CIRQUID: Complex Information Retrieval QUeries In a Database

Djoerd Hiemstra[1], Arjen P. de Vries[2], Henk Ernst Blok[1],
Maurice van Keulen[1], Willem Jonker[1] and Martin L. Kersten[2]

[1]University of Twente
PO Box 217, 7500 AE Enschede
The Netherlands

[2]CWI
PO Box 94079, 1090 GB Amsterdam
The Netherlands

Abstract

The CIRQUID project plans to design and build a DBMS that seemlessly integrates relevance-oriented querying of semi-structured data (XML) with traditional querying of this data. The project is funded by the Netherlands Organisation of Scientific Research (NWO project number 612.061.210).

1 Introduction

The World-Wide Web has boosted the development of data management techniques for semi-structured textual data, emphasizing content and structure. On the one hand, content-based (ranked) retrieval of textual documents has become widely accepted by internet search engines like Google and AltaVista. On the other hand, XML has become the standard document model allowing structure and (text) content to be represented in a combined way. Currently, much attention is paid to querying such documents. The development concerning XML query languages progresses towards combined querying of both the content and structure elements. Query processing technology has not fully kept pace with this development.

Current XML Query languages like XQuery lack one of the most important requirements of content-based retrieval: The relevance-oriented ranking of retrieval results. Ranking is of the utmost importance if large textual collections are queried. The average text query on the web results in thousands of documents that contain the words. Ranking algorithms sort retrieved sets of thousands of documents in decreasing order of the estimated probability of relevance, saving the user valuable time in examining the retrieved set in search for the relevant information.

As an example, consider the following simple query: *Find the* 10 *most relevant news items for which an image caption and the body text are about "Balcony scene of royal wedding".* An information retrieval system would typically treat a news item as one textual object, so such a system would not take advantage of the structural conditions in the query. On the other hand, the query could be easily expressed in XQuery, except for "the 10 most relevant" part, which suggests a ranking of the retrieved results. What is needed is a system that is able to do both efficiently.

In this project, we will develop ranked retrieval algorithms for semi-structured data by focusing on improving processing support for combined querying of both structure and content. In particular, we restrict ourselves to retrieval of text content.

2 Problem statement

Until recently, text content for ranked retrieval was usually represented as a 'bag of words', and a 'bag of words' is all that can be handled with well-known theoretical information retrieval models like Salton's vector space model [48] or Robertson et al.'s probabilistic model [46]. However, the recently proposed language modeling approach to information retrieval (LMIR) [44, 26] suggests some exciting new ways to look at texts that go beyond the 'bag of words' representations, i.e., by combining content representations in a non-trivial way. Language models are an emerging and promising direction in information retrieval research: two entire sessions were devoted to this topic at the SIGIR 2001 conference [13]. Because of the language models' possibilities to reason about complex document representations, we believe that language models are specifically promising for reasoning about semi-structured textual data and XML [27, Section 4.8]. The further development of language models requires much data manipulation for obtaining experimental evidence for these claims. Therefore, research in information retrieval based on language models is expected to greatly benefit from rethinking the design of retrieval systems.

Combined querying of structure and text content can be realized in several ways. One of the options is coupling dedicated systems. However, this option is prone to be highly inefficient when data grows to realistic sizes, which are typically huge. Since each individual part of the query has to be processed by a separate system, either each system contains much processing redundancy, or much work is left for the integrating application level. So, integrating information retrieval functionality in a DBMS remains as a major class of possible solutions which can be divided in three subclasses:

- Expressing the entire combined retrieval algorithms in terms of the high-level query language as provided by a standard (commercial) DBMS.
- Exploiting the extension mechanisms of modern DBMSs for providing special-purpose text retrieval functionality.
- Flexible and transparent integration of information retrieval functionality in a query engine.

Unfortunately, current database state-of-the-art has serious deficiencies, making integration of information retrieval with database technology either too inefficient or too limited. The first option above, expressing the retrieval algorithm in a high-level query language, typically suffers from efficiency problems, due to missing application knowledge at the query optimization level in combination with the high complexity of the query expression. The second option, exploiting the extension mechanism, usually approaches information retrieval processing in a black-box manner. This results in similar inefficiencies as in the first option, a system coupled at application level: extension modules often lead to redundant processing and additional work left for the integrating layer [59]. Furthermore, interconnecting and coordinating functionality between modules may lead to additional problems which are usually hard to overcome.

The third option, transparently integrating information retrieval functionality in a DBMS, has the potential resolving these deficiencies. To realize this potential, developments in both information retrieval and database technology are required. Furthermore, on the database side, this requires adaptations on physical and logical levels.

Our proposal addresses the following three closely intertwined research questions:

1. How can we transparently integrate language model information retrieval primitives in the logical algebra layer of a DBMS to provide capabilities for combined querying of both structure and text content?

2. How can we provide physical storage, processing, and optimization support for these logical information retrieval and structure querying primitives?

3. How can we extend database query optimization technology to overcome the typical efficiency bottlenecks described above to support combined querying of both structure and text content?

The remainder of this document proposes a research track giving insight into answering these research questions, as a contribution to advancing both database technology and information retrieval research.

3 Research method and envisioned results

The research plan is divided in two major parts: database architecture and optimization.

The database architecture part addresses the first two research questions. We conjecture that a traditional three level design of DBMSs – distinguishing a physical, a logical, and a conceptual level – provides the best opportunity for balancing flexibility and efficiency. We will concentrate on the core database primitives, facilitating physical data independence and logical data independence. Physical and logical data independence should be ensured by defining requirements for the storage and retrieval of XML data (both structure and text content) at both the physical and logical level. From these requirements, a small number of text retrieval primitives will be defined. These primitives are oriented toward bulk processing, defining a physical algebra and a logical algebra for text retrieval. These will be embedded in existing algebras for structured querying, involving both the ordered and unordered collection types.

Essential for the third research question regarding optimization is the design of rules that define the equivalence between algebraic formulas at the logical level. Recall that black-box approaches are prone to redundant processing. A transparent definition of logical operators would enable a query optimizer to minimize these redundancies. Since this typically occurs where more than one structure/object is involved, we call this inter-object optimization. Also essential is a cost model at the physical level, guided by application knowledge provided by the logical level, to support the proper choice of optimization rules at the logical level.

More concretely, we envision the following three research goals corresponding to our three research questions:

1. to design and develop as a database extension, a logical algebra for basic ranked retrieval operations, including basic operations for the combination of complex representations,

2. to design and develop as a database extension a physical algebra for the efficient storage and ranked retrieval of XML data, and

3. to design and develop rules for inter-object optimization.

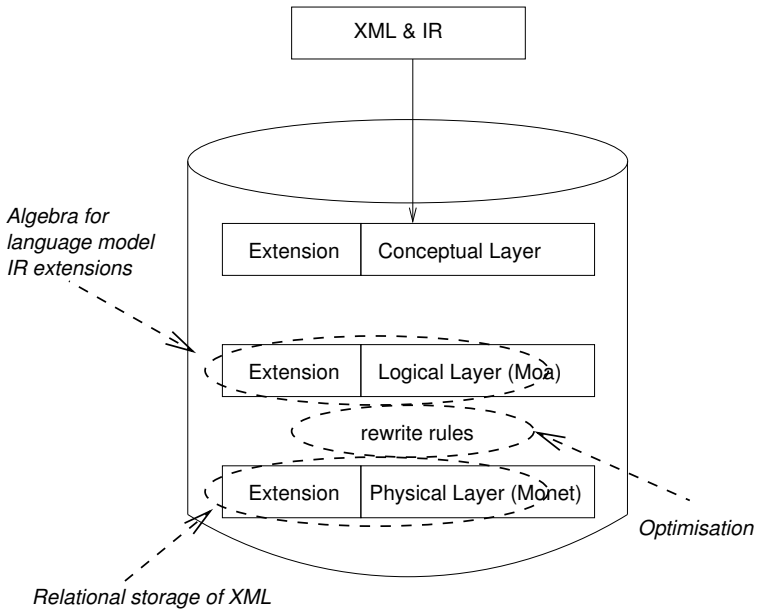

Figure 1: Database internals.

Figure 1 visualizes the prototype database architecture. In the next sections, we elaborate on each of these topics.

3.1 Transparent DBMS information retrieval primitives

A new promising direction in the field of information retrieval is the use of statistical language models. Statistical language models were introduced in 1998 by Ponte and Croft [44], Hiemstra [26], and Miller et al. [41] as a new approach to information retrieval. The language models, which are based on discrete hidden Markov models, do no require the use of complex *tf.idf* term weights. As such, they provide a previously unavailable way to reason about ranking of retrieval results.

Language models have been developed at the University of Twente for cross-language information retrieval, relevance feedback and manually formulated Boolean-structured queries [28, 29], adaptive filtering [27] and web search [35]. As suggested in [41, 1, 35], language models also open exciting new possibilities for combining different content-representations of semi-structured data. In theory that is, because none of these publications actually reported that they were able to implement none trivial versions of such a system. Implementing general purpose solutions to these problems takes a lot of effort without DBMS-support for text retrieval.

This part of the project has to bridge the gap between information retrieval and database principles. The language models will be used to define a small number of retrieval primitives, i.e., content-based text queries referring to regular path expressions, content-based versions

of proximity operators for 'near' and 'adjacent' words, and content-based versions of the traditional Boolean AND, and OR operators to combine the information. New operators should meet the following database requirements:

- adhere to collection-based processing,
- adhere to data independence, and
- adhere to semantic granularity of the logical algebra.

This should lead to a transparent implementation that allows for inter-object query optimization.

3.2 Physical support for complex text queries

Database research at CWI is focused on the database internals, and in particular the effect of changes in both the hardware platforms on which database systems run (much bigger, but (relatively) ever slower memory) and the applications that require database support (in particular, small number of expensive queries instead of high numbers of cheap queries). These changes are studied using the Monet database kernel, developed especially to experiment with implementation techniques for novel application domains on parallel and distributed processing platforms.

Monet has been applied as backend for the storage and retrieval of XML documents in a generic way, that is much more efficient for associative querying than flat XML querying [50]. The database uses a decomposition of the XML structure into binary relational tables. The scheme supports traditional querying of semi-structured data; however, querying does not result in a ranking of the retrieved results, and full-text querying still requires inefficient scans over large parts of the textual data.

This project proposes to build upon this work by decomposing the XML content (the so-called parsed character data, or PCDATA in XML) into tables as well, similar to the way Monet has functioned as backend for information retrieval experiments, see e.g. [60, 58, 57]. To improve the efficiency of such a decomposition, the proposed research investigates the use of techniques from inverted file index structures, e.g., the use of compression techniques discussed in [63], as well as the application of new data structures, in particular the suffix array (see [65]).

Two classes of queries that cause particular difficulty for traditional database systems are containment and proximity queries, such as 'find all text fragments between given tags containing a personname and the keyword president within 10 words from each other'. As discussed in detail by [67], a database system is often less efficient on containment queries than the equivalent information retrieval system, unless the DBMS is extended with new physical operators (like the multi-predicate merge join proposed there).

Not all improvements for such algorithms require the implementation of new operators at the kernel level. For example, the algorithm presented in [61] for efficient k-NN query processing (where k is large, e.g., more than 100) is expressed perfectly as an (efficient) query plan in the common relational operators, and the same holds for approximate string matching using the strategy outlined in [21].

Strategies that can be expressed as highly specialized query plans using 'standard' relational operators have a particular advantage over new kernel operators: their implementation is open and therefore allows for better cooperation with a query optimizer. For example, in

a query for photographs from 1999 with a specific color layout, the constraint 'from 1999' can be pushed through the query plan specified by the k-NN query strategy defined in [61], while this would be infeasible had the k-NN search been a physical operator. This style of physical support is thus of a particular interest, as no query processor could be expected to figure such optimizations on its own, but the number of alternatives for the query optimizer to choose from is not severely restricted.

The envisioned results are better query processing techniques for the types of queries that are currently not handled well by relational database systems. The proposed research is focused on the discovery of several such strategies for containment and proximity queries, supported by new physical operators if necessary. A research question is whether the traditional set-oriented view on physical operators is sufficient, or an ordered physical algebra is more attractive for these types of queries.

3.3 Logical query optimization

As argued in Section 2, a modern database query optimizer should be able to reason over queries that contain clauses over data structures that are typically implemented in different extensions of the DBMS. Current, state-of-the-art optimizer technology can deal with extensions in isolation. We call this intra-object query optimization [4]. By designing an inter-object optimizer layer that is able to bridge the typical orthogonality of database extensions at the logical level, the query optimizer is extended to handle pairs or even groups of interacting extensions.

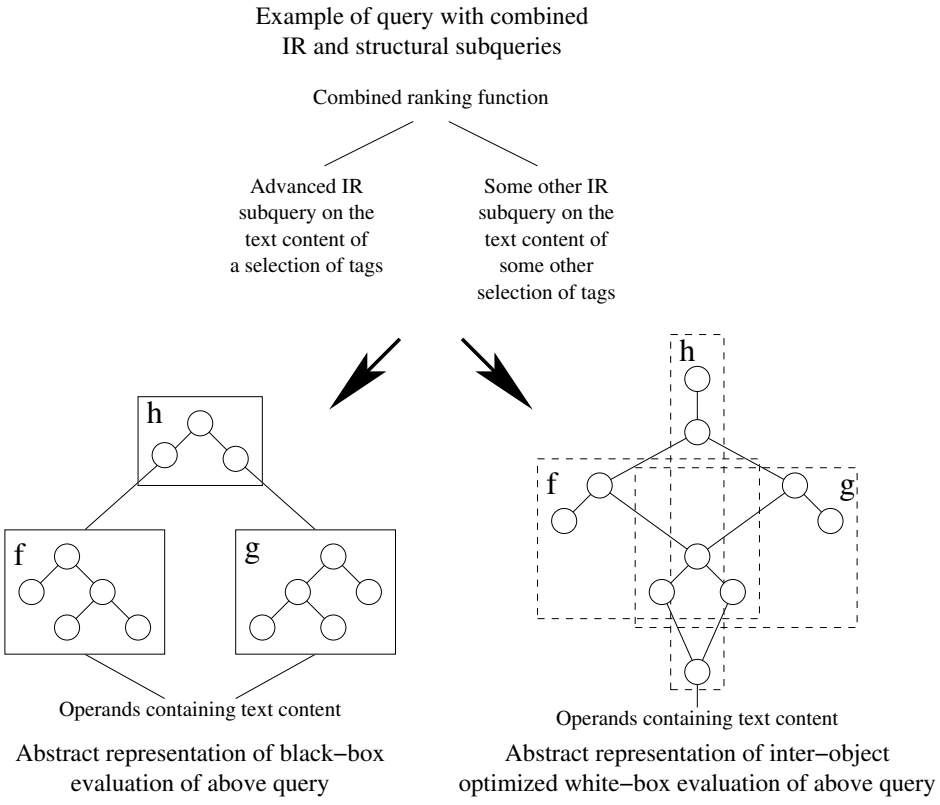

Figure 2: Optimization opportunities for inter-object optimization in a white-box approach.

The example shown in Figure 2 illustrates how otherwise redundant processing might be avoided by 'opening' the black box. On the left-hand side, 'f', 'g', and 'h' represent complex black-box operators (internals shown for explanatory purposes). Consider again the example query from the introduction. In this case 'f' and 'g' would represent the search in image captions and body texts, respectively, and 'h' would represent the restriction that both occur in the same news item and the selection of the top-10. The right-hand side of the figure demonstrates how parts of the black-boxes can be processed together reducing redundancy. It also shows that operators can be reordered vertically due to the transparency of 'f', 'g', and 'h', for example pushing a selection in 'h' through 'f' and 'g'. Note that the black-box approach would typically first retrieve all matching image captions and retrieve all matching body texts, then impose the restriction that they need to be in the same news item, and finally select the top 10. Transparency of 'f' and 'g' allows both the 'same news item' restriction and the top-10 selection to be partly pushed down by the query optimizer, thus reducing the size of intermediate results.

In the above, examples of two classes of inter-object optimization can be distinguished:

parallel Combine common processing tasks in parallel branches of the query operator tree (e.g., combining common parts of 'f' and 'g').

sequential Combine and reorder processing tasks that occur sequentially in the query (e.g., pushing a selection in 'h' through 'f' and 'g').

We envision a three layered optimizer architecture at the logical level:

1. A global optimizer kernel layer that deals with only very basic optimization issues.

2. An inter-object optimizer layer that is spawned by the top layer to handle parts of the query that require different complex operators to cooperate. This layer should be extensible such that optimization functionality for new combinations of extensions can be added when required.

3. An intra-object optimizer layer, similar to the E-ADT approach used in PREDATOR [51]. This layer should be extensible as well, to allow adding new optimizer functionality as new extensions are added to the DBMS.

The extensibility of the query optimizer at the logical level is crucial. It allows adaptation of the query optimization capabilities of the system, similar to the need to have an extension mechanism in a DBMS to cope with new media types. It also allows for proper embedding and exploitation of application knowledge in the query optimizer.

We envision this application knowledge to be used to provide hints to the lower physical optimizer and its cost model. A text retrieval optimizer extension might, for instance, give hints about which selectivity model to use to the cost model at the physical level [5, 6]. By generating different alternative physical expressions and offering them to the physical cost model for a cost prediction, including a selectivity hint, the logical optimizer can then choose which variant to send to the physical layer for further processing and execution.

Additionally, the logical optimizer extension for text retrieval might exploit quality information learned from user feedback [7]. This information, like the choice of the selectivity model for cost prediction, is typical application dependent knowledge. At the logical level this information is still available, in contrast with the physical level, thus allowing for more sophisticated optimization decisions.

3.4 Discussion

To allow experimental validation of the concepts to be developed, a prototype database system is needed. A similar prototype database system has been developed with as logical layer Moa and as physical layer Monet [34]. Both layers are well attuned to each other. Furthermore, both systems have been designed with the purpose to support problems like this project has in mind. From a practical point of view, access to the internals of the systems is an advantage.

Such a prototype database system needs a conceptual layer with an accompanying query language as well, for example XQuery including an extension for ranked retrieval as proposed in [18].

4 Related work

This section addresses related work from the fields of Database research and Information Retrieval research.

4.1 The integration of information retrieval and databases

In Section 2 we mentioned three ways to realize combined querying of structure and text content by integrating information retrieval techniques in a DBMS. Here we provide a brief overview of related work regarding these three classes.

In the first case, the entire combined retrieval algorithms is expressed in terms of the high-level query language, e.g., SQL, as provided by a standard (commercial) DBMS. This means that existing elements of a query language are used to express the entire ranking algorithm. This is an easy means to realize information retrieval functionality but, as we argued above, has several major drawbacks. Examples in literature can be found in [24, 23, 22, 16]. The case where the extension mechanism of a DBMS is exploited can be divided in two different situations [53]: the extension acts as a wrapper for an external dedicated information retrieval system or the extension contains an entire retrieval system, e.g., the commercial Excalibur text datablade [15] for the Informix DBMS [31]. Furthermore, two ways exist to provide information retrieval functionality: either capture the entire retrieval procedure by one single function or provide several functions that together provide the requested functionality [17, 19]. The last approach can be characterized as the 'back-to-basics' approach. Many completely different results are known that belong to this class. An early example of such an approach is the AIM system and related projects [14, 43]. Another well-known system is MULTOS [3]. The Mirror project [59] proposed the division of information retrieval processing parts over the three typical architectural DBMS layers to improve efficiency.

Several key principles behind the problem of integrating information retrieval in a DBMS are described in [62]. In [26] the language modeling approach was introduced. Van Zwol and Apers [68, 69] describe the integration of querying structure and multi media data in the context of searching the web and XML document collections. When optimizing text retrieval queries not only execution performance plays a role, but quality is as least as important. Prediction of the quality implications is therefore an interesting issue [7]. Managing uncertainty in database design is addressed in [12].

4.2 Language models for information retrieval

Although hidden Markov models have been successfully used for e.g. automatic speech recognition for over 20 years now [33], their use was only recently proposed for text retrieval [44, 26, 41] and image retrieval [54]. Inspired by the speech recognition community, the text approach is called a language modeling approach to information retrieval. The approach allows for high-quality retrieval results without the need for complex term weighting algorithm like the familiar *tf.idf* weights (see e.g. [47]). As such, they have really contributed to our understanding of content-based retrieval. Because of the basic understanding of the underlying algorithms, the models are easily extended to support multiple query representations [2, 28, 64], and multiple content-representations [41, 1, 35, 36], which makes them the ideal candidate for developing general purpose primitives for the combined querying of content and structure.

4.3 Physical support for complex text queries

As has been observed, most previous attempts at integrating database technology and information retrieval techniques have followed the approach of extending the DBMS with a more or less complete information retrieval system. A notable exception to this rule is the recent work by Zhang et al. [67], who propose to extend the DBMS with new join operators for processing containment queries efficiently. In the field of data structures and theoretical computer science however, many papers address the problem of solving proximity queries efficiently, and even the more general class of regular expression matching, also on large archives, see e.g. [37]. We plan to base new physical operators on existing techniques using suffix arrays, comparable to the approach taken in [21] for approximate string matching with *q-grams*.

As discussed, complex access structures are better designed as query plans expressed in more basic physical operators than as black-box complex physical operators. In [61] a case study is presented in image retrieval where this argument has been proven true. In [40] the claim is supported that Monet's physical operators exploit better the modern CPU's (which have relatively high amounts of cache). De Vries [56] details a notion of content independence, as a design analogon to data independence, which allows the separation of retrieval model from queries posed to the database. Schmidt et al. [49] present how to map complex XML documents to binary tables in Monet, resulting in highly efficient query processing for associative queries. The canonical Monet publication [8], details the specific design features and advantages for complex structured data provided by Monet.

4.4 Query optimization

Query optimization has played a central role in database research for a long time. Good overviews can be found in any standard database text book. At the logical level algebra properties like commutativity and associativity are used in transformation rules that are specializations of the common rule to limit the intermediate results as soon and as much as possible. Typical examples are rules to push selections and the most restrictive joins down the operator tree. To allow the optimizer to choose the cheapest of the possible equivalent expressions, it requires a simple cost model that can predict the cardinalities of intermediate results. Such a cardinality estimate draws heavily on a selectivity model. As such, selectivity estimation has been subject to extensive research [9, 66, 42, 30, 39, 11, 32, 20, 45, 10].

In [25, 38] predicate reordering techniques have been proposed that are even more sophisticated. NF^2 databases require a more sophisticated optimizer in general since the NF^2 paradigm allows nesting of relations [52]. This also holds for OO databases. As mentioned in Section 2 the modern day DBMS extension mechanism poses new demands to the query optimizer. The PREDATOR project [51] proposed the E-ADT approach allowing the ADT programmer to add optimizer functionality for the ADT as well. We call this approach the intra-object optimization approach.

5 Conclusion and acknowledgements

The CIRQUID project bridges the gap between structured query capabilities of XML query languages and relevance-oriented querying. Current techniques for XML querying, originating from the database field, do not support relevance-oriented querying, but techniques for ranking documents, originating from the information retrieval field, typically do not take document structure into account. Ranking is of the utmost importance if large collections are queried, to assist the user in finding the most relevant documents in a retrieved set.

The ideas and plans addressed in this report will be experimentally validated by building one or more prototype databases systems, which will be evaluated on benchmark testbeds. The project will run from from end 2002 to begin 2007. We are very grateful to the Netherlands Organization for Scientific Research (NWO) for making the project possible. Many thanks go to Peter Apers for useful comments and advice.

References

[1] J. Baan, A.R. van Ballegooij, J.M. Geusenbroek, J. den Hartog, D. Hiemstra, J.A. List, I. Patras, S. Raaijmakers, C. Snoek, L. Todoran, J. Vendrig, A.P. de Vries, T. Westerveld, and M. Worring. Lazy users and automatic video retrieval tools in (the) lowlands. In Voorhees and Harman [55], pages 159–168.

[2] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 222–229, 1999.

[3] E. Bertino, F. Rabitti, and S. Gibbs. Query Processing in a Multimedia Document System. *ACM Transactions on Office Information Systems*, 6(1):1–41, 1988.

[4] H.E. Blok. *Database Optimization Aspects for Information Retrieval.* PhD thesis, University of Twente, 2002.

[5] H.E. Blok, R.S. Choenni, H.M. Blanken, and P.M.G. Apers. A selectivity model for fragmented relations in information retrieval. Technical Report 01-02, Centre for Telematics and Information Technology (CTIT), University of Twente, 2001.

[6] H.E. Blok, R.S. Choenni, H.M. Blanken, and P.M.G. Apers. A selectivity model for fragmented relations in information retrieval. *IEEE Trans. on Knowledge and Data Engineering*, (to appear).

[7] H.E. Blok, D. Hiemstra, R.S. Choenni, F.M.G. de Jong, H.M. Blanken, and P.M.G. Apers. Predicting the cost-quality trade-off for information retrieval queries: Facili-

tating database design and query optimization. In *Tenth International Conference on Information and Knowlegde Management (ACM CIKM'01)*, ACM Press, 2001.

[8] P.A. Boncz and M.L. Kersten. MIL Primitives for Querying a Fragmented World. *The VLDB Journal*, 8(2):101–119, 1999.

[9] A.F. Cardenas. Analysis and performance of inverted data base structures. *Communications of the ACM*, 18(5):253–263, 1975.

[10] S. Chaudhuri, R. Motwani, and V. Narasayya. On Random Sampling over Joins. In *Proceedings of the 1999 ACM SIGMOD International Conference on the Management of Data*, pages 263–274. ACM Press, 1999.

[11] C.M. Chen and N. Roussopoulos. Adaptive Selectivity Estimation Using Query Feedback. In *Proceedings of the 1994 ACM SIGMOD International Conference on the Management of Data*, pages 161–172. ACM Press, 1994.

[12] R.S. Choenni and H.M. Blanken. Modelling uncertainty in physical database design. In *Proceedings of the 7th International Workshop on Knowledge Representation meets Databases (KRDB 2000), Berlin, Germany, 2000*, number 29 in CEUR Workshop Proceedings, pages 31–44, 2000.

[13] W.B. Croft, D.J. Harper, D.H. Kraft, and J. Zobel, editors. *Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01).* ACM Press, 2001.

[14] P. Dadam, K. Kuespert, F. Andersen, H.M. Blanken, R. Erbe, J. Guenauer, V. Lum, P. Pistor, and G. Walch. A DBMS Prototype to Support NF^2 Relations: An Integrated View on Flat Tables and Hierarchies. In *Proceedings of the 1986 ACM SIGMOD International Conference on the Management of Data*, pages 356–367. ACM Press, 1986.

[15] Excalibur Text Search Datablade. Informix Tech Brief, 1999. See also [31].

[16] O. Frieder, D.A. Grossman, A. Chowdhury, and G. Frieder. Efficiency Considerations for Scalable Information Retrieval Servers. *Journal of Digital Information*, 1(5), 2000.

[17] N. Fuhr. Models for Integrated Information Retrieval and Database Systems. *IEEE Data Engineering Bulletin*, 19(1):3–13, 1996.

[18] N. Fuhr and K. Grossjohann. XIRQL: A query language for information retrieval in XML documents. In [13], pages 172–179, 2001.

[19] N. Fuhr and T. Rölleke. A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. *ACM Transactions on Information Systems*, 15(1):32–66, 1997.

[20] S. Ganguly, P.B. Gibbons, Y. Matias, and A. Silberschatz. Bifocal Sampling for Skew-Resistant Join Size Estimation. In *Proceedings of the 1996 ACM SIGMOD International Conference on the Management of Data*, pages 271–281.

[21] L. Gravano, P. Ipeirotis, H.V. Jagadish, N. Koudas, S. Muthukrishnan, and D. Srivastava. Approximate string joins in a database (almost) for free. In *Proceedings of the 27th International Conference on Very Large Databases (VLDB 2001)*, 2001.

[22] D.A. Grossman and O. Frieder. *Information retrieval: algorithms and heuristics.* The Kluwer international series in engineering and computer science. Kluwer Academic, Boston, 1998.

[23] D.A. Grossman, O. Frieder, D.O. Holmes, and D.C. Roberts. Integrating Structured Data and Text: A Relational Approach. *Journal of the American Society of Information Science*, 48(2):122–132, 1997.

[24] D.A. Grossman, D.O. Holmes, and O. Frieder. A Parallel DBMS Approach to IR in TREC-3. In *Proceedings of the Third Text Retrieval Conference (TREC-3)*, NIST Special publications, pages 279–288, 1994.

[25] J.M. Hellerstein and M. Stonebreaker. Predicate Migration: Optimizing Queries with Expensive Predicates. In *Proceedings of the 1993 ACM SIGMOD International Conference on the Management of Data*, pages 267–276. ACM Press, 1993.

[26] D. Hiemstra. A linguistically motivated probabilistic model of information retrieval. In *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, pages 569–584, 1998.

[27] D. Hiemstra. *Using language models for information retrieval*. PhD thesis, Centre for Telematics and Information Technology, University of Twente, 2001.

[28] D. Hiemstra and F.M.G. de Jong. Disambiguation strategies for cross-language information retrieval. In *Proceedings of the third European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, pages 274–293, 1999.

[29] D. Hiemstra and A.P. de Vries. Relating the new language models of information retrieval to the traditional retrieval models. Technical Report TR-CTIT-00-09, Centre for Telematics and Information Technology, 2000.
http://www.ub.utwente.nl/webdocs/ctit/1/00000022.pdf

[30] A. IJbema and H.M. Blanken. Estimating bucket accesses: A practical approach. In *Proceedings of the Second International Conference on Data Engineering (ICDE'86)*, pages 30–37. IEEE Computer Society, IEEE, 1986.

[31] Informix Corporation. URL: http://www.informix.com/.

[32] Y.E. Ioannidis and V. Poosala. Balancing Histogram Optimality and Practicality for Query Result Size Estimation. In *Proceedings of the 1995 ACM SIGMOD International Conference on the Management of Data*, pages 233–244. ACM Press, 1995.

[33] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

[34] M. van Keulen, J. Vonk, A.P. de Vries, J. Flokstra, and H.E. Blok. Moa: extensibility and efficiency in querying nested data. Technical report, Centre for Telematics and Information Technology, 2002.

[35] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pages 27–34, 2002.

[36] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization. In [13], pages 111–119, 2001.

[37] N. Jesper Larsson. *Structures of String Matching and Data Compression*. PhD thesis, Department of Computer Science, Lund University, Sweden, 1999.

[38] A.Y. Levy, I.S. Mumick, and Y. Sagiv. Query Optimization by Predicate Move-Around. In *Proceedings of the 20th VLDB Conference*, 1994.

[39] R.J. Lipton, J.F. Naughton, and D.A. Schneider. Practical Selectivity Estimation through Adaptive Sampling. In H. Garcia-Molina and H.V. Jagadish, editors, *Proceedings of the 1990 ACM SIGMOD International Conference on the Management of Data*, pages 1–11. ACM Press, 1990.

[40] S. Manegold, P. A. Boncz, and M. L. Kersten. Optimizing Main-Memory Join On Modern Hardware. *IEEE Transactions on Knowledge and Data Eng.*, 2002. To appear. A preprint version is available from the IEEE Digital Library or as CWI Technical Report INS-R9912.

[41] D.R.H. Miller, T. Leek, and R.M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 214–221, 1999.

[42] B. Muthuswamy. A Detailed Statistical Model for Relational Query Optimization. In *Proceedings of the 1985 ACM SIGMOD International Conference on the Management of Data*, pages 439–448. ACM Press, 1985.

[43] H.B. Paul, H.J. Schek, M.H. Scholl, G. Weikum, and U. Deppisch. Architecture and Implementation of the Darmstadt Database Kernel System. In *Proceedings of the 1987 ACM SIGMOD International Conference on the Management of Data*, pages 196–207. ACM Press, 1987.

[44] J.M. Ponte and W.B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 275–281, 1998.

[45] V. Poosala, Y.E. Ioannidis, P.J. Haas, and E.J. Shekita. Improved Histograms for Selectivity Estimation of Range Predicates. In *Proceedings of the 1996 ACM SIGMOD International Conference on the Management of Data*, pages 294–305.

[46] S.E. Robertson and K. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[47] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[48] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[49] A.R. Schmidt, M.L. Kersten, M.A. Windhouwer, and F. Waas. Efficient Relational Storage and Retrieval of XML Documents (Extended Version). In *The World Wide Web and Databases - Selected Papers of WebDB 2000*, volume 1997 of *Lecture Notes in Computer Science (LNCS), Springer-Verlag*, pages 137–150, 2000.

[50] A.R. Schmidt, F. Waas, M.L. Kersten, D. Florescu, I. Manolescu, M.J. Carey, and R. Busse. The XML benchmark project. Technical report, Centrum voor Wiskunde en Informatica, 2001.

[51] P. Seshradri and M. Paskin. PREDATOR: An OR-DBMS with Enhanced Data Types. In *Proceedings of the 1997 ACM SIGMOD International Conference on the Management of Data*, pages 568–571. ACM Press, 1997.

[52] H. Steenhagen. *Optimization of Object Query Languages*. PhD thesis, University of Twente, Enschede, The Netherlands, 1995.

[53] S.R. Vasanthakumar, J.P. Callan, and W.B. Croft. Integrating INQUERY with an RDBMS to Support Text Retrieval. *IEEE Data Engineering Bulletin*, 19(1):24–33, 1996.

[54] N. Vasconcelos and A. Lippman. Embedded mixture modeling for efficient probabilistic content-based indexing and retrieval. In *Proceedings of the SPIE*, pages 134–143, 1998.

[55] E.M. Voorhees and D.K. Harman, editors. *Proceedings of the tenth Text Retrieval Conference (TREC-10)*, NIST Special publications, 2002.

[56] A.P. de Vries. Content Independence in Multimedia Databases. *Journal of the American Society for Information Science and Technology*, 52:954–960, 2001.

[57] A.P. de Vries. A poor man's approach to CLEF. In *CLEF 2000: Workshop on cross-language information retrieval and evaluation*, 2000.

[58] A.P. de Vries. The Mirror DBMS at TREC-9. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Nineth Text Retrieval Conference TREC-9*, number 500–249 in NIST Special publications, pages 171–177, 2001.

[59] A.P. de Vries. *Content and Multimedia Database Management Systems*. PhD thesis, University of Twente, 1999.

[60] A.P. de Vries and D. Hiemstra. The Mirror DBMS at TREC-8. In *Proceedings of the Eighth Text Retrieval Conference TREC-8*, number 500–246 in NIST Special publications, pages 725–734, 1999.

[61] A.P. de Vries, N. Mamoulis, N. J. Nes, and M. L. Kersten. Efficient k-NN Search on Vertically Decomposed Data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2002.

[62] A.P. de Vries and A.N. Wilschut. On the Integration of IR and Databases. In *8th IFIP 2.6 Working Conference on Data Semantics 8*, pages 16–31, 1999.

[63] I.A. Witten, A. Moffat, and T.C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, 1999.

[64] J. Xu, R. Weischedel, and C. Nguyen. Evaluating a probabilistic model for cross-lingual information retrieval. In [13], pages 105–110, 2001.

[65] M. Yamamoto and K. Church. Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. *Computational Linguistics*, 27(1):1–30, 2001.

[66] S.B. Yao. Approximating Block Accesses in Database Organizations. *Communications of the ACM*, 20(4):260–261, 1977.

[67] C. Zhang, J.F. Naughton, D.J. DeWitt, Q. Luo, and G. Lohman. On supporting containment queries in relational database management systems. In *Proceedings of the ACM SIGMOD Conference*, 2001.

[68] R. van Zwol and P.M.G. Apers. The webspace method: On the integration of database technology with information retrieval. In *In proceedings of CIKM'00*, 2000.

[69] R. van Zwol. *Modelling and Searching web-based document collections*. PhD thesis, University of Twente, 2002.