean logical operators *and*, *or*, and *not.* The retrieval status value (RSV) is a measure of the query-document similarity. The RSV is 1 if the query expression evaluates to *true;* otherwise the RSV is 0.

All documents whose RSV evaluates to 1 are considered relevant to the query. The Boolean model is simple to implement and many commercial systems are based on this model. User queries can be quite expressive since they can be arbitrarily complex Boolean expressions. Boolean model-based IR systems tend to have poor retrieval performance. It is not possible to rank the output since all retrieved documents have the same RSV. The model does not allow assigning weights to query terms to indicate their relative importance. The results produced by this model are often counter-intuitive. As an example, if the user query specifies ten terms linked by the logical connective *and,*

a document which has nine of these terms is not retrieved. User relevance feedback is often used in IR systems to improve retrieval effectiveness. Typically, a user is asked to indicate the relevance or non-relevance of a few documents placed at the top of the output. Since the output is not ranked, however, the selection of documents for relevance feedback elicitation is difficult.

1.3.2 Retrieval Models With Ranking of Output

Retrieval models under this category include Fuzzy Set, Vector Space, Probabilistic, Extended Boolean or p-*norm*.

1.3.2.1 Fuzzy Set Retrieval Model

Fuzzy set retrieval model is based on fuzzy set theory [Radecki, 1979]. In conventional set theory, a member either belongs to or does not belong to a set. In contrast, fuzzy sets allow partial membership. We define a membership function F which measures the degree of importance of a term T_j in document D_i by $F(D_i, T_j) = k$, for $0 \leq k \leq 1$. Term weights w_{ij} computed using the $tf \cdot idf$ scheme can be used for the value of k. Logical operators *and*, *or*, and *not* are appropriately redefined to include partial set membership. User queries are expressed as in the case of the Boolean model and are also processed in a similar manner using the redefined Boolean logical operators. The query output is ranked using the $RSVs$. It has been found that, fuzzy set based IR systems suffer from lack of discrimination among the retrieved output nearly to the same extent as systems based on the Boolean model. This leads to difficulties in the selection of output documents for elicitation of relevance feedback. The query output is often counter-intuitive. The model does not allow assigning weights to user query terms.

1.3.2.2 Vector Space Retieval Model

The *vector space retrieval model* is based on the premise that documents in a collection can be represented by a set of vectors in a space spanned by a set of normalized term vectors [Raghavan and Wong, 1986]. If the vector space is spanned by n normalized term vectors, then each document will be represented by an n-dimensional vector. The value of the first component in this vector reflects the weight of the term in the document corresponding to the first dimension of the vector space, and so forth. A user query is similarly represented by an n-dimensional vector. The RSV of a query-document is given by the scalar product of the query and the document vectors. The higher the RSV, the greater is the document's relevance to the query.

The strength of the model lies in its simplicity. Relevance feedback can be easily incorporated into this model. However, the rich expressiveness of query specification inherent in the Boolean model is sacrificed in the vector space model. The vector space model is based on the assumption that the term vectors spanning the space are orthogonal, and existing term relationships need not be taken into account. Furthermore, the query-document similarity measure is not specified by the model and must be chosen somewhat arbitrarily.

1.3.2.3 Probabilistic Retrieval Model

Probabilistic retrieval models take into account the term dependencies and relationships, and major parameters such as the weights of the query terms and the form of the query-document similarity are specified by the model itself. The model is based on two main parameters, $Pr(rel)$ and $Pr(nonrel)$, which are probabilities of relevance and non-relevance of a document to a user query. These are computed by using the probabilistic term weights (section 1.2.1) and the actual terms present in the document. Relevance is assumed to be a binary property so that $Pr(rel) = 1 - Pr(nonrel)$. In addition, the model uses two cost parameters, a_1 and a_2, to represent the loss

associated with the retrieval of a non-relevant document and non-retrieval of a relevant document, respectively.

As noted in section 1.2.1, the model requires term-occurrence probabilities in the relevant and non-relevant parts of the document collection, which are difficult to estimate. However, the probabilistic retrieval model serves an important function for characterizing retrieval processes, and provides a theoretical justification for practices previously used on an empirical basis (e.g., introduction of certain term weighting systems).

1.3.2.4 Extended Boolean Retrieval Model

In the extended Boolean model, as in the case of the vector space model, a document is represented as a vector in a space spanned by a set of orthonormal term vectors. However, the query-document similarity is measured in the *extended Boolean* (or p*-norm*) model by using a generalized scalar product between the corresponding vectors in the document space [Salton et al., 1983]. This generalization uses the well-known L_p norm defined for an n-dimensional vector, $\vec{d}$, where the length of $\vec{d}$ is given by $\| \vec{d} \| = \| (w_1, w_2, \cdots, w_n) \| = (\Sigma_{j=1}^{n} w_j^p)^{\frac{1}{p}}$, where $1 \leq p \leq \infty$, and $w_1, w_2, \cdots, w_n$ are the components of the vector $\vec{d}$.

Generalized Boolean *or* and *and* operators are defined for the p-norm model. The interpretation of a query can be altered by using different values for p in computing query-document similarity. When $p = 1$, the distinction between the Boolean operators *and* and *or* disappears as in the case of the vector space model. When the query terms are all equally weighted and $p = \infty$, the interpretation of the query is same as that in the fuzzy set model. On the other hand, when the query terms are not weighted and $p = \infty$, the p-norm model behaves like the strict Boolean model. By varying the value of p from 1 to ∞, we obtain a retrieval model whose behavior corresponds to a point on the continuum spanning from the vector space model to the fuzzy and strict Boolean models. The best value for p is determined empirically for a collection, but is generally in the range $2 \leq p \leq 5$.

1.4 Language Modeling Approach

Unlike the classical probabilistic model, which explicitly models user relevance, a language model views documents themselves as the source for modeling the processes of querying and ranking documents in a collection. In such models, the rank of a document is determined by the probability that a query Q would be generated by repeated random sampling from the document model $M_D : P(Q|M_D)$ [Ponte and Croft, 1998; Lavrenko and Croft, 2001]. As a new alternative paradigm to the traditional IR approach, it integrates document indexing and document retrieval into a single model. In order to estimate the conditional probability $P(Q|M_D)$, explicitly or implicitly, a two-stage process is needed: the *indexing stage* estimates language model for each document and the *retrieval stage* computes the query likelihood based on the estimated document model.

In the simplest case, for the first stage, the maximum likelihood estimate of the probability of each term t under the term distribution for each document D is calculated as:

$$\hat{p}_{ml}(t|M_D) = \frac{tf_{(t,D)}}{dl_D}$$

where $tf_{(t,D)}$ is the raw term frequency of term t in document D and dl_D is the total number of tokens in D [Ponte and Croft, 1998]. For the retrieval stage, given the assumption of independence of query terms, the ranking formula can simply be expressed as:

$$\prod_{t \in D} p_{ml}(t|M_D)$$

for each document. However, the above ranking formula will assign zero probability to a document that is missing one or more query terms. To avoid this, various smoothing techniques are proposed with the aim to adjust the maximum likelihood estimator of a language model so that the unseen terms can be assigned proper non-zero probabilities. A typical smoothing method called *linear interpolation smoothing* [Berger and Lafferty, 1999], which adjusts maximum likelihood model with the collection model $p(t|C)$ whose influence is controlled by a coefficient parameter λ, can be expressed as:

$$p(Q|M_D) = \prod_{t \in Q} (\lambda p(t|M_D) + (1 - \lambda) p(t|C))$$

The effects of different smoothing methods and different settings of smoothing parameter on retrieval performance can be referred to [Zhai and Lafferty, 1998].

Obviously, language model provides a well-interpreted estimation technique to utilize collection statistics. However, the lack of explicit models of relevance makes it conceptually difficult to incorporate language model with many popular techniques in Information Retrieval, such as relevance feedback, psudo-relevance feedback, and automatic query expansion [Lavrenko and Croft, 2001]. In order to overcome this obstacle, more sophisticated frameworks are proposed recently that employ explicit models of relevance and incorporate language model as a natural component, such as risk minimization retrieval framework [Lafferty and Zhai, 2001] and relevance-based language models [Lavrenko and Croft, 2001].

1.5 Query Expansion and Relevance Feedback Techniques

Unlike the database environment, ideal and precise representations for user queries and documents are difficult to generate in an information retrieval environment. It is typical to start with an imprecise and incomplete query and iteratively and incrementally improve the query specification, and consequently, the retrieval effectiveness [Aalbersberg, 1992; Efthimiadis, 1995; Haines and Croft, 1993]. There are two major approaches to improving retrieval effectiveness: automated query expansion, and relevance feedback techniques. The following section discusses automated query expansion techniques and section 1.5.2 presents relevance feedback techniques.

1.5.1 Automated Query Expansion and Concept-based Retreival Models

Automated query expansion methods are based on term co-occurrences [Baeza-Yates and Ribeiro-Neto, 1999], Pseudo-Relevance Feedback (PRF) [Baeza-Yates and Ribeiro-Neto, 1999], concept-based retrieval [Qiu and Frei, 1993], and language analysis [Bodner and Song, 1996; Bookman and Woods, 2003; Mitra, Singhal and Buckley, 1998; Sparck-Jones and Tait, 1984]. Language analysis based query expansion methods are not discussed in this chapter.

1.5.1.1 Term Co-occurrences Based Query Expansion

Term co-occurrences based methods involve identifying terms related to the terms in the user query. Such terms might be synonyms, stemming variations, or terms that are physically close to the query terms in the document text. There are two basic approaches to term co-occurrence identification: global and local analysis. In the global analysis, a similarity thesaurus based on term-term relationships is generated. This approach doesn't work well in general since the term relationships captured in the similarity thesaurus are often invalid in the local context of the user query [Baeza-Yates and Ribeiro-Neto, 1999]. Automatic local analysis employs clustering techniques. Term

co-occurrences based clustering is performed on top-ranked documents retrieved in response to the user's initial query. Local analysis is not suitable in the Web context since it requires accessing the actual documents from a Web server. The idea of applying global analysis techniques to a local set of documents retrieved is referred to as *local context analysis*. A study reported in [Xu and Croft, 1996] demonstrate the advantages of combining local and global analysis.

1.5.1.2 Pseudo-Relevance Feedback Based Query Expansion

In the PRF method, multiple top-ranked documents retrieved in response to the user's initial query are assumed to be relevant. This method has been found to be effective in cases where the initial user query is relatively comprehensive but precise [Baeza-Yates and Ribeiro-Neto, 1999]. However, it has been noted that the methods often results in adding unrelated terms, which has detrimental effect on retrieval effectiveness.

1.5.1.3 Concept-based Retrieval Model

Compared to term phrases, which capture more conceptual abstraction than the individual terms, concepts are intended to capture even higher-level of domain concepts. Concept-based retrieval treats the terms in the user query as representing domain concepts, and not as literal strings of letters. Therefore, it can fetch documents even if they don't contain the specific words in the user query.

There have been several investigations into concept-based retrieval [Belew, 1989; Bollacker et al., 1998; Croft, 1987; Croft et al., 1989; McCune et al., 1989; Resnik, 1995]. RUBRIC (Rule-Based Information Retrieval by Computer) is a pioneer system in this direction. It uses *production rules* to capture user query concepts (or topics). Production rules define a hierarchy of retrieval subtopics. A set of related production rules is represented as an AND/OR tree referred to as *rule-based tree*. RUBRIC facilitates users to define detailed queries starting at a conceptual level.

Only a few concept-based information retrieval systems have been used in the real domains for the following reasons. These systems pay much attention to representing conceptual information without addressing the acquisition of that knowledge. The latter itself is challenging in its own right. Users would prefer to retrieve documents of interest without having to define the rules for their queries. If the system features pre-defined rules, users can then simply make use of the relevant rules to express concepts in their queries.

The work reported in [Kim, 2000] provides a logical semantics for RUBRIC rules, defines a framework for defining rules to manifest user query concepts, and demonstrate a method for automatically constructing rule-based trees from typical thesauri. The latter has the following fields: USE, BT (Broad Term), Narrow Term (NT), RT (Related Term). The USE field represents the terms to be used instead of the given term with almost the same meaning. For example, Plant associations and Vegetation types can be used instead of the term Habitat types. As the names imply, BT, NT, and RT fields list more general terms, more specific terms, and related terms of the thesaurus entry. Typically, NT, BT, and RT fields contain numerous terms. Indiscriminately using all the terms results in an explosion of rules. In [Kim, 2000] a method has been suggested to select a subset of terms in NT, BT, and RT fields. Experiments conducted on small corpus with a domain-specific thesaurus show that concept-based retrieval based on automatically constructed rules is more effective than hand-made rules in terms of precision.

An approach to constructing query concepts using document features is discussed in [Chang et al., 2002]. The approach involves first extracting features from the documents, deriving primitive concepts by clustering the document features, and using the primitive concepts to represent user queries.

The notion of *Concept Index* is introduced in [Nakata, 1998]. Important concepts in the document collection are indexed, and concepts are cross-referenced to enable concept-oriented navi-

gation of the document space. An incremental approach to clustering document features to extract domain concepts in the Web context is discussed in [Wong and Fu, 2000]. The approach to concept-based retrieval in [Qiu and Frei, 1993] is based on language analysis. Their study reveals that language analysis approaches require a deep understanding of queries and documents, which entails in higher computational cost. Furthermore, deep understanding of language still stands as an open problem in the Artificial Intelligence field.

1.5.2 Relevance Feedback Techniques

The user is asked to provide evaluations or relevance feedback on the documents retrieved in response to the initial query. This feedback is used subsequently in improving the retrieval effectiveness. Issues include methods for relevance feedback elicitation and means to utilize the feedback to enhance retrieval effectiveness. Relevance feedback is elicited in the form of either *two-level* or *multi-level* relevance relations. In the former, the user simply labels a retrieved document as *relevant* or *non-relevant*, whereas in the latter, a document is labelled as *relevant*, *somewhat relevant*, or *non-relevant*. Multilevel relevance can also be specified in terms of relationships. For example, for three retrieved documents d_1, d_2, and d_3, we may specify that d_1 is more relevant than d_2 and that d_2 is more relevant than d_3. For simplifying the presentation, we assume two-level relevance and the vector space model. The set of documents deemed relevant by the user comprise *positive feedback*, and the non-relevant ones comprise *negative feedback*.

As shown in Figure 1.1, two major approaches to utilizing relevance feedback are based on modifying the query and document representations. Methods based on modifying the query representation affect only the current user query session and have no effect on other user queries. In contrast, methods based on modifying the representation of documents in a collection can affect the retrieval effectiveness of future queries. The basic assumption for relevance feedback is that documents relevant to a particular query resemble each other in the sense that the corresponding vectors are similar.

Modifying Query Representation

There are three ways to improve retrieval effectiveness by modifying the query representation.

Modification of Term Weights

The first approach involves adjusting the query term weights by adding document vectors in the positive feedback set to the query vector. Optionally, negative feedback can also be made use of by subtracting the document vectors in the negative feedback set from the query vector. The reformulated query is expected to retrieve additional relevant documents that are similar to the documents in the positive feedback set. This process can be carried out iteratively until the user is satisfied with the quality and number of relevant documents in the query output [Rocchio and Salton, 1965].

Modification of query term weights can be based on the positive feedback set, the negative feedback set, or a combination of both. Experimental results indicate that positive feedback is more consistently effective. This is due to the fact that documents in the positive feedback set are generally more homogeneous than the documents in the negative feedback set. However, an effective feedback technique, termed *dec hi*, uses all the documents in the positive feedback set and subtracts from the query only the vectors of highest ranked non-relevant documents in the negative feedback set [Harman, 1992].

The above approaches only require a weak condition to be met with respect to ensuring that the derived query is optimal. A stronger condition, referred to as *acceptable ranking* was introduced and algorithm that can iteratively learn an optimal query has been introduced in [Wong and Yao, 1990].

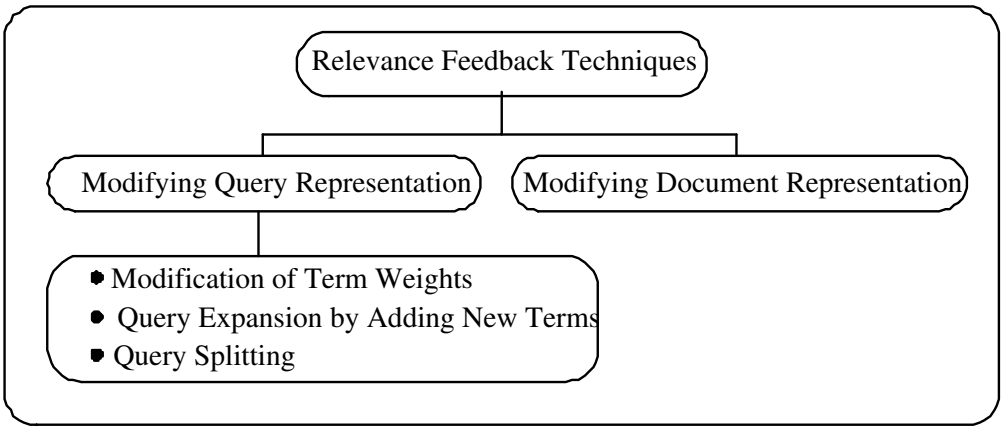

Figure 1.1: A taxonomy for relevance feedback techniques

More recent advances relating to deterministic strategies for deriving query weights optimally are reported in [Herbrich et al., 1998; Tadayon and Raghavan, 1999]. Probabilistic strategies to obtain weights optimally have already been mentioned in sections 1.2.1 and 1.2.2.

Query Expansion by Adding New Terms

The second method involves modifying the original query by adding new terms to it. The new terms are selected from the positive feedback set, and are sorted using measures such as noise (a global term distribution measure similar to idf), postings (the number of retrieved relevant documents containing the term), noise within postings, $noise \times frequency$ within postings (frequency is the $\log_2$ of the total frequency of the term in the retrieved relevant set), $noise \times frequency \times postings$, and $noise \times frequency$. A predefined number of top terms from the sorted list are added to the query. Experimental results show that last three sort methods produced the best results, and adding only selected terms is superior to adding all terms. There is no performance improvement by adding terms beyond twenty [Harman, 1992]. Probabilistic methods that take term dependencies into account may also be included under this category and they have been mentioned in section 1.1.2. There have also been proposals for generating term relationships based on user feedback [Yu, 1975; Wong and Yao, 1993; Jung and Raghavan, 1990].

Query Splitting

In some cases, the above two techniques do not produce satisfactory results because the documents in the positive feedback set are not homogeneous (i.e., they do not form a tight cluster in the document space), or because the non-relevant documents are scattered among certain relevant ones. One way to detect this situation is to cluster the documents in the positive feedback set to see if more than one homogeneous cluster exists. If so, the query is split into a number of sub-queries such that each sub-query is representative of one of the clusters in the positive feedback set. The weight of terms in the sub-query can then be adjusted or expanded as in the previous two methods.

1.5.2.1 Modifying Document Representation

Modifying the document representation involves adjusting the document vector based on relevance feedback, and is also referred to as *user-oriented clustering* [Deogun et al., 1989; Bhuyan et al., 1997]. This is implemented by adjusting the weights of retrieved and relevant document vectors

to move them closer to the query vector. The weights of retrieved non-relevant documents vectors are adjusted to move them farther from the query vector. Care must be taken to insure that individual document movement is small, since user relevance assessments are necessarily subjective. In all the methods, it has been noted that more than two or three iterations may result in minimal improvements.

1.6 Retrieval Models for Web Documents

IR field is enjoying a renaissance and widespread interest as the Web is getting entrenched more deeply into all walks of life. The Web is perhaps the largest, dynamic library of our times and sports a large collection of textual documents, graphics, still images, audio and video collections [Yu and Meng, 2003]. Web search engines debuted in mid 1990s to help locate relevant documents on the Web. IR techniques need suitable modifications to work in the Web context for various reasons. The Web documents are highly distributed — spread over hundreds of thousands of Web Servers. The size of the Web is growing exponentially. There is no quality control — authenticity, editorial process — on Web document creation. The documents are highly volatile — appear and disappear at the will of the document creator.

The aforementioned issues create unique problems for retrieving Web documents. The first issue is what portion of the document to index? Choices are document title, author name(s), abstract, and full text. Though this problem is not necessarily unique to the Web context, it is accentuated given the absence of editorial process and the diversity of document types. Because of the high volatility, any such index can get outdated very quickly and needs to be rebuilt quite frequently. It is an established goodwill protocol that the Web servers be not accessed for full text of the documents in determining its relevance to user queries. Otherwise, the Web servers will get overloaded very quickly. Full text of the documents is retrieved once it has been determined that the document is relevant to a user query. Typically, document relevance to a user query is determined using the index structure built *a priori*. Users of Web search engines, on average, use only two or three terms to specify their queries. About 25% of the search engine queries were found to contain only one term [Baeza-Yates and Ribeiro-Neto, 1999]. Furthermore, the *ploysemy problem* — having multiple meanings for a word — is more pronounced in the Web context due to the diversity of documents.

Primarily, there are three basic approaches to searching the Web documents: Hierarchical Directories, Search Engines, and Metasearch Engines. Hierarchical Directories, such as the ones featured by Yahoo (www.yahoo.com) and Open Directory Project (dmoz.org), feature a manually created hierarchical directory. At the top level of the directory are categories such as Arts, Business, Computers, and Health. At the next (lower) level, these categories are further refined into more specialized categories. For example, Business category has Accounting, Business and Society, Cooperatives (among others) at the next lower level. This refinement of categories can go to several levels. For instance, Business/Investing/Retirement Planning is a category in Open Directory Project (ODP). At this level, the ODP provides hyperlinks to various Web sites that are relevant to Retirement Planning. This level also lists other related categories such as Business/Financial Services/Investment Services and Society/People/Seniors/Retirement. Directories are very effective in providing guided navigation and reaching the relevant documents quite quickly. However, the Web space covered by directories is rather small. Therefore, this approach entails high precision but low recall. Yahoo pioneered the hierarchical directory concept for searching the Web. ODP is a collaborative effort in manually constructing hierarchical directories for Web search.

Early search engines used Boolean and Vector Space retrieval models. In case of the latter, document terms were weighted. Subsequently, HTML (Hypertext Markup Language) introduced meta-tags, using which Web page authors can indicate suitable keywords to help the search engines

in indexing task. Some of the search engines even incorporated relevance feedback techniques (Section 1.5) to improve retrieval effectiveness. Current generation search engines (e.g., Google) consider (hyper)link structure of Web documents in determining relevance of a Web page to a query. Link-based ranking is of paramount importance given that Web page authors often introduce spurious words using HTML meta-tags to alter their page ranking to potential queries. The primary intent of rank altering is to improve Web page hits to promote a business, for example. Link-based ranking helps to diminish the effect of spurious words.

1.6.1 Web Graph

Web graph is a structure obtained by considering Web pages as nodes and the hyperlinks (simply, links) between the pages as directed edges. It has been found that the average distance between connected Web pages is only 19 clicks [Efe et al., 2000]. Furthermore, the Web graph contains densely connected regions that are in turn only a few clicks away from each other. Though an individual link from page p_1 to page p_2 is a weak evidence that the latter is related to the former (since the link may be there just for navigation), an aggregation of links is a robust indicator of importance. When the link information is supplemented with text-based information on the page (or the page text around the anchor), even better search results that are both important and relevant have been obtained [Efe et al., 2000].

When only two links are considered in the Web graph, we obtain a number of possible basic patterns: endorsement, co-citation, mutual reinforcement, social choice, and transitive endorsement. Two pages pointing to each other — *endorsement* — is a testimony to our intuition about their mutual relevance. *Co-citation* occurs when a page points to two other pages. Bibliometric studies reveal that relevant papers are often cited together [White and McCain, 1989]. A page that cites the home page of New York Times is most likely to site the home page of Washington Post also — *mutual reinforcement*. *Social choice* refers to two documents linking to the same page. This pattern implies that the two pages are related to each other since they point to the same document. Lastly, *transitive endorsement* occurs when a page p_1 points to another page p_2, and p_2 in turn points to p_3. Transitive endorsement is a weak one measure of p_3 being relevant to p_1..

Blending these basic patterns gives rise to more complex patterns of the Web graph: *complete bipartite graph*, *clan graph*, *in-tree*, and *out-tree*. If many different pages link (directly or transitively) to a page — that is, the page has high in-degree, it is likely that the (heavily linked) page is an authority on some topic. If a page links to many authoritative pages (e.g., a survey paper) — that is, the page has high out-degree, then the page is considered to be a good source (i.e., hub) for searching relevant information. In the following, we briefly discuss two algorithms for Web page ranking based on Web graph.

1.6.2 Link Analysis Based Page Ranking Algorithm

Google is a search engine, which ranks Web pages by importance based on link analysis of a Web graph. This algorithm is referred to as *PageRank*[1] [Brin and Page, 1998]. The rank of a page depends the number of pages pointing to it as well as the rank of those pointing pages. Let r_p be the rank of a page p and x_p be the number of outgoing links on a page. The rank of p is recursively computed as:

$$r_p = (1 - d) + d \sum_{\forall p;\, q \to p} \frac{r_q}{x_q} \tag{1.3}$$

where d is a damping factor, whose value is selected to be between 0 and 1. It assigns higher importance to pages with high in-degrees or pages that are linked to by highly ranked pages.

[1] Our reference is to the original PageRank algorithm.

1.6.3 HITS Algorithm

Unlike PageRank algorithm (which computes page ranks off-line, independent of user query), HITS (Hyperlink Induced Topic Search) algorithm relies on deducing authorities and hubs in a subgraph comprising results of a user query and the local neighborhood of the query result [Kleinberg, 1998]. *Authorities* are those pages to which many other pages in the neighborhood point to. *Hubs*, on the other hand, point to many good authorities in the neighborhood. They have mutually reinforcing relationships: authoritative pages on a search topic are likely to be found near good hubs, which in turn link to many good sources of information on the topic. The kind of relationships of interest are modeled by special subgraph structures such as bipartite and clan. One challenging problem that arises in this context is called *topic drift*, which refers to the tendency of the HITS algorithm to converge to a strongly connected region that represents just a single topic.

The algorithm has two major steps: sampling and weight-propagation. In the first step, using one of the commercially available search engines, about 200 pages are selected using keyword based search. This set of pages is referred to as the *root set*. The latter is expanded into a *base set* by adding any page on the Web that has a link to/from a page in the root set. The second step computes a weight for each page in the base set. This weight is used to rank the relevance of the page to a query. The output of the algorithm is a short list of pages with the largest hub weights, and a separate list of pages with the largest authority weights.

1.6.4 Topic-Sensitive PageRank

The PageRank algorithm computes the rank of a page statically — page rank computation is independent of user queries. There have been extensions to PageRank in which page ranks are computed for each topic in a pre-determined set off-line [Haweliwala, 2002]. This is intended to capture more accurately the notion of importance of a page with respect to several topics. The page rank value corresponding to a topic which closely corresponds to the query terms is selected in ranking the pages.

1.7 Multimedia and Markup Documents

Though the current Web search engines primarily focus on textual documents, ubiquity of multimedia data (graphics, images, audio, and video) and markup text (e.g., XML documents) on the Web mandate future search engines be capable of indexing and searching multimedia data. Multimedia information retrieval area addresses these issues and the results have culminated in MPEG-7 MPEG-7 [2002] — a standard for describing multimedia data to facilitate efficient browse, search, and retrieval. The standard is developed under the auspices of Moving Pictures Expert Group (MPEG).

1.7.1 MPEG-7

MPEG-7 is called *multimedia content description interface* and is designed to address the requirements of diverse applications — Internet, Medical Imaging, Remote Sensing, Digital Libraries, E-commerce, to name a few. The standard specifies a set of descriptors (i.e., syntax and semantics of features/index terms) and description schemes (i.e., semantics and structure of relationships between descriptions and description schemes), an XML-based language to specify description schemes, and techniques for organizing descriptions to facilitate effective indexing, efficient storage and transmission. However, it does not encompass the automatic extraction of descriptors and features. Furthermore, it does not specify how search engines can make use of the descriptors.

Multimedia feature extraction/indexing is a manual and subjective process especially for semantic-level features. Low-level features such as color histograms are extracted automatically. However, they have limited value for content-based multimedia information retrieval. Because of the semantic richness of audio-visual content, difficulties in speech recognition, natural language understanding, and image interpretation, fully automated feature extraction tools are unlikely to appear in the foreseeable future. Robust semi-automated tools for feature extraction and annotation are yet to emerge in the market place.

1.7.2 XML

The eXtensible Markup Language (XML) is a W3C standard for representing and exchanging information on the Internet. In recent years, documents in widely varied areas are increasingly represented in XML and by 2006 about 25% of LAN traffic will be in XML [EETimes, 2003]. Unlike HTML, XML tags are not predefined and are used to markup the document *content*.

An XML document collection, D, contains a number of XML documents (d). Each such d contains XML-elements (p) and associated with elements are words (w). An element p can have: zero or more words (w) associated with it, a sub-element p, or zero or more attributes (a) with values (w) bound to them. From the information-content point of view, a is similar to p, except that p has more flexibility in information expression and access. An XML document, d, therefore is a hierarchical structure.

D can be represented in (d, p, w) format. This representation has one more component then a typical full-text collection, which is represented as (d, w). Having p in XML document collection D entails benefits including: ability to access D by content-based retrieval; D can be displayed in different formats; and D can be evolved by re-generating p with w. Document d can be parsed to construct a *document tree* (by DOM parser) or to identify events for corresponding event-handlers (by SAX parser). Information content extracted via parsing is used to build an index file and to convert into a database format.

Indexing d encompasses building *occurrence frequency table* freq(p, w) — number of times w occurs in p. Frequency of occurrence of w in d, freq(d, w), is defined as: freq$(d, w) = \sum_p$ freq(p, w). Based on the value of freq(p, w), d is placed in a fast-search data structure.

1.8 Metasearch Engines

A metasearch engine (or metasearcher) is a Web-based distributed IR system that supports unified access to multiple existing search engines. Metasearch Engines emerged in the early 1990s, which provided simple common interfaces to a few major Web search engines. With the fast growth of Web technologies, largest metasearch engines become complex portal systems that can now search on around 1,000 search engines. Some early and current famous metasearch engines are WAIS [Kahle and Medlar, 1991], STARTS [Gravano et al., 1997], MetaCrawler [Selberg and Etzioni, 1997], SavvySearch [Howe and Dreilinger, 1997] and Profusion [Gauch et al., 1996].

In addition to rendering convenience to users, metasearch engines increase the search coverage of the Web by combining the coverage of multiple search engines. A metasearch engine does not maintain its own collection of Web pages but it may maintain information about its underlying search engines in order to achieve higher efficiency and effectiveness.

1.8.1 Software Component Architecture

A generic metasearch engine architecture is shown in Figure 1.2. When the user submits a query to the metasearch engine, it selects a few underlying search engines to dispatch the query; when it

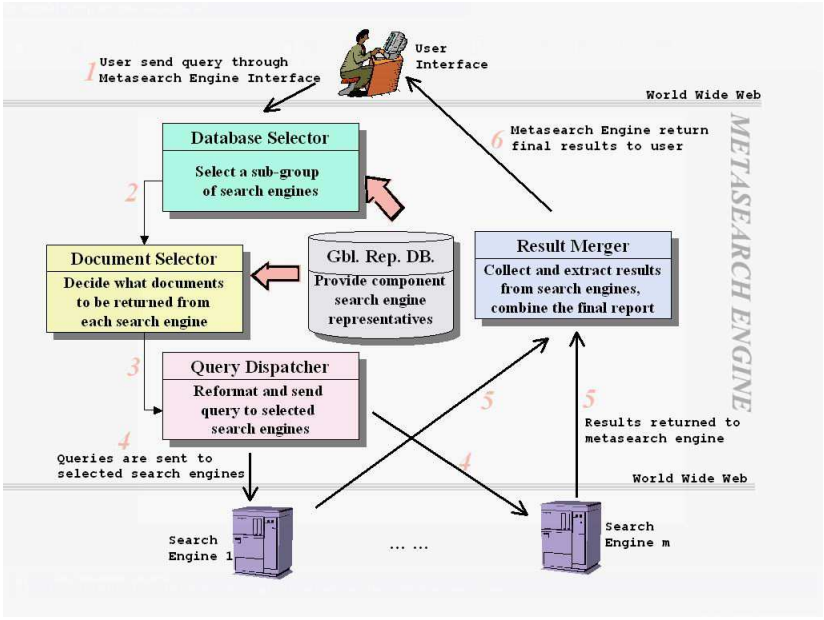

Figure 1.2: Metasearch engine reference software component architecture with the flow of query processing (Numbers associated with the arrows indicate the sequence of steps for processing a query)

receives the Web pages from its underlying search engines, it merges the results into a single ranked list and display them to the user.

1.8.2 Component Techniques For Metasearch Engines

Several techniques are applied to build efficient and effective metasearch engines. In particular, we introduce two important component technologies used in query processing: database selection and result merging.

Database Selection

When a metasearch engine receives a query from a user, the database selection mechanism is invoked (by the database selector in Figure 1.2) to select local search engines that are likely to contain useful Web pages for the query. To enable database selection, the database representative, which is some characteristic information representing the contents of the document database of each search engine, needs to be collected at the global representative database and made available to the selector. From all the underlying search engines, by comparing database representatives, metasearch engine can decide to select a few search engines that are most useful to the user query. Selection is especially important when the number of underlying search engines is large, as it is unnecessary, expensive and unrealistic to send query to many search engines for one user query.

Database selection techniques can be classified into three categories: Rough Representative, Statistical Representative, and Learning-based approaches [Meng, Yu, and Liu, 2002]. In the first approach, the representative of a database contains only a few selected key words or paragraphs; which are relatively easy to obtain, require little storage but are usually inadequate. These approaches are applied in WAIS [Kahle and Medlar, 1991] and other early systems. In the second approach, database representatives have detailed statistical information (such as document frequency

of each term) about the document databases, so they can represent databases more precisely than rough representatives. [Meng, Yu, and Liu, 2002] provides survey of several of this kind of approaches. In the learning-based approach, the representative is the historical knowledge indicating the past performance of the search engine with respect to different queries; it is then used to determine the usefulness of the search engine for new queries. Profusion (www.profusion.com) and SavvySearch (www.search.com), which are current leading metasearch engines, both fall into this category.

Result Merging

Result merging is the process in which, after dispatching a user query to multiple search engines and receiving results returned back from those search engines, a metasearch engine arranges results from different sources into a single ranked list to provide to users.

Ideally, merged results should be ranked in descending order of global similarities. However, the heterogeneities and autonomies of local search engines make the result merging problem difficult. One simple solution is to actually fetch all returned result documents and compute their global similarities in the metasearch engine (Inquirus) [Lawrence and Lee Giles, 1998]. However, since the process of fetching and analyzing all documents is computational expensive and time consuming, most result merging methods utilize the local similarities or local ranks of returned results to effect merging. For example, local similarities of results from different search engines can be re-normalized to a unified scale to be used as global ranking scores; For another example, if one document d is returned by multiple search engines, in a certain way, the global similarity of d can be calculated by combining its local similarities in search engines. These approaches, along with a few others, have been discussed in [Meng, Yu, and Liu, 2002].

1.9 IR Products and Resources

We use the phrase products and resources to refers to commercial and academic IR systems, and related resources. A good number of IR systems are available today, some are generic whereas others target a specific market — automotive, financial services, government agencies, and so on. In the recent years they are evolving toward being full-fledged, off-the-shelf product suites providing a full range of services — indexing, automatic categorization and classification of documents, collaborative filtering, graphical and natural language based query specification, query processing, relevance feedback, results ranking and presentation, user profiling and automated alerts, and support for multimedia data. Not every product provides all these services. Also, they differ in indexing models employed, algorithms for similarity computation, and types of queries supported.

Due to rapid advances in IR in the Web scenario, IR products are also evolving fast. These products primarily work with textual media in a distributed environment. Those that claim to handle other media such as audio, images, and video essentially convert the media to text. For example, broadcast television programs content is represented by the text of closed-captions. Video scene titles and captions are used as content descriptors of the former. Digital text of titles and captions is obtained by using OCR technology. The content of audio clips and sound tracks is represented by digital text, which is obtained by textual transcription of the media using speech recognition technology.

A survey of 23 vendors located in USA and Canada done in 1996 is presented in [Kuhns, 1996]. We also list a few important, well-known resources, which is by no means representative or comprehensive:

- TREC (trec.nist.gov): The purpose of Text REtrieval Conference is to support research within the information retrieval community by providing the infrastructure necessary for large-scale

evaluation of text retrieval methodologies. TREC provides large-scale test sets of documents, questions and relevance judgments. These testbeds enable performance evaluation of various approaches to IR in a standardized way.

- Lemur (www-2.cs.cmu.edu/∼lemur/): Lemur is a toolkit for Language Modeling and Information Retrieval.
- SearchTools.com (www.searchtools.com): Provides a list of tools that you can use to construct your own search engines.

1.10 Conclusions and Research Direction

In this chapter, we trace the evolution of theories, models and practice relevant to the development of IR systems in the contexts of both unstructured text collections and documents on the Web, which are semi-structured and hyperlinked. A retreival model usually refers to the techniques employed for similarity computation, and ranking the query output. Often, multiple retrieval models are based on the same indexing techniques and differ mainly in the approaches to similarity computation and output ranking. Given this context, we have described and discussed strengths and weaknesses of various retrieval models.

The following considerations apply when selecting a retrieval model for Web documents: computational requirements, retrieval effectiveness, and ease of incorporating relevance feedback. Computational requirements refer to both the disk space required for storing document representations as well as the time complexity of crawling, indexing and computing query-document similarities. Specifically, strict Boolean and fuzzy set models are preferred over vector space and p-norm models on the basis of lower computational requirements. However, from a retrieval effectiveness viewpoint, vector space and p-norm models are preferred over Boolean and fuzzy set models. Though the probabilistic model is based on a rigorous mathematical formulation, in typical situations where only a limited amount of relevance information is available, it is difficult to accurately estimate the needed model parameters. All models facilitate incorporating relevance feedback, though learning algorithms available in the context of Boolean models are too slow to be practical for use in real-time adaptive retrieval. Consequently, deterministic approaches for optimally deriving query weights, of the kind mentioned in section 1.5.2, offer the best promise for achieving effective and efficient adaptive retrieval in real-time.

More recently, a number of efforts are focusing on unified retrieval models that incorporate not only similarity computation and ranking aspects, but also document and query indexing issues. The investigations along these lines, which fall in the category of the language modeling approach, are highlighted section 1.4. Interest in methods for incorporating relevance in the context of the language modeling approach is growing rapidly. It is important to keep in mind that several interesting investigations have already been made in the past, even as early as two decodes ago, that can offer useful insight for future work on language modeling [Robertson et al., 1982; Jung and Raghavan, 1990; Wong and Yao, 1993; Yang, 1994].

Another promising direction of future research is to consider the use of the language modeling approach at other levels of document granularity. In other words, the earlier practice has been to apply indexing methods like $tf * idf$ and Okapi not only at the granularity of a collection for document retrieval, but also at the levels of a single document or mutiple search engines (i.e. multiple collections) for, respectively, passage retrieval or search engine selection. Following through with this analogy, suggests that the language modeling approach ought to be investigated with the goals of passage retrieval and search engine selection (the latter, of course, in the context of improving the effectiveness of metasearch engines).

In addition to (and, in some ways, as an alternative to) the use of relevance feedback for enhancing the effectiveness of retrieval systems, there have been several important advances in the direction of automated query expansion and concept-based retrieval. While some effective techniques have emerged, much room still exists for additional performance enhancements and more future research on how rule bases can be automatically be generated is warranted.

Among the most exciting advances, with respect to retrieval models for Web documents, is the development of methods for ranking web pages on the basis of analyzing the hyperlink structure of the Web. While early work ranked pages independently of a particular query, more recent research emphasizes techniques that derive topic-specific page ranking. Results in this area, while promising, still need to be more rigorously evaluated. It is also important to explore ways to enhance the efficiency of methods available for topic-specific page ranking.

Metasearch engine technologies are still far away from being mature. Scalability is still a big issue. It is an expensive and labor-intensive task to build and maintain a metasearch engine that searches on a few hundreds of search engines. Researches are carried on to solve problems such as automatically connecting to search engines, categorizing search engines, automatically and effectively extracting search engine representatives and so on. It is predictable that in the near future, metasearch engines can be built on hundreds of thousands of search engines. Web-searchable databases will become unique and effective tools to retrieve the Deep Web contents. The latter is estimated to be hundreds of times larger than the Surface Web contents [Bergman, 2000].

Other active IR research directions include Question Answering (QA), Text Categorization, Human Interaction, Topic Detection and Tracking (TDT), multimedia IR, Cross-lingual Retrieval. The Website of ACM Special Interest Group on Information Retrieval (www.sigir.org) is a good place to visit to know more about current research activities in the IR field.

Acknowledgements

: The authors would like to thank Kemal Efe, Jong Yoon, Ying Xie, and anonymous referees for their insight, constructive comments, and feedback. This research is supported by a grant from Louisiana State Governor's Information Technology Initiative (GITI).

References

I.J. Aalbersberg. Incremental relevance feedback. *Proceedings of the 15th Annual International ACM SIGIR Conference*, ACM Press, pp. 11-22, June 1992.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Reading, MA, 1999.

R. Belew. Adaptive information retrieval: using a connectionist representation to retrieve and learn about documents. *Proceedings of the 12th Annual International ACM SIGIR Conference*, ACM Press, New York, pp. 11-20, 1989.

A. Berger, J. Lafferty. Information retrieval as Statistical Translation. *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 222-229, 1999.

M. Bergman. The Deep Web: Surfacing the hidden value. *BrightPlanet* , available at www.completeplanet.com/Tutorials/DeepWeb/index.asp (Date of access: April 25, 2002).

J.N. Bhuyan, J.S. Deogun, and V.V. Raghavan. Algorithms for the boundary selection problem. *Algorithmica*, 17:133–161, 1997.

R. Bodner and F. Song. In Lecture Notes in Computer Science, Volume 1081, pp. 146–158. URL: http://citeseer.nj.nec.com/bodner96knowledgebased.html, 1996.

L. Bookman and W. Woods. Linguistic Knowledge Can Improve Information Retrieval. *http://acl.ldc.upenn.edu/A/A00/A00-1036.pdf*, (visited January 30th, 2003).

K.D. Bollacker, S. Lawrence, and C.L. Giles. CiteSeer: An autonomous Web agent for automatic retrieval and identification of interesting publications. *Proceedings of the Second International Conference on Autonomous Agents*, ACM Press, New York, pp. 116-123, May 1998.

A. Bookstein and D.R. Swanson. Probabilistic model for automatic indexing. *Journal of the American Society for Information Science*, 25(5):312-318, 1974.

S. Brin and L. Page. The Anatomy of a Large Scale Hypertextual Web Search Engine. In Proceedings of *WWW 7*/Computer Networks 30(1-7): 107-117, April 1998.

Y. Chang, I. Choi, J. Choi, M. Kim, and V. V. Raghavan. Conceptual retrieval based on feature clustering of documents. In *Proc. of ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval*, Tampere, Finland, August 2002.

W. B. Croft. Experiments with representation in a document retrieval system. *Information Technology*, 2:1–21, 1983.

W.B. Croft. Approaches to intelligent information retrieval. *Information Processing and Management*, 23(4):249-254, 1987.

W.B. Croft, T.J. Lucia, J. Cringean, and P. Willett. Retrieving documents by plausible study: an experimental study. *Information Processing and Management*, 25(6):599-614, 1989.

W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35:285–295, 1979.

J. S. Deogun, V. V. Raghavan, and P. Rhee. Formulation of the term refinement problem for user-oriented information retrieval. In *The Annual AI Systems in Government Conference*, pp. 72–78, Washington, D.C., March 1989.

EETimes. URL: www.eetimes.com, July 2003.

K. Efe, V. V. Raghavan, C. H. Chu, A. L. Broadwater, L. Bolelli, and S. Ertekin. The shape of the web and its implications for searching the web. In *International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, Proceedings at http://www.ssgrr.it/en/ssgrr2000/proceedings.htm. Rome, Italy, July-August, 2000.

E. Efthimiadis. User choices: a new yardstick for the evaluation of ranking algorithms for interactive query expansion. *Information Processing & Management*, 31(4):605-620, 1995.

C. Fellbaum. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

S. Gauch, G. Wang, and M. Gomez. ProFusion: Intelligent fusion from multiple, distributed search engines. *Journal of Universal Computer Science*, 2(9):637-649, 1996.

L. Gravano, C. Chang, H. Garcia-Molina, and A. Paepcke. Starts: Stanford proposal for Internet meta-searching. *ACM SIGMOD Conference*, Tucson, Arizona, ACM Press, pp. 207-218, 1997.

D. Haines and W. Bruce Croft. Relevance feedback and inference networks. *Proceedings of the 16^{th} Annual International ACM SIGIR Conference*, ACM Press, pp. 2-11, June 1993.

D. Harman. Relevance feedback revisited. *Proceedings of the 15^{th} Annual International ACM SIGIR Conference*, ACM Press, pp. 1-10, June 1992.

T. Haweliwala. Topic-Sensitive PageRank. *Proceedings of WWW2002*, May 2002.

R. Herbrich, T. Graepel, P. Bollmann-Sdorra, and K. Obermayer. Learning preference relations in IR. In *Proc. of the Workshop Text Categorization and Machine Learning, International Conference on Machine Learning-98*, pp. 80–84, March 1998.

A. Howe and D. Dreilinger. SavvySearch: A MetaSearch Engine that Learns Which Search Engines to Query. *AI Magazine*, 18(2):19–25, 1997.

G. S. Jung and V. V. Raghavan. Connectionist learning in constructing thesaurus-like knowledge structure. In *Working Notes of AAAI Symposium on Text-based Intelligent Systems*, pp. 123–127, Palo Alto, CA, March 1990.

S. Jones M. M. Hancock-Beaulieu S. E. Robertson, S. Walker and M. Gatford. Okapi at TREC-3. In *The Third Text REtrieval Conference (TREC-3)*, pp. 109–126, Gaithersburg, MD, April 1995.

B. Kahle and A. Medlar. An information system for corporate users: Wide area information servers. *Technical Report TMC199*, Thinking Machines Corporation, 1991.

M. Kim, F. Lu, and V. Raghavan. Automatic Construction of Rule-based Trees for Conceptual Retrieval. *SPIRE-2000*, pp. 153-161, 2000.

J. Kleinberg. Authoritatives Sources in a Hyperlinked Environment. In Proceedings of *ACM-SIAM Symposium on Discrete Algorithms*, pp. 668-677, January 1998.

R. Kuhns. A Survey of Information Retrieval Vendors. *Technical Report*: TR-96-56, Sun Microsystems, Santa Clara, CA, October 1996.

J. Lafferty and C. Zhai. Document Language Models, Query Models, and Risk Minimization for Information Retrieval, *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.111–119, 2001.

V. Lavrenko and W. Croft. Relevance-Based Language Models, *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 120–127, 2001.

S. Lawrence and C. Lee Giles. Inquirus, the NECI meta search engine. *Seventh International World Wide Web Conference*, Brisbane, Australia, pp. 95-105, 1998.

B. P. McCune, R. M. Tong, J. S. Dean, and D. G. Shapiro. RUBRIC: A system for Rule-Based Information Retrieval. *IEEE Transactions on Software Engineering*, 11(9):939-945, September, 1985.

W. Meng, C. Yu, and K. Liu. Building efficient and effective metasearch engines. *ACM Computing Surveys*, 34(1):48-84, 2002.

M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proc. of the 21st ACM SIGIR conference*, pp. 206–214, Melbourne, Australia, August 1998.

International Standards Organization (ISO). MPEG-7 Overview (version 8). ISO/IEC JTC1/SC29/WG11 N4980, July 2002. URLs: mpeg.tilab.com, www.mpeg-industry.com.

K. Nakata, A. Voss, M. Juhnke, and T. Kreifelts. Collaborative concept extraction from documents. *Proceedings of the 2^{nd} International Conference on Practical Aspects of Knowledge Management (PAKM 98)*, Basel, Switzerland, pp. 29-30, 1998.

J. Ponte and W. Croft. A language modeling approach to Information Retrieval, *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 275–281, 1998.

Y. Qiu and H. Frei. Concept based query expansion. *Proceedings of the 16th Annual International ACM SIGIR Conference*, ACM Press, pp. 160-170, June 1993.

T. Radecki. Fuzzy set theoretical approach to document retrieval. *Information Processing and Management*, 15:247-259, 1979.

V. Raghavan and S. K. M. Wong. A critical analysis of vector space model for information retrieval. *Journal of the American Society for Information Science*, 37(5):279-287, 1986.

P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pp. 448-453, 1995.

S. E. Robertson, M. E. Maron, and W. S. Cooper. Probability of relevance: A unification of two competing models for document retrieval. *Information Technology, Research & Development*, 1:1–21, 1982.

S. E. Robertson. Okapi. *http://citeseer.nj.nec.com/correct/390640*.

S. E. Robertson and K. Sparck-Jones. Relevance weighting of search terms. *Journal of American Society of Information Sciences*, pp. 129–146, 1976.

J. J. Rocchio and G. Salton. Information optimization and interactive retrieval techniques. In *Proc. of the AFIPS-Fall Joint Computer Conference 27 (Part I)*, pp. 293–305, 1965.

G. Salton, E. A. Fox, and H. Wu. Extended boolean information retrieval. *Communications of the ACM*, 36:1022–1036, 1983.

G. Salton. *Automatic Text Processing*. Addison-Wesley, Reading, MA, 1989.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.

E. Selberg and O. Etzioni. The MetaCrawler architecture for resource aggregation on the Web. *IEEE Expert*, 12(1):8-14, 1997.

C. E. Shannon. Prediction and entropy in printed English. *Bell Systems Journal*, 30(1):50-65, 1951.

K. Sparck Jones and J. I. Tait. Automatic search term variant generation. *Journal of Documentation*, 40:50-66, 1984.

N. Tadayon and V. V. Raghavan. Improving perceptron convergence algorithm for retrieval systems. *Journal of the ACM*, 20(11-13):1331–1336, 1999.

C. J. Van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33:106–119, June 1977.

H. White and K. McCain. Bibliometrics. In *Annual review of Information Science and Technology*, Elsevier, pp. 119-186, 1989.

W. Wong and A. Fu. Incremental document clustering for web page classification. *IEEE 2000 Int. Conference on Information Society in the 21st Century: Emerging Technologies and New Challenges* (IS 2000), pp. 5-8, 2000.

S. K. M. Wong and Y. Y. Yao. Query Formulation in Linear Retrieval Models. *Journal of the American Society for Information Science*, 41:334–341, 1990.

S. K. M. Wong and Y. Y. Yao. A probabilistic method for computing term-by-term relationships. *Journal of the American Society for Information Science*, 44(8):431–439, 1993.

J. Xu and W. Croft. Query exapnsion using local and global document analysis. *Proceedings of* 19^{th} *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 4–11, 1996.

Y. Yang and C. G. Chute. An Example-based Mapping Method for Text Categorization and Retrieval. *ACM Transactions on Information Systems*, 12:252–277, 1994.

C. T. Yu. A Formal Construction of Term Classes. *Journal of the ACM*, 22:17–37, 1975.

C. T. Yu and G. Salton. Precision Weighing – An Effective Automatic Indexing Method. *Journal of the ACM*, pp. 76–88, 1976.

C. T. Yu, C. Buckley, K. Lam, and G. Salton. A Generalized Term Dependence Model in Information Retrieval. *Information Technology, Research & Development*, 2:129–154, 1983.

C. Yu and W. Meng. Web Search Technology. In *The Internet Encyclopedia*, Editor: H. Bidgoli, Wiley Publishers (to appear), 2003.

C. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Infromation Retrieval, *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 334–342, 2001.