Invading the Fortress: How to Besiege Reinforced Information Bunkers

Jeroen Hoppenbrouwers, Hans Paijmans
Tilburg University/Infolab
PO Box 90153, NL-5000 LE Tilburg, The Netherlands
hoppie@kub.nl, paai@kub.nl

Abstract

Information retrieval (IR) research has been very active over the last decades to develop approaches that allow machine indexing to significantly improve indexing practice in libraries. However, due to practical limitations, this technology is not often used in large-scale libraries. We propose a mix of existing technologies and new ideas that enable traditional libraries to adopt modern IR technology and offer improved services to their customers, while leveraging their existing infrastructure and legacy databases.

Keywords*: Information retrieval, Decomate, legacy systems, bibliographical database, meta-data.*

1 Introduction

Despite decades of research in information retrieval (IR), few actual implementations of the results of this research in large-scale libraries have come into existence. Partially this is due to the discrepancies of researchers' assumptions and the reality (e.g., users are not willing to spend more than a few moments on query formulation), partially due to conservative library organizations that do not easily adopt new technologies, and partially due to practical restrictions.

One of the most noticeable restrictions met in practice is the fact that many existing library databases are the result of significant work, usually over several decades, and are heavily optimized for the standard types of queries that bibliographical reference systems should answer. These databases are not open to experiments, cannot easily be adapted to new environments, and usually do not even contain the type of material that current IR research is targeting at. In many senses, these databases are *legacy systems*: systems that significantly resist modification and evolution to meet new and constantly changing business requirements – but that at the same time are absolutely essential for the primary process of the library.

Although modern library databases begin to contain much more material, such as abstracts or even full text of resources in machine-readable form, the fact that most library databases are still plain bibliographical information systems cannot be ignored. Even if full text is available, that does not mean that it can be used for any other purpose than just display to the end user.

What is required is a way to open up these traditional (but sometimes technically very modern) bibliographical databases so that they can be augmented by advanced IR systems, plus a way to integrate the IR system's output with the current interface of the library. In this way, the existing system can be maintained without any disruption of service, while more advanced services can be gradually introduced as they become available. We view this evolutionary instead of revolutionary approach as the only viable way of introducing IR techniques to large existing libraries.

This paper describes an example of such a hybrid system, currently in production at Tilburg University and several other libraries in Europe. It outlines the basic architecture of the system, the currently available modules, the new ways of accessing traditionally organized information it enables, several types of advanced IR techniques that could successfully be implemented on top of the now available information, and proposals on how to integrate the results of these advanced IR techniques with the standard bibliographical record list output.

2 Decomate

Several university libraries in Europe (Universitat Autònoma de Barcelona, London School of Economics and Political Science, European University Institute near Florence, and Tilburg University) have teamed up to create the European Digital Library for Economics [20]. Partially sponsored by the European Union through the Decomate-II Project,[1] they have linked up their individual library

[1] LIB-5672/B, `http://www.bib.uab.es/decomate2`

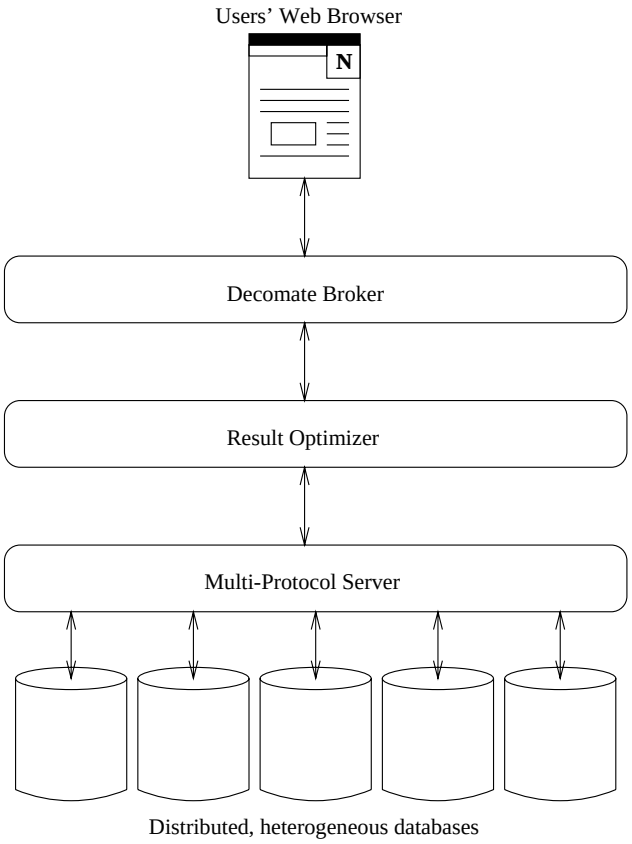

Figure 1. The Decomate Architecture

databases and provide one large virtual library to their users. By means of a common interface (tuned to local preferences and language), all underlying databases are queried simultaneously, the results are merged and de-duplicated, and the final result set is presented to the user in order of relevance to the query. Lastly, users can retrieve a significant part of the journal collection of all four participating libraries in full text thanks to the participation of several publishing companies in the project, noticeably Elsevier Science Publishers, Swets & Seitlinger, Kluwer Academic Publishers and full project partner SilverPlatter Information Ltd.

2.1 Decomate Architecture

The Decomate architecture emphasizes (library) standards wherever possible. Most library (bibliographical) databases use the Z39.50 access protocol. A *Multi-Protocol Server* (MPS) maps differences between protocols (not necessarily Z39.50 at all) into one, Z39.50-based canonical protocol. The MPS also takes care of the parallelization of queries to all databases. Through the canonical protocol, the MPS can be treated as a single Z39.50-like database that is divided in several sections. At the user end, a *Broker* converts the Z39.50-like canonical protocol into standard HTML and adds session awareness. The Broker in fact decides exactly what the local implementation of the user interface and the whole library application looks and feels like. Both the MPS and the Broker are the same software for all participating libraries. However, especially the Broker is extremely configurable through a dedicated language. In the Decomate project, each library maintains its own Broker configuration, but the MPSes are generic (except for the connected databases) and all modules speak the same canonical protocol in XML. This means that from any Broker (interface), all MPSes can be reached, and multiple MPSes can be chained if required.

When offering full text[2] to users, there must be an authentication and authorization mechanism in place, plus a thorough logging system, since publishers require these to block public access to their copyrighted material and bill users in case of pay-per-view access. Decomate includes all these features, which was one of the reasons the publishers joined the project team.

Another major improvement over existing systems is the Current Awareness Server that can compile new additions to the library (usually journal articles) according to per-user interest profiles. The CAS offers these compilations in the form of 'personal journals,' with regular monthly issues that can be called up like any other journal issue, plus E-mail notification if required.

Lastly, all current Decomate systems include a Document Server which gives users instant access to the full text versions of many journals. The Document Server basically is just a PDF file store; specifically, it does not add any form of full text indexing.

2.2 Decomate Advanced Access

Next to the basic parts, extra modules can be easily added to the system. For example, the optional *Result Optimizer* [11] is a module that sits in between a Broker and a MPS. Its task is to convert incoming MPS multiple result sets from multiple databases into one large result set, grouping duplicate records together in the process. The Result Optimizer never changes or removes information, it only collates it. When enough additional information is available, the Result Optimizer can also reorder the result set (the bibliographical records). Standard rankings are on author and on publication date, but the Result Optimizer also has the possibility to do relevance ranking.

For proper relevance ranking an extensive query is preferred. Relevance ranking on the basis of a simple Boolean query is not often very successful. Since most current li-

[2]Full text in this paper means: the unabridged, formatted article exactly as it has been published, usually in PostScript or PDF, but possibly in scanned TIFF.

Figure 2. Prototype Concept Browser

brary users limit themselves to one or two keywords per query, a habit that got even worse with the recent flood of simplified Internet search engines, any attempt of relevance ranking (pure Boolean) result sets was indeed considered to be fruitless.

For this reason we introduced the *Concept Browser* [10, 11], a two-dimensional browsing interface that visualizes the existing library thesauri that have been used for (manually) indexing the databases. Instead of bypassing the thesaurus completely (what users do when confronted with a free keyword search engine), they are now offered an attractive visual environment in which they can explore the available 'proper' keywords. Besides compiling queries that contain only correct keywords, this thesaurus-driven approach makes the users more aware of the underlying knowledge structure of the databases. They spend more time putting a query together, but get better results straight away. As a bonus the Concept Browser offers users the option to *order the keywords in terms of perceived relevance to their information need*. This ordering information is used by the Result Optimizer to rank the resulting bibliographical records according to perceived relevance.

Note that the whole Advanced Access system (Result Optimizer plus Concept Browser) is based on the idea of *added value*. Users can still bypass both modules, or ignore their suggestions and just go on using the traditional unrestricted Boolean keyword matching. This allows people to gradually adjust to the new mechanisms, and also motivates system designers to pay due attention to true user-friendliness and real added value. It offers us the opportunity to measure the impact of system changes and emphasize approaches that have a significant effect on the effectiveness of the system as a whole.

3 Extending Traditional Search Mechanisms

Although the Decomate system allows for addition of basically any module anywhere in the pipeline, it still is based on the existing, traditional bibliographical databases which usually do not offer more than extended Boolean keyword searching on bibliographical records. Actions like relevance ranking or query expansion must be done by generating extra information at the user end and post-processing the Boolean output of the databases with this extra information.

3.1 Modern Full Text Indexing

Simply replacing the Boolean database engines with more modern engines based on document vectors is no viable option. First, the Boolean retrieval engines are highly tuned and cope easily with very high volumes of traffic (thousands of simultaneous users and billions of records), while maintaining response times under a second. More advanced IR systems usually cannot offer this type of performance yet on the same hardware [13].

Second, most IR techniques require as much raw data as possible, preferably the full text of every resource in the system. But current library databases only store bibliographical data, condensed meta-data *about* the resource, not the resource itself.

In fact, this condensation process may include two separate activities: first a *document surrogate* is created or selected from the document, possibly an abstract or even just title and author. This document surrogate is then processed by the indexing system to become the internal *document representation*. Often this is is just a set of keywords extracted from the document surrogate. To complete the cycle, we have the *on-line document* that is presented to the user as the final result of his query. Often, the on-line document is very similar to the document surrogate. When we refer to *meta-data*, we will mean these three abstractions of the original document.

Although some of the new material added to current library databases is available in full text, the majority of the material already in the databases, in the form of the document representation and the on-line document, is not and likely never will be.[3]

Interestingly, leading-edge (digital) libraries are already *moving away again* from storing full text, leaving the full text warehousing to the authors or publishers and concentrating on resource description only. A digital library becomes more and more a *guide to information*, since the Internet provides instant access to virtually all resources. We think it is not a good idea to keep pulling all new available

[3] Efforts in digitizing resources, such as by scanning, do not necessarily deliver machine-readable full text.

information into the library just to create a local full-text index.[4]

3.2 Distributed Meta-data

The complications outlined above lead to the conclusion that systems and people alike will have to more and more rely on distributed meta-data to initially separate huge collections of documents into interesting and non-interesting clusters, possibly followed by more fine-grained processing of the individual documents [16]. Note that meta-data is not necessarily restricted to the traditional bibliographical meta-data; it can also be enriched with 'automatic extracts' coming from IR engines. Several types of these machine-readable extracts could be made available, ranging from sparse and coarse but very efficient to highly detailed ones. They would be produced in a standard way by the publisher of the document collection (increasingly often the author himself) and offered together with the full text, also in a standard way.

There are several initiatives such as the Dublin Core [19] to generalize and standardize meta-data and to ensure their exchangeability.[5] Also efforts have been undertaken to build indexes in a distributed way, so that raw data shipment is minimized [3]. What to our knowledge is not yet available is a standard of distributed machine-readable meta-data. We see no reason why the Dublin Core could not accommodate standard fields for machine-readable data, and why Dublin Core meta-data could not be distributed in a network. However, this is not to say that such a solution is straightforward [14, 15].

We think that distributed meta-data, partially acquired by automated full-text indexing, might offer a solution to the information warehousing and retrieval problem. However, such an approach cannot be taken by a single isolated library, and no library will abandon its existing databases overnight. Therefore we propose to take the middle road, and to build advanced IR engines *next to* the existing Boolean engines. These new engines should be able to tap into full resources only when available, and otherwise to use limited meta-data ranging from a bibliographical record to machine-readable 'extracts.' They also have an important task in generating meta-data from their local resources to be used by other, remote engines.

The challenge is to combine the old and new engines into the same system, with one common user interface, and while re-using existing resources such as thesauri and library catalogs as much as possible. Taking the Decomate system as a reference, the next sections will present an overview of current best practice in IR research and suggest places in Decomate where IR techniques could play an important role.

[4] Some libraries might still be interested in maintaining full local copies of certain collections, like national libraries that have a depot function.

[5] `http://purl.org/dc`

4 Sneaking in the Back Door

In the Decomate environment, retrieval that is based on the availability of the indexed full text of the documents is not feasible, and creation and maintenance of the document-keyword matrices necessary for traditional implementations of statistical models is not an option either. The main reason is that many Decomate databases are not under control of the library giving access to it, and therefore difficult to index. On top, many databases are bibliographical in nature and do not contain (indexable) full text.

Nonetheless, the frequency-based retrieval models have a long track record that is *at least* as good as the best manual indexing systems (see for instance the conclusions of [12] or [4]) and only the conservative attitude of the management in most libraries has barred more general use of these models [1]. If there is any opportunity to introduce at least the option to use such techniques somewhere in the Decomate structure, it should be grasped. This section concentrates on finding appropriate approaches in information retrieval to suit the Decomate model.

Essentially, frequency-based retrieval models consist of two actions: the weighing of the keywords and the comparing of the document representations. We will ignore the various methods of comparing the document representations for the moment and concentrate on the creation of such representations in the form of vectors. Intuitively, it is clear that not all words in a document have the same importance or weight. In this section, we will give short examples of successful attempts to compute the weight of a keyword on the basis of its frequency in both the document and the collection, and how we may capture at least a part of this effectiveness for our proposed system.

In almost all existent models, the combined properties that form the weight of the keyword or term i are the three frequency figures *term frequency* (tf_{ik}), *document frequency* (df_i) and *collection frequency* (cf_i), being respectively the frequency of term i in the document k, the number of documents in which the term i occurs and the total number of times that the term i occurs in the collection. N is generally reserved to represent the number of documents in the database.

Word weights come in two 'flavors': one in which the weight is related only to the keyword itself, so that it is the same for all occurrences of a keyword in the collection, and one in which the properties of individual documents are also taken into consideration: the *plain word weights* and the *word-document weights* [17].

In the next section we will consider two examples of

plain word weights: the Poisson models and the discrimination value model, followed by the most popular of the word-document weights: the $tf.idf$ weight. We will also describe the basics of the Latent Semantic Indexing approach; although this approach does not result in weights for the individual keywords, it is based on frequency data and would be a candidate technique for our library system.

4.1 Plain Word Weights

Poisson Models Perhaps the simplest scheme by which to weigh the usability of a word as a keyword, i.e., a word that by its occurrence separates the body of documents into two separate groups, relative to an information need, is its deviation from the Poisson distribution. This distribution describes the probability that a certain random event occurs a certain number of times over units of fixed size. The Poisson distribution applies to terms in documents if the probability of an occurrence of that term in a piece of text is proportional to the length of that text and if the occurrence of terms is independent from previous or subsequent occurrences. This latter assumption holds for function words and does not hold for content words. Therefore, the Poisson or its derivates may be used to discriminate between 'important' and 'less important' words ([2], [9]).

The Discrimination Value Another, computationally rather expensive method is the computation of the discrimination value of a term, which is the influence that a term has on the mutual similarity of the documents. The documents are viewed as a cloud of dots in a space. Keywords that represent the documents influence the density of the cloud: 'good' keywords bring similar documents closer to each other and farther away from dissimilar documents. The discrimination value of a keyword Dv_i is computed by comparing the average density Q of the document cloud in which the keyword i is part of the document vector, with the average density Q_i of the cloud *without* keyword i:

$$Dv_i = Q - Q_i$$

If the database is represented as a term-document matrix with documents as rows of M distinct terms $t_1, t_2, ..., t_M$, Q is computed by taking the average $(N(N-1))$ pair-wise similarity values of all possible document pairs:

$$Q = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{k=1, i \neq k}^{N} sim(D_i, D_k)$$

where N is as usual the number of documents and D_i and D_k are documents.

There is a variety of techniques with which to compute the similarity of document vectors; for an overview, see e.g. [21]. The most commonly used method involves the cosine function, which is also used in the experiments of [22], [8] and [5].

The $tf.idf$ Family of Weights There is a second family of weighing schemes that uses the frequency of the words within documents and their distribution over the database as a measure for the suitability of a word as a keyword for a particular document. The most popular of these schemes is the so-called $tf.idf$ weight, or rather *one* of the $tf.idf$-related weights, as there are several variations. The $tf.idf$ is composed of the term frequency (tf) and the inverse document frequency (idf) or one of its derivates or normalizations. An appropriate indication of a term as a document discriminator can be computed by taking an inverse function of the document frequency of that term, e.g. $idf = N/df_t$, for N documents, or $idf = logN/df_t + 1$. The product of the term frequency and the inverse document frequency, $tf.idf$, may then be used as an indicator of the importance of a term in a document.

4.2 Feature Reduction Models

In the methods described above, the features created by the translation from the document into the document representation in the index language remain intact. After the application of filters and different weighing methods, a number of features are selected to represent the document. A different method is the re-mapping of the original features on a smaller number of new features. Here we like to use the expression 'feature *reduction*' or 'transformation'. A method of feature reduction that has received much attention is *latent semantic indexing* [6]. This reduction is brought about by applying singular value decomposition (SVD) to the original document-keyword matrix, creating a new semantic space in which both documents and keywords can be mapped. If the relation between each keyword and each document is expressed in a $d : t$ matrix of weights (w), where d is the number of documents and t the number of terms, the application of SVD creates three new matrices; a $d : s$ matrix (W), a diagonal $s : s$ matrix (S) and a $s : t$ matrix (T).

$$\begin{bmatrix} w_{0,0} \ldots\ldots w_{0,t} \\ \ldots \\ \ldots \\ \ldots \\ w_{d,0} \ldots\ldots w_{d,t} \end{bmatrix} =$$

$$\begin{bmatrix} W_{0,0} \ldots W_{0,s} \\ \ldots \\ \ldots \\ \ldots \\ W_{d,0} \ldots W_{d,s} \end{bmatrix} \begin{bmatrix} S_{0,0} \ldots \\ \ddots \\ \ldots S_{s,s} \end{bmatrix} \begin{bmatrix} T_{0,0} \ldots\ldots T_{0,t} \\ \ldots \\ \ldots \\ T_{s,0} \ldots\ldots T_{s,t} \end{bmatrix}$$

This new s-dimensional space describes the co-occurrence of the original keywords and the diagonal matrix S is ordered in such a way that the upper left elements describe strong co-occurrence tendencies of documents when expressed in keywords and vice-versa. Towards the right lower part of the diagonal, only spurious co-occurrences and weak relations occur. By keeping the n first singular values and zeroing out the others, a semantic space can be defined in which to compare documents, keywords or combinations of both. In the TREC proceedings, very good results have been reported by Dumais, using LSI in combination with $tf.idf$ weighing [7].

4.3 Application in Complete Databases

As we already indicated, the models described above have never been very popular in library production environments [18], although Blair [1] seems to suggest that a certain conservativeness of the managing staff may have something to do with it. And of course, when applied to real full text indexing of complete documents, the sheer size of the document-keyword matrix of a typical library may become prohibitive. In the context of Decomate, it may be possible to confine ourselves to the smaller document surrogates of the Dublin Core, or MARC, or other bibliographical formats. As long as the statistical integrity of the database is not compromised, advanced term weighing and querying according to the Vector Space Model could be offered as an alternative to the Boolean queries. However, an absolute prerequisite for these models is that the database on which they are applied, is complete, or that a subset is chosen, which statistically is representative for the frequency distributions of the database.

4.4 Application in Incomplete Databases

If a document-keyword database with frequency data, as described above cannot be obtained, we still may build an incremental database of the bibliographic material that came with earlier queries. Such a database in no way satisfies as a description of the underlying databases. However, it may still contain a good description of the interests of the local users of the general databases, because the accumulated documents are the documents that have been returned as potentially relevant for some query or queries in the past.

It is important to realize that when such a database of earlier retrieved documents is created, using it is not just a matter of copying the traditional frequency-based retrieval tools. There are three reasons why this cannot immediately be done:

1. To start with, the documents are not necessarily unique, because some documents may have turned up several times as an answer to different queries and this should be taken into account.
2. Then again many of these documents are false drops and it is difficult to detect those without active cooperation of the users, which cannot be counted upon.
3. And of course the normal assumptions on the distribution of keywords do not apply: good keywords should be relatively rare in the database, but here the documents are selected because the good keywords do in fact occur in them.

In an incomplete database, this makes it all but impossible to implement variations on the first three frequency-based models that we mentioned, because the weights that are computed are only valid when interpreted relative to a complete database. The same is not true for the Latent Semantic Indexing model. As the name implies, this model aims at grouping the keywords (or documents) according to hidden semantic relations that are extracted from the co-occurrence patterns of the keywords in the documents. Such co-occurrence patterns make sense, especially when the documents concerned are biased towards some user or user group. An added bonus is that such incomplete databases are much smaller than the original databases – a very important property in the light of the heavy processing needed for creating such LSI tables.

4.5 Application to Decomate

The technique described in the previous paragraphs can be fitted into the existing Decomate architecture as indicated in Figure 3: either as a module that acts as a full indexing and retrieval system parallel to the existing indices (left) or as a module that only operates on the information that has been collected as the result of earlier queries. In this latter case, we have a special case of relevance feedback.

4.5.1 Parallel Indexing with Frequency Information

The frequency-based weighing schemes outlined in the previous section lend themselves to keyword ranking in case a user does not want to specify a preferred ranking, or has no idea how to weigh keywords in the first place. Although it can be argued that plain Boolean queries with only certified thesaurus terms by definition produce an unordered set and that any relevance ranking would be arbitrary, current practice shows that most systems do offer some form of ranking. A common ranking is by reversed system entry date, with the ‘newest’ publication on top. We interpret this as a wish to have even Boolean queries ranked by some sort of relevance to the user. Any extra help in this area is welcome, and advanced algorithms that use word frequencies offer opportunities.

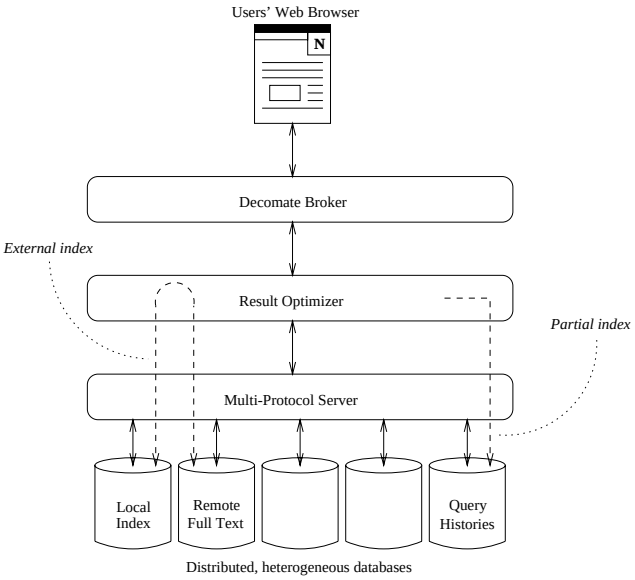

Figure 3. The Proposed Decomate Architecture Extension

Frequency-based algorithms may also be helpful to select candidate keywords to start searching in other databases, both before and after the initial query. In this way they could suggest some kind of auto-thesaurus of keywords that are known to have a good discrimination value. Although care must be taken when presenting such thesauri to the user, because of the sometimes counter-intuitive results, such an auto-thesaurus with links between terms that often co-occur might be a handy tool for initial term suggestion to users.

In a system like Decomate, frequency distributions could be calculated periodically for all connected databases. This can be done off-line on the raw datasets of the bibliographical records, or in some cases on the full text. If no other way is feasible, the normal query mechanism could be used to extract all records from the databases, but a lower-level access is preferred for performance reasons. Since statistics do not change significantly when documents are added, they need not be recalculated daily or even weekly, saving much effort. The statistical information then could be offered either as an add-on to the existing database, using a new field in the Z39.50 protocol, or through a specialized statistics database that needs to be included in the MPS database pool. These additions would not in any way affect the existing library system, but they provide modules such as the Result Optimizer with an extra source of valuable information. It would be very easy to add experimental versions of the Result Optimizer to the system and let selected groups of users try them out, or have users select which version they prefer. As long as Decomate modules do not crash or downgrade the system in any other way, and the default choices are safe and familiar, offering more choice in anything should not be any problem. On the contrary: if care is taken with lay-out and interfacing, alternative weighing and query modules may significantly add to the general value of the system.

4.5.2 Relevance Feedback

As we have stated, it is also possible to concentrate on the data that already have been retrieved for earlier queries and to use this for relevance feedback in combination with Latent Semantic Indexing. Of course, both techniques may be used on the basis of *all* information in the system, but especially in the case of LSI, this may prove to be too expensive in terms of processing. But as LSI actually offers information on the co-occurrence of terms, it does not have to operate on a correct statistical sample of the complete database and may be used also on the subset of the database that has been *ipso facto* marked as interesting by inclusion in the result of a query.

As with the frequency distributions, reduced feature representations can be calculated at the source and distributed through the existing MPS, or a third party can do the work, using off-peak hours for raw data access. The features cannot be used for presentation to humans, because the 'human' keywords have disappeared. The only informational structure that is available to the user is the on-line document; by indicating his preferences, other documents with similar LSI representations can be collected.

This means that frequency-based algorithms will not be used by the Result Optimizer, but by the database back end. These feature-based engines therefore replace traditional engines, but since they can be offered alongside each other while keeping the same raw data set, they are an extension and not a true replacement. Besides, for queries such as *'all documents from author X'*, the traditional engines still excel.

4.6 User Presentation

Care must be taken in how to present these alternative database engines to users. Since the underlying raw collection is the same, users might get confused if they get two dissimilar results in reply to the same query. The result set merging and de-duplication service offered by the Result Optimizer can help, but it does not seem a very good idea to always have both engines run the same query in parallel.

We suggest to present the new engines in the beginning as 'experimental' and to provide a good, concise explanation about their nature. In this way the system can offer 'hard' and 'soft' search engines, so that the users can select the engine they like most for specific tasks. With a simple 'try the other engine'-button on search results screens

or a list of just additional documents that 'the other engine' found, it can be easily made clear to users what the performance of both systems is, relative to each other. Current practice in user interfaces is to give systems a 'face' or 'persona' in the form of a cartoon or other image, which might help users intuitively select the most appropriate system for their current information need. Modern (Internet) users are accustomed to all kinds of graphical abuse and environments that fundamentally change overnight, which makes such an approach feasible even for quite traditionally oriented libraries.

5 Conclusion

A modular, distributed library system such as Decomate allows for much easier addition of experimental modules than a traditional monolithic system. Separation of raw data storage, search engines, result set optimizers, and user interfaces leads to a flexible architecture that scales well and can be adapted to changing situations without abandoning legacy systems.

Within such a modular framework, advanced IR engines based on statistics can gradually take over from traditional Boolean engines. However, for most statistics the availability of full text, or at least a document representation that consists of keywords, is mandatory, and the more data available, the wider the range of statistical methods that can be applied.

Full text is increasingly common thanks to developments in technology and business processes, but advanced digital libraries might opt not to store all full text locally. Remote full text asks for a different approach to indexing, based on distributed meta-data and feature reduction.

IR techniques suitable for inclusion in Decomate-like full production library systems are already available. They need to be adapted to the particular situation in many libraries, where meta-data is the norm but standards for meta-data still are in their infancy. Using the Decomate approach, advanced IR can help libraries to bridge the current gaps without negatively influencing existing systems.

5.1 Future Research

Work needs to be done in the area of IR techniques to select appropriate approaches for the current needs. These approaches should, if possible, perform at least as good as Boolean retrieval on plain bibliographical data and perform better when the available material increases (abstracts, extended abstracts, reduced documents, full text). Subsequently the selected approaches must be implemented in robust modules and integrated in a production system to see whether the users actually *perceive* any improvement.

Next to this, we need standards in the area of meta-data that include machine-readable reduced documents in various depths, preferably continuous from a single keyword to full text. On top, reliable and scalable architectures to distribute this information over networks must become available to exploit the promises of network information technology.

Lastly, we plan to look into ways of suggesting extensions of existing thesauri, still the mainstay of library knowledge representation. A good thesaurus is of great help to searchers, but it must be up to date and accessible. IR techniques may be able to improve both aspects.

References

[1] D. C. Blair. STAIRS redux: thoughts on the STAIRS evaluation, ten years after. *Journal of the American Society for Information Science*, 47(1):4–22, January 1996.

[2] A. Bookstein and D. R. Swanson. A decision theoretic foundation for indexing. *JASIS*, 26(1):45–50, 1975.

[3] C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, and M. F. Schwartz. The harvest information discovery and access system. In *Proceedings of the Second International WWW Conference '94: Mosaic and the Web*, 1994.

[4] C. W. Cleverdon. The significance of the cranfield tests. In A. Bookstein, Y. Chiaramella, G. Salton, and V. V. Raghavan, editors, *SIGIR '91; Proceedings of the 14th international conference on research and development in information retrieval*, pages 3–12. Association for Computing Machinery, 1991.

[5] C. Crouch. An analysis of approximate versus exact discrimination values. *Information Processing and Management*, 24(1):5–16, 1988.

[6] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, pages 391–407, 1990.

[7] S. Dumais. Latent semantic indexing (lsi): Trec-3 report. In *Overview of the Third Text Retrieval Conference (TREC-3)*, pages 219–230, Gaithersburg, Maryland, November 1994.

[8] A. El-Hamdouchi and P. Willett. An improved algorithm for the calculation of exact term discrimination values. *Information Processing and Management*, 24(1):17–22, 1988.

[9] S. Harter. A probabilistic approach to automated keyword indexing: Part ii. an algorithm for probabilistic indexing. *JASIS*, 26(4):280–289, 1975.

[10] J. Hoppenbrouwers. Advanced Conceptual Network Usage in Library Database Queries. Technical report, Infolab, Tilburg University, 1998. `http://infolab.kub.nl/people/hoppie`.

[11] J. Hoppenbrouwers. Analysis and Design Advanced Access Functionality Decomate-II. Technical report, Infolab, Tilburg University, 1998. `http://infolab.kub.nl/people/hoppie`.

[12] M. Keen. Term position ranking: some new test results. In N. Belkin, editor, *SIGIR '92; Proceedings of the 15th international conference on research and development in infor-*

mation retrieval, pages 66–75. New York, ACM press - 353 pp., 1992.

[13] Michael Lesk. “Real World” Searching Panel at SIGIR ’97. *SIGIR Forum*, 32(1):1–4, 1998.

[14] Mike Papazoglou and Jeroen Hoppenbrouwers. Contextualizing the Information Space in Federated Digital Libraries. *SIGMOD Record*, 28(1), Mar. 1999.

[15] Mike Papazoglou and Jeroen Hoppenbrouwers. Knowledge Navigation in Networked Digital Libraries. In *Lecture Notes in Computer Science, LNAI 1621*, volume 28, pages 40–46. Springer Verlag, Mar. 1999.

[16] M.P. Papazoglou and H. Weigand and S. Milliner. TopiCA: A Semantic Framework for Landscaping the Information Space in Federated Digital Libraries. In *DS-7: 7th Int’l Conf. on Data Semantics*, pages 301–328. Chapman & Hall, Leysin, Switzerland, Oct. 1997.

[17] J. Paijmans. *Explorations in the document vector model of information retrieval*. PhD thesis, Tilburg University, 1999.

[18] J. J. Paijmans. Some alternatives for the boolean model in information retrieval. In *TICER Summer school*, 1996.

[19] S. Weibel and J. Goldby and E. Miller. OCLC/NCSA Meta-Data Workshop Report.

[20] Thomas Place. Developing a European Digital Library for Economics: the DECOMATE II project. *Serials*, 12(2):119–124, July 1999.

[21] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, sec. edition. 208 pp., 1979.

[22] P. Willett. An algorithm for the calculation of exact term discrimination values. *Information Processing and Management*, 21(3):225–232, 1985.