The Database Research Group at the Max-Planck Institute for Informatics

Gerhard Weikum
Max-Planck Institute for Informatics
Stuhlsatzenhausweg 85
D-66123 Saarbruecken, Germany
weikum@mpi-inf.mpg.de

1. INTRODUCTION

The Max-Planck Institute for Informatics (MPI-INF) is one of 80 institutes of the Max-Planck Society, Germany's premier scientific organization for foundational research with numerous Nobel prizes in natural sciences and medicine. MPI-INF hosts about 150 researchers (including graduate students) and comprises 5 research groups on algorithms and complexity, programming logics, computational biology and applied algorithmics, computer graphics, and databases and information systems (DBIS). This report gives an overview of the DBIS group's mission and ongoing research.

2. VISION AND RESEARCH DIRECTIONS

The research of the DBIS group pursues two major directions:

1. intelligent organization and search of semistructured information, in intranets, digital libraries, and on the Web;
2. architecture and strategies for self-organizing distributed information systems, particularly, peer-to-peer systems.

2.1 Intelligent Organization and Search of Information

The age of information explosion poses tremendous challenges regarding the intelligent organization of data and the effective search of relevant information in business and industry (e.g., market analyses, logistic chains), society (e.g., health care), and virtually all sciences that are more and more data-driven (e.g., gene expression data analyses and other areas of bioinformatics). The problems arise in intranets of large organizations, in federations of digital libraries and other information sources, and in the most humongous and amorphous of all data collections, the World Wide Web and its underlying numerous databases that reside behind portal pages. The Web bears the potential of being the world's largest encyclopedia and knowledge base, that would be of great value to all kinds of "knowledge workers" from students to Nobel laureates, but we are very far from being able to exploit this potential.

Search-engine technologies provide support for organizing and querying information; for simple mass-user queries that aim to find popular Web pages on pop stars, soccer clubs, or the latest Hollywood movies, existing engines like Google are probably the best solution. But for advanced information demands search engines all too often require excessive manual preprocessing, such as manually classifying documents into a taxonomy for a good Web portal, or manual postprocessing such as browsing through large result lists with too many irrelevant items or surfing in the vicinity of promising but not truly satisfactory approximate matches. The following are example queries where current Web and intranet/enterprise search engines fall short:

Q1: Which professors from Saarbruecken in Germany teach information retrieval and participate in EU projects?

Q2: Which drama has a scene in which a woman makes a prophecy to a Scottish nobleman that he will become king?

Q3: Who was the French woman that I met in a program committee meeting where Paolo Atzeni was the PC chair?

Why are these queries difficult (too difficult for current Web search engines, unless one invests a huge amount of time to manually explore large result lists with mostly irrelevant and some mediocre matches)? For Q1 no single Web site is a good match; rather one has to look at several pages together within some bounded *structural context*: the homepage of a professor with his address, a page with course information linked to by the homepage, and a research project page that is a few hyperlinks away from the homepage.

Q2 cannot be easily answered because a good match does not necessarily contain the keywords "woman", "prophecy", "nobleman", etc., but may rather say something like "Third witch: All hail, Macbeth, thou shalt be king hereafter!" and the same document may contain the text "All hail, Macbeth! hail to thee, thane of Glamis!". So this query requires some *background knowledge* to recognize that a witch is usually female in the literature, "shalt be" refers to a prophecy, and "thane" is a title for a Scottish nobleman.

Q3 combines the difficulties of Q1 and Q2 as it requires background knowledge but also needs to put together bits and pieces from different information sources, including semistructured desktop data: PC meetings that I attended

according to my electronic calendar, conferences on which I served on the PC found in my email archive, PC members listed in electronic proceedings, and detailed information found on researchers' homepages. And after having identified a candidate like Sophie Cluet from Paris, one needs to infer that Sophie is a typical female first name and that Paris most likely denotes the capital of France rather than the 500-inhabitants town of Paris, Texas, which became known through a movie.

Answering such queries with high precision, despite high diversity and noise in the underlying data, calls for a new kind of *"semantic search" engine* that combines concepts and techniques from database and information-retrieval (IR) systems. A promising starting point is *XML IR* [1], especially when combined with background knowledge in the form of ontologies and thesauri. We will outline our approach along these lines in Subsection 3.1.

The envisioned "semantic search" should not only address the querying itself, but also aim at *intelligent organization of information*. Data with more explicit structure, semantic annotations, and clean data items (e.g., reconciled names of persons, organizations, etc.) becomes easier to search with high precision. But the Web and also federations of digital libraries and other data sources are highly diverse in terms of structure, annotations, and data quality (e.g., authority, resolution, completeness, freshness, etc.). Therefore, we also pursue ways of automatically annotating and structuring information, using techniques from natural language processing (NLP) and statistical machine learning. We will outline our approaches in this area in Subsection 3.3.

2.2 Self-organizing Distributed Information Systems

Even when we focus on scientific information alone and leave out business and entertainment data, new information is produced and compiled world-wide in a highly distributed manner, in federations of digital libraries or e-science repositories and, of the course, on the Web with millions of scholars and students. Thus, it is natural to pursue a completely decentralized peer-to-peer (P2P) architecture [30] for managing this information explosion, intelligently organizing the information, and efficiently searching it.

The P2P approach bears the potential of overcoming the shortcomings of today's Web search engine technology and successfully tackling the kinds of killer queries mentioned in Subsection 2.1. In our architecture every peer, e.g., the home PC of a scientist or student, has a full-fledged search engine that indexes a small portion of the Web, according to the interest profile of the user. Such an architecture has four major advantages over a centralized server farm:

1. As the data volume and the query load per peer are much lighter, the peer's search engine can employ much more advanced techniques for concept-based rather than keyword-based search, leveraging background knowledge in the form of thesauri and ontologies and powerful mathematical and linguistic techniques such as spectral analysis and named entity recognition.

2. Peers can collaborate for finding better answers to difficult queries: if one peer does not have a good result locally it can contact a small number of judiciously chosen peers who are considered "knowledgeable" on the query topic. This approach should often be able to exploit the small-world phenomenon on the Web: knowledgable peers are only a short distance away.

3. A P2P system can gather and analyze bookmarks, query histories, user click streams, and other data about user and community behavior; the implicit and explicit assessments and recommendations derived from this input can be leveraged for better search results. In contrast to a central server, the P2P approach provides each user with direct, fine-grained control over which aspects of her behavior may be collected and forwarded to other peers.

4. A politically important issue is that a P2P search engine is less susceptible to manipulation, censorship, and the bias induced by purchased advertisements.

Thus, a P2P approach to information search could pave the way towards a *"social search"* super-engine that leverages the collaborative "wisdom of crowds" [14]. Challenges that we face towards this vision are scalability and the desire to make the P2P network self-organizing and resilient to high dynamics (failures and churn) and possible manipulation (cheating, spam, etc.). We will outline our approach to P2P search in Subsection 3.2. Again, search needs to be complemented by powerful Web mining, which will be discussed in Subsection 3.3.

3. ONGOING PROJECTS

3.1 XML Ranked Retrieval: TopX and Sphere Search

Non-schematic XML data that comes from many different sources and inevitably exhibits heterogeneous structures and annotations (i.e., XML tags) cannot be adequately searched using database query languages like XPath or XQuery. Often, queries either return too many or too few results. Rather the ranked-retrieval paradigm is called for, with relaxable search conditions, various forms of similarity predicates on tags and contents, and quantitative relevance scoring.

TopX [35] is a search engine for ranked retrieval of XML data, developed at MPI-INF. TopX supports a probabilistic-IR scoring model for full-text content conditions and tag-term combinations, path conditions for all XPath axes as exact or relaxable constraints, and ontology-based relaxation of terms and tag names as similarity conditions for ranked retrieval.

While much of the TopX functionality was already supported in our earlier work on the XXL system [32, 33], TopX has an improved scoring model for better precision and recall, and it is much more efficient and scalable. TopX has been stress-tested and experimentally evaluated on a variety of datasets including the TREC Terabyte benchmark, the INEX XML information retrieval benchmark, and an XML version of the Wikipedia encyclopedia. For the INEX 2006 benchmark [17], TopX serves as the official

reference engine for topic development and several benchmarking tasks. For good performance, TopX employs various novel techniques: carefully designed index structures [24, 35], probabilistic models as efficient score predictors [34, 35], judicious scheduling of index accesses [35, 3], and the incremental merging of index lists for on-demand, self-tuning query expansion [36].

For top-k similarity queries we have extended the TA family of threshold algorithms [13]. Large disk-resident index lists for content terms and tag names strongly suggest using the TA variant with sorted access only for disk I/O efficiency. This method maintains a priority queue of score intervals for candidate items and performs a conservative threshold test based on upper bounds of candidate scores. Our new method uses a probabilistic threshold test for fast approximation of top-k results. It is based on score predictors that compute convolutions of score distributions in the yet to be scanned tails of index lists. This method provides an order of magnitude improvement in run-time at high and probabilistically controllable levels of precision and recall relative to the TA baseline [34, 35]. Our recent techniques for judicious scheduling of block-level sequential and item-level random accesses yields an additional performance gain by another factor of five [3].

For enhanced quality of search results, relevance feedback is a well-known IR technique that expands keyword queries from the content of elements marked as relevant by the user. We have developed novel feedback techniques that expand a keyword query into a possibly complex content-and-structure query that specifies additional XPath conditions on the structure of the desired results [25]. This way the user is largely relieved from understanding the subtleties of XPath and the XML structure of the underlying, possibly highly heterogeneous, XML documents. As an example, consider a query about the life of the physicist Max Planck on a richly tagged XML version of Wikipedia. As a keyword query, the user may formulate this request as "life physicist Max Planck". Unfortunately, the results may be dominated by articles about Max Planck institutes, for example, in the area of life sciences. By providing the feedback that elements on persons and their biographies and the topic physics in mediocre result documents are preferred over completely irrelevant results, the engine would ideally be able to automatically generate a content-and-structure query such as

```
//article[.//person ftcontains(‘‘Max Planck’’)]
        [.//category ftcontains(‘‘physicist’’)].
```

This XPath Full-Text query would yield much more precise results.

As XML data is still rare on the surface Web and even structured Deep-Web sources do often expose only HTML pages, we have started working on converting Web data into XML format. This involves identifying and tagging named entities like persons, organizations, locations, time-points, and time periods. To this end, we are using open-source tools like ANNIE from the University of Sheffield [12] and MinorThird from CMU [11], resulting in a richly (but heuristically) tagged corpus that unifies XML and HTML and other Web documents by casting all data into XML. As Web pages are often highly inter-linked, the resulting corpus is no longer a repository of XML trees, but includes many XLinks (created from href hyperlinks) leading to an arbitrary XML graph. We have developed the SphereSearch prototype engine [15] that can search such graph data and rank search results, using a query language that is simpler than XPath Full-Text but much more powerful than keyword-based Web search. The result of a query is a subgraph spanned by nodes that approximately match and have high scores for the query's elementary conditions. Finding the top-k results involves an approximation to the Steiner tree problem, based on mimimum spanning trees on a precomputed connection graph, using a top-k threshold algorithm.

TopX and SphereSearch are incomparable in terms of query expressiveness. TopX is generally more efficient, but currently limited to XML trees. Our future research will aim at extending TopX to include the richer graph-based search capabilities of SphereSearch in a highly efficient and scalable manner.

3.2 Peer-to-Peer Search: Minerva

For P2P search over Web data with ranked retrieval, we are developing the Minerva system [5].[1] Each peer has a full-fledged Web search engine, including a crawler and an index manager. The crawler may be thematically focused or crawl results may be postprocessed so that the local index contents reflects the corresponding user's interest profile. For collaborative search, the peers are connected by by an overlay network based on a distributed hash table (DHT). The DHT also forms the basis of a conceptually global but physically decentralized and scalable directory that contains metadata and statistics about the peers' contents and quality. Note that, for scalability, the directory is not designed as a page-granularity global Web index, but is limited in size to the number of indexed features (e.g., keywords or topics) times the number of peers. This avoids the pitfalls outlined in [19]. Also note that the pursued P2P Web search includes ranked retrieval and is thus fundamentally much more difficult than Gnutella-style file sharing or simple key lookups via DHTs.

With a user's highly specialized and personalized "power search engine" most queries should be executed locally, but once in a while the user may not be satisfied with the local results and would then want to contact other peers. This is the *query routing* (or peer-selection) problem, the cornerstone of the P2P search engine. Although the problem is related to earlier work on metasearch engines and distributed information retrieval [21], the P2P setting is much more challenging because of larger scale and high dynamics. A "good" peer to which the user's query should be forwarded would have thematically relevant index contents, which could be measured by statistical notions of similarity between peers. On the other hand, each additional target peer for a query should yield novel results that are not yet provided by previously selected peers or even the local index of the query initiator itself. To this end, we have developed an overlap-aware query routing strategy that aims to optimize a weighted combination of search result quality and novelty [4, 6]. As the query routing decision made for execution planning is in the critical path of user-perceived response time, fast estimation of quality-novelty measures is crucial. We have developed

[1]Minerva is the Roman goddess of science, wisdom, and learning, and is also the icon of the Max Planck Society.

new methods for this purpose, utilizing compact synopses like hash sketches and min-wise independent permutations in combination with the underlying DHT.

For the actual top-k query processing in the P2P network, it is sometimes necessary to aggregate index entries from multiple peers, combining their local scores into a global quality measure. The KLEE algorithm for distributed top-k queries [22] aims to minimize network latency, network bandwidth consumption, and the local CPU and disk IO cost of the participating peers. KLEE proceeds in three or, optionally, four phases, driven by the query originator as a per-query coordinator.[2] These phases serve to 1) determine an initial set of top-k candidates and obtain score-distribution histograms and Bloom-filter summaries from peers, 2) estimating the cost/benefit ratio of requesting additional synopses, 3) optionally obtaining this extra information, and 4) obtaining missing scores for the remaining candidates and computing the total scores for the final top-k result. KLEE has been implemented in the Minerva testbed, and has consistently outperformed various competitors on several real-life datasets and query benchmarks.

The above outlines of query routing and query processing in a P2P network show that distributed management of statistical information about peers and their data collections is a key issue. This involves efficient gathering and dissemination of statistics as well as estimations for specific purposes. A global measure of particular interest (for query routing and query result merging) is the document frequency (df) of a keyword, i.e., the total number of distinct documents in the entire network that contain the keyword. Estimating this number is difficult because of duplicate documents at different peers. We have developed a highly efficient and accurate method for this problem, by combining hash sketches and the DHT-based overlay network [8]. Another network-wide statistical estimation problem is to efficiently determine pairs of highly correlated or anti-correlated keywords, either in queries or in the data. In [7] we have developed a technique that utilizes the DHT-based infrastructure for an efficient solution, which can piggyback all necessary message exchanges on the network traffic that is needed for standard query routing and execution anyway. Awareness of keyword correlations is useful for more effective query routing decisions.

As mentioned in Subsection 2.2, a P2P network is a natural habitat for "social search" that leverages community assessments. A simple form of such community input are link analysis methods like Google's PageRank or Kleinberg's HITS. But these are centralized algorithms with very high memory demand, and their distributed variants assume that the underlying Web graph can be nicely partitioned among sites. In contrast, a P2P system like Minerva emphasizes the autonomy of peers that can crawl Web fragments and gather their own local content at their discretion. JXP [23] is a new algorithm for dynamically computing, in a decentralized P2P manner, global authority scores when the Web graph is spread across many autonomous peers. In this setting, the peers' graph fragments may overlap arbitrarily, and peers are (a priori) unaware of other peers' fragments. With JXP, each peer computes the authority scores of the pages that it has in its local index, by locally running the standard PageRank algorithm. A page may be known and indexed by multiple peers, and these may have different scores for that same page. A peer gradually increases its knowledge about the rest of the network by meeting with other, randomly chosen, peers and exchanging information, and then recomputes the PageRank scores for its pages of interest. The local computations are very space-efficient (as they require only the local graph and the authority-score vector), and fast (as they operate on much smaller graph fragments than a server-side global PageRank computation). We have proven, using the theory of Markov-chain aggregation/disaggregation, that the JXP scores do indeed converge to the same values as a global PageRank computation on the full Web graph [23].

[2]Klee is an expressionistic painter, but also the German word for clover, which has usually three leaves but infrequently occurs with four leaves. The latter is traditionally viewed as a symbol of good luck.

3.3 Web Mining

The duality between better organization of information and more effective search has motivated us to embark also on various aspects of Web mining, so as to provide a richer basis for better search capabilities. The following subsections briefly discuss the work along these lines.

3.3.1 Query Logs and Click Streams

Information about user behavior is a rich source to build on. This includes relatively static properties like bookmarks or embedded hyperlinks pointing to high-quality Web pages, but also dynamic properties inferred from query logs and click streams. For example, suppose a user clicks on a specific subset of the top 10 results returned by a search engine for a query with several keywords, based on having seen the summaries of these pages. This implicit form of relevance feedback establishes a strong correlation between the query and the clicked-on pages. When a user does not click on any of the top 10 results for a given query and rather chooses to rephrase the query using different keywords, this may be interpreted as negative feedback on the relevance or quality of the initial results. Exploiting this kind of user-behavior information can help to improve search result quality for individuals as well as entire communities by adding more "cognitive" elements to the search engine.

Our approach is based on a Markov-chain model with queries as additional nodes, additional edges that capture query refinements and result clicks, and corresponding transition probabilities. Similarly to Google's PageRank, the computation of stationary visiting probabilities yields the behavior-aware authority ranking. Our original model could not express negative feedback, as probabilities are non-negative and L1-normalized. To capture and exploit also negative assessment such as assigning trust levels to Web pages (e.g., marking a Web page as spam, low-quality, out-of-date, or untrusted), we are pursueing several extended approaches, one of which is based on a Markov reward model where the assessment part is uncoupled from the random walk in the extended Web graph. A page receives a specific lump reward each time a transition is made to it, and this reward depends on the transition's source and target and will be derived from the query-log and click-stream information as well as explicit page assessments. The reward itself can be positive or negative.

The most interesting measure in such reward models is the expected earned reward per time-step. Preliminary results with this approach are presented in [20].

3.3.2 Web Evolution

Today, virtually all Web repositories, including digital libraries and the major Web search engines, capture only current information. But the history of the Web, its lifetime over the last 15 years and many years to come, is an even richer source of information and latent knowledge. It captures the evolution of digitally born content and also reflects the near-term history of our society, economy, and science. Web archiving is done by the Internet Archive, with a current corpus of more than 2 Petabytes and a Terabyte of daily growth, and, to a smaller extent, some national libraries in Europe. These archives have tremendous latent value for scholars, journalists, and other professional analysts who want to study sociological, political, media usage, or business trends, and for many other applications such as issues of intellectual property rights. However, they provide only very limited ways of searching timelines and snapshots of historical information.

We are working on Web search and ranking of authoritative pages with a user-adjustable time focus, supporting both snapshots and timelines. Our notions of T-Rank and BuzzRank [9, 10] provide time-aware generalizations of the PageRank importance measure. As an example for the benefits obtained from time-aware ranking, consider looking for a database publication of the year 1993 with a strong trend of increasing popularity. When you apply static PageRank to Web sources (including portals such as DBLP), you obtain Codd's seminal paper as the best result, whereas BuzzRank identifies the Association-Rule-Mining paper by Agrawal, Imielinski, and Swami as the best emerging authority. As the underlying link matrices for these computations are huge, one of the challenges that we are addressing is to implement these new kinds of extended link analyses in an efficient and scalable manner. To this end, we are investigating compact representations for time-series of link graphs, so as to reduce both the space and time complexity of time-aware ranking and time-travel queries over Web archives.

3.3.3 Ontology Learning

Several of our projects, especially the TopX engine for XML IR, make use of ontologies and thesauri. The latter can be constructed, to some extent, from hand-crafted knowledge such as WordNet, but a larger-scale and self-maintaining, promising approach is to automatically learn the concepts and relations for an ontology directly from rich text corpora such as Wikipedia and other Internet sources. This requires a combination of linguistic analysis, pattern matching, and statistical learning to identify, for example, person names or locations and to extract instances of binary relations such as `located-at (city, river)`, `born-in (person, place)`, `plays-instrument (person, instrument)`, or the generic `is-instance-of (entity, concept)`.

For finding and extracting information on binary relations we pursue a novel approach based on a link-grammar representation of natural-language sentences and an SVM-based statistical learner for determining robust, generalizable linguistic patterns. This approach is implemented in the LEILA prototype system [31]. The method is almost unsupervised by starting with merely a small set of user-provided positive examples such as *(Paris, Seine), (Calcutta, Ganges), (London, Thames)* and either explicit or token-based negative examples such as *(New York, Mississippi)* or *(<proper noun>, <number>*. The system then automatically finds candidate patterns such as *<river> flows through <city>*, *<city> is located on the banks of the <river>*, or *<city> not only offers many cultural attractions, but visitors can also enjoy tubing or swimming in the <river>*. These patterns cannot be directly applied as they do not generalize well and would lead to many false positives. Instead, we run the CMU link-grammar parser [18] on the individual sentences, compute a characteristic feature vector from the resulting graph representation of each sentence, and feed these vectors into an SVM classifier. For higher recall, we also perform anaphora resolution across neighboring sentences. The method outperforms competitors in terms of the F1 measure (i.e., the harmonic mean of precision and recall). We are currently working on further extensions and better scalability.

3.3.4 Document Classification

Automatic classification of text documents is an important building block for many forms of intelligent data organization, for example: assigning Web pages to topics of a hierarchical directory, filtering news feeds or thematic subscriptions in digital libraries, steering a focused crawler towards Web regions that are more likely to contain pages that fall into the user's specific interest profile. Classifiers are models over high-dimensional feature spaces that are derived from supervised learning, with positive and negative examples as training input. The underlying methods for statistical learning such as SVM or Bayesian methods have become fairly mature, and the practical bottleneck is typically the scarceness of training data, because compiling training samples is a laborious and time-consuming human activity. Our research is primarily aiming at overcoming this training bottleneck and improving the accuracy and robustness of text classifiers.

We have developed a family of adjustable metamethods [26, 27, 28] based on ensemble learning that can be tuned towards the specific goals of the classifier: when precision and accuracy are critical we can prioritize the mutual agreement of multiple classifiers, but when recall is more important we can relax the settings of the meta-classifier. A particular application of these methods is our focused crawler BINGO! [29], which can start with a small set of training samples and seed URLs and then automatically finds characteristic "archetype" pages for dynamic and automated re-training. In this semisupervised-learning context, it is crucial to minimize the classification error, and restrictive metamethods can be effectively tuned towards this goal.

In a second line of research we investigate richer feature models and contextual features of training samples so as to improve classification accuracy with relatively small training sets. One approach is to consider neighbors of text documents in environments with many cross-references such as Web links (but covering also settings such as book, music, or blog recommendations or citation graphs for publications). The graph-based classifier presented in [2] builds on the theory of Markov Random Fields, and addi-

tionally develops new techniques for enhanced robustness. This method outperforms purely text-based classifiers and also all previously proposed link-aware classifiers. Finally, we investigate new models that map text features onto semantic concepts in an ontology and leverage the concepts as additional information. In [16] we have developed a generative probabilistic model that uses concepts as a latent-variable layer and an efficient EM-based parameter-estimation procedure. In contrast to previous work on latent semantic models, concepts are explicit and directly interpretable by humans and greatly simplify the model-selection problem of choosing the right number of latent dimensions. In combination with a transductive learning procedure that leverages the richer feature space of unlabeled background corpora, we have built a highly effective classifier that excels especially when training data is very scarce.

4. ACKNOWLEDGEMENTS

All members of the DBIS group at MPI-INF have been greatly contributing to our research results and ongoing projects. As of June 2006, the group has the following scientific staff and graduate students (born and raised in 7 different countries): Ralitsa Angelova, Srikanta Bedathur, Matthias Bender, Klaus Berberich, Andreas Broschart, Tom Crecelius, Jens Graupmann, Georgiana Ifrim, Gjergji Kasneci, Julia Luxenburger, Sebastian Michel, Thomas Neumann, Hanglin Pan, Josiane Parreira, Maya Ramanath, Ralf Schenkel, Stefan Siersdorfer, Fabian Suchanek, Martin Theobald, Gerhard Weikum, Christian Zimmer. In addition, several visitors and external collaborators have contributed, most notably, Peter Triantafillou and Michalis Vazirgiannis. Our projects have various funding sources. We are grateful to the German Science Foundation (DFG) and the Commission of the European Union (EU) for supporting the projects CLASSIX (DFG), DELIS (EU), and the Network of Excellence DELOS (EU).

5. REFERENCES

[1] S. Amer-Yahia et al.: Report on the DB/IR Panel at SIGMOD 2005, SIGMOD Record 34(4): 71-74, 2005.

[2] R. Angelova, G. Weikum: Graph-based Text Classification: Learn from your Neighbors, SIGIR 2006.

[3] H. Bast, D. Majumdar, R. Schenkel, M. Theobald, G. Weikum: IO-Top-k: Index-access Optimized Top-k Query Processing, VLDB 2006.

[4] M. Bender, S. Michel, P. Triantafillou, G. Weikum, C. Zimmer: Improving Collection Selection with Overlap Awareness, SIGIR 2005.

[5] M. Bender, S. Michel, P. Triantafillou, G. Weikum, C. Zimmer: Minerva: Collaborative P2P Search, Demo Paper, VLDB 2005.

[6] M. Bender, S. Michel, P. Triantafillou, G. Weikum: IQN Routing: Integrating Quality and Novelty in P2P Querying and Ranking, EDBT 2006.

[7] M. Bender, S. Michel, P. Triantafillou, G. Weikum, C. Zimmer: P2P Content Search: Give the Web Back to the People, IPTPS 2006.

[8] M. Bender, S. Michel, P. Triantafillou, G. Weikum: Global Document Frequency Estimation in Peer-to-Peer Web Search, WebDB 2006.

[9] K. Berberich, M. Vazirgiannis, G. Weikum: Time-aware Authority Ranking, Internet Mathematics 2(3), 2006.

[10] K. Berberich, S. Bedathur, M. Vazirgiannis, G. Weikum: BuzzRank . . . and the Trend is your Friend, WWW 2006.

[11] W.W. Cohen: MinorThird, http://minorthird.sourceforge.net/

[12] H. Cunningham: ANNIE – a Robust Cross-Domain Information Extraction System, http://gate.ac.uk/ie/annie.html

[13] R. Fagin, A. Lotem, M. Naor: Optimal Aggregation Algorithms for Middleware, Journal of Computer and System Sciences 66(4): 614-656, 2003.

[14] S. Golder, B. Huberman: The Structure of Collaborative Tagging Systems, Journal of Information Science, 2006.

[15] J. Graupmann, R. Schenkel, G. Weikum: The SphereSearch Engine for Unified Ranked Retrieval of Heterogeneous XML and Web Documents, VLDB 2005.

[16] G. Ifrim, G. Weikum: Transductive Learning for Text Classification using Explicit Knowledge Models, PKDD 2006.

[17] INEX 2006: Initiative for the Evaluation of XML Retrieval, http://inex.is.informatik.uni-duisburg.de/2006/

[18] J. Lafferty, D. Sleator, D. Temperley: Link Grammar, http://www.link.cs.cmu.edu/link/

[19] J. Li, B.T. Loo, J.M. Hellerstein, F. Kaashoek, D.R. Karger, R. Morris: On the Feasibility of Peer-to-Peer Web Indexing and Search, IPTPS 2003.

[20] J. Luxenburger, G. Weikum: Exploiting Community Behavior for Enhanced Link Analysis and Web Search, WebDB 2006.

[21] W. Meng, C.T. Yu, K.-L. Liu: Building Efficient and Effective Metasearch Engines, ACM Computing Surveys 34(1): 48-89, 2002.

[22] S. Michel, P. Triantafillou, G. Weikum: KLEE: A Framework for Distributed Top-k Query Algorithms, VLDB 2005.

[23] J.X. Parreira, D. Donato, S. Michel, G. Weikum: Efficient and Decentralized PageRank Approximation in a Peer-to-Peer Web Search Network, VLDB 2006.

[24] R. Schenkel, A. Theobald, G. Weikum: Efficient Creation and Incremental Maintenance of the HOPI Index for Complex XML Document Collections, ICDE 2005.

[25] R. Schenkel, M. Theobald: Structural Feedback for Keyword-Based XML Retrieval, ECIR 2006.

[26] S. Siersdorfer, S. Sizov: Restrictive Clustering and Metaclustering for Self-organizing Document Collections, SIGIR 2004.

[27] S. Siersdorfer, S. Sizov, G. Weikum: Goal-oriented Methods and Meta Methods for Document Classification and their Parameter Tuning, CIKM 2004.

[28] S. Siersdorfer, S. Sizov: Automatic Document Organization in a P2P Environment, ECIR 2006.

[29] S. Sizov, M. Theobald, S. Siersdorfer, G. Weikum, J. Graupmann, M. Biwer, P. Zimmer: The BINGO! System for Information Portal Generation and Expert Web Search, CIDR 2003.

[30] R. Steinmetz, K. Wehrle (Editors): Peer-to-Peer Systems and Applications, Springer, 2005.

[31] F. Suchanek, G. Ifrim, G. Weikum: Combining Linguistic and Statistical Analysis to Extract Relations from Web Documents, KDD 2006.

[32] A. Theobald, G. Weikum: Adding Relevance to XML, WebDB 2000.

[33] A. Theobald, G. Weikum: The XXL Search Engine: Ranked Retrieval of XML Data using Indexes and Ontologies, EDBT 2002.

[34] M. Theobald, G. Weikum, R. Schenkel: Top-k Query Evaluation with Probabilistic Guarantees, VLDB 2004.

[35] M. Theobald, R. Schenkel, G. Weikum: An Efficient and Versatile Query Engine for TopX Search, VLDB 2005.

[36] M. Theobald, R. Schenkel, G. Weikum: Efficient and Self-Tuning Incremental Query Expansion for Top-k Query Processing, SIGIR 2005.