Information Retrieval Research at UFMG

Nivio Ziviani[1], Marcos André Gonçalves[1], Edleno Silva de Moura[2],
Berthier Ribeiro-Neto[1], Altigran Soares da Silva[2], Adriano Veloso[1]

[1] Universidade Federal de Minas Gerais, Brazil
{nivio,mgoncalv,berthier,adrianov}@dcc.ufmg.br
[2] Universidade Federal do Amazonas, Brazil
{alti,edleno}@dcc.ufam.edu.br

Abstract. This article summarizes Information Retrieval (IR) research conducted at the Universidade Federal de Minas Gerais (UFMG), over more than a quarter of a century. The work of the UFMG IR group has covered some of the key areas in modern IR from crawling, indexing, compression and ranking methods to search engines and recommender systems. Further, its focus on addressing practical problems of relevance to society and on building prototypes to validate the proposed solutions has led to the spin-off of two key start-up companies in Brazil, one of them acquired by Google Inc. to become its R&D center for Latin America.

Categories and Subject Descriptors: H. Information Systems [**Information Storage and Retrieval**]: Document and Text Processing

Keywords: Information retrieval, Classification, Web search engines, Recommender systems

1. INTRODUCTION

This article presents a summary of the main activities of the Information Retrieval (IR) Research Group at the Universidade Federal de Minas Gerais (UFMG), over the last 27 years. In this period of more than a quarter of a century, the group has engaged in a variety of projects, all of them developed at the LATIN - Laboratory for Treating Information, located at the UFMG Computer Science Department. As a result, the group now combines a large experience in technology-based enterprises with a wide network of collaborators in Brazil and abroad.

With more than seventy graduate students formed, many of them occupying key positions in academia and industry, and a large scientific production spread across many of the major IR journals and conferences, the group has established a solid reputation as a world class research group in its topics of interest. Today, the group is focused on exploring technologies related to web IR and their application to practical problems—a trademark of the group over the years, which has led to the successful spin-off of two key Web start-up companies in Brazil, one of them acquired by Google.

The research results described here cover the following topics: IR models for web search, automatic text classification, web search engines, and semantic enrichment. Section 2 reviews the group's history. Next, Section 3 presents the group profile and Section 4 details the current research lines developed by the group. Finally, Section 5 discusses perspectives and future directions for the group's activities.

This work was partially sponsored the Brazilian National Institute of Science and Technology for the Web (grant MCT/CNPq 573871/2008-6), by UOL (www.uol.com.br), through its UOL Bolsa Pesquisa program, and authors individual grants and scholarships from CNPq.

2. GROUP HISTORY

The research activities on the application of IR techniques to natural language texts started in June 1984 when Nivio Ziviani, the group's principal investigator, visited Gaston Gonnet at the Computer Science Department of the University of Waterloo. At the time, the algorithms and data structures group of the University of Waterloo signed a 10 years contract with the Oxford English Dictionary (OED) to computerize the dictionary then comprising 21,000 pages and 600,000 word definitions. The OED project was coordinated by Gaston Gonnet and Frank Tompa. Many important results on algorithms to search natural language texts came out of that project.

Upon return to his visit to Waterloo, Nivio Ziviani started the UFMG Text Research Group still in 1984 to study efficient algorithms to retrieve information from natural language texts. In 1987, this group built the PatPlus text search prototype [Ziviani and Albuquerque 1987; Ziviani 1991], which used suffix arrays [Gonnet et al. 1992] for indexing documents. The PatPlus system was used by librarians to index and search documents at the UFMG Central Library. The UFMG Text Research Group eventually gave birth to the Laboratory for Treating Information (LATIN) in 1994.

At the time, in Waterloo, Gaston Gonnet was launching the Open Text search engine[1] inside the start-up Open Text Corporation, which was founded in 1991 as a spin-off of the OED project. The search engine, which was one of the first search engines for the Web, used suffix arrays for indexing the web pages [Gonnet et al. 1992]. Since then, the Open Text Corporation has grown to become the largest Canadian company in IT technology.

In 1989, our group started a fruitful cooperation with Ricardo Baeza-Yates and his group at University of Chile, with the first joint paper published in 1990 [Baeza-Yates et al. 1990]. Since then Ricardo Baeza-yates and Nivio Ziviani have published together close to 40 papers in conference proceedings and journals. In 1993, they co-founded SPIRE (International Symposium on String Processing and Information Retrieval) whose first edition was held at UFMG and the eighteenth one will be held on October 17th, 2011 in Pisa, Italy. In 2005, they co-chaired the most important IR conference, the 2005 ACM SIGIR, in Salvador, Brazil. They also participated in the RITOS[2] and AMYRI[3] projects funded by the Spanish agency CYTED in the 1990s.

In 1995, Berthier Ribeiro-Neto joined UFMG and LATIN, bringing his experience in core IR and ranking to the group. In 1998, he took a key role in the writing of the SIAM - Sistemas de Informação em Ambientes de Computação Móvel (Mobile Environment Information Systems) project, which received long term funding from CNPq/PRONEX, the Brazilian National Research Council, with funding close to one million dollars. In 1999, jointly with Ricardo Baeza-Yates, Berthier Ribeiro-Neto published the book Modern Information Retrieval [Baeza-Yates and Ribeiro-Neto 1999], with some chapters written in collaboration with researchers from our group (e.g., [Ziviani 1999; Navarro and Ziviani 2011; Gonçalves 2011a; 2011b]). This is the one of most cited publication in the history of IR, with 8,442 citations at the time of this writing, according to Harzing's Publish or Perish[4]. In fact, it is one of the most cited books in the whole computer science field. A revised and greatly expanded second edition of the book has just been published [Baeza-Yates and Ribeiro-Neto 2011].

From 1986 to 2010, Ziviani published a series of five books on the design of algorithms and data structures[5] [Ziviani 1986; 1993; 2004; 2007; 2010]. These five books cover algorithms on text searching, sorting, text compression, hashing, and perfect hashing, among other topics. Further, they also provide various useful basic algorithms and their associated programs, which can be used as a basis to build search engine prototypes.

[1] http://www.opentext.com/2/global/company/company-history.htm
[2] http://sunsite.dcc.uchile.cl/cyted/ritos.html
[3] http://www2.dcc.ufmg.br/laboratorios/latin/amyri/
[4] http://www.harzing.com
[5] http://www.dcc.ufmg.br/algoritmos/

In 1998, we launched the first Brazilian search engine called MINER, based on meta-searching. It was built as part of Victor Ribeiro's Master Thesis [Ribeiro 1998]. In April 1998, this work led to the creation of the start-up Miner Technology Group, one of the first web technology companies in Brazil, which was sold to the group Folha de São Paulo/UOL[6] in June 1999. This was one of the first experiences in spinning off a Web start-up company from research conducted at a Brazilian university, showing that research results can be transferred to society by creating knowledge intensive start-ups.

In November 1999, an important sequel happened with the launching of the TodoBR search engine—a vertical search engine for the Brazilian web—within the scope of the SIAM research project. In April 2000, professors Ivan Campos, Alberto Laender, Berthier Ribeiro-Neto and Nivio Ziviani and venture capitalists Guilherme Emrich and Marcus Regueira co-founded the start-up Akwan Information Technologies[7], based on the TodoBR search technology. We observe that this type of knowledge intensive technology is dependent on high quality research, usually derived from PhD dissertations and MSc theses. Akwan, which became a successful start-up and a reference for web search in Brazil, was acquired by Google Inc. in July 2005—an acquisition that became worldwide news. With Akwan, Google bootstrapped its R&D Center for Latin America, which is located in Belo Horizonte.

In 2006, in collaboration with the UFMG Database Group headed by Alberto Laender, we developed at LATIN the Perfil-CC Project[8] with the objective of assessing the research and education quality of the top Brazilian Computer Science (CS) graduate programs [Laender et al. 2008]. Within that project, we conducted a study of the scientific production of these programs in the 2004-2006 triennium. That study, which compared the scientific production of the Brazilian programs against that of reputable programs in North America and Europe, was based on data from DBLP - Digital Bibliography & Library Project[9]. The results suggest that the scientific production of the top CS graduate programs in Brazil compares well with that of key European and American programs, both in terms of publication rates and number of graduates. Indeed, the study shows that the Brazilian programs follow international publication rates of more than two conference papers per journal article. That is, the results provide a clear indication that the CS field has reached maturity in Brazil. Most important, in a country whose scientific community has long been led by more traditional areas of knowledge, these results gave prestige to the CS area in Brazil. As an immediate consequence, CAPES, a Brazilian Ministry of Education's agency, classified the top CS programs in Brazil at the same level of top programs in more traditional areas of knowledge, a key development which will ensure a renewed influx of research resources such as grants, students and facilities.

Following LATIN's tradition of promoting and exploring the transfer of research results and technology to society, in 2008, we co-founded Zunnit Technologies[10]—a new start-up company focused on software for recommending items of interest to Web users. One year later, in 2009, invited by the IR research group at Universidade Federal do Amazonas (UFAM), Ziviani co-founded Nhemu Technologies[11]—a price comparison enterprise focused on presenting product offers to Web users.

3. GROUP PROFILE

Table I lists the main LATIN researchers. The LATIN permanent researchers are Marcos Gonçalves, Berthier Ribeiro-Neto and Nivio Ziviani, all from UFMG. The LATIN associate researchers are from Brazil, Portugal, Spain and USA. Four of these researchers are members of the Brazilian Academy of Sciences (Alberto Laender and Nivio Ziviani as permanent members and Edleno Silva de Moura and Marcos Gonçalves as affiliate members).

[6]http://www.uol.com.br
[7]http://www.akwan.com.br
[8]http://www.latin.dcc.ufmg.br/perfilcc
[9]http://dblp.unitrier.de
[10]http://www.zunnit.com.br
[11]http://www.nhemu.com.br

Table I. LATIN Researchers.

Permanent Researchers		Associate Researchers	
Marcos A. Gonçalves	UFMG, Brazil	Ricardo Baeza-Yates	Yahoo! Barcelona & UPF, Spain
Berthier Ribeiro-Neto	UFMG and Google, Brazil	Pavel Calado	IST, Portugal
Nivio Ziviani (Coordinator)	UFMG, Brazil	Fabiano C. Botelho	EMC^2, USA
		Alberto H.F. Laender	UFMG, Brazil
		Wagner Meira Jr	UFMG, Brazil
		Edleno S. de Moura	UFAM, Brazil
		Altigran S. da Silva	UFAM, Brazil
		Adriano Veloso	UFMG, Brazil

The LATIN team also includes undergraduate, MSc and PhD students. LATIN has also received Post Docs and visiting researchers from institutions abroad. The permanent researchers have supervised more than 70 graduate students by August 2011, being 58 MSc and 13 PhD students.

The LATIN group has a long time tradition of cooperation with foreign institutions, such as the Instituto Superior Técnico in Lisbon (Portugal), University of Chile, University Pompeu Fabra (Spain), Virginia Tech (USA), Yahoo! Barcelona, and Yahoo! Santiago (Chile). In Brazil, we have a strong collaboration with the database and IR group from UFAM. Also important is the collaboration with the Universidade Federal do Rio Grande do Sul (UFRGS), Universidade de Campinas (UNICAMP) and Universidade de São Paulo (USP).

4. RESEARCH AREAS

This section presents the main results in the group's research areas: information retrieval models, automatic text classification, web search engines and semantic enrichment.

4.1 Information Retrieval Models

Models are at the core of IR systems. They determine the accuracy in providing relevant answers to the users, and are also the technological basis of the main component of a IR system, the query processor. Hence, we have focused in *developing new IR models* [Ahnizeret et al. 2004; Coelho et al. 2004; Fonseca et al. 2004; Pôssas et al. 2005; Vale et al. 2003; Fernandes et al. 2007].

4.1.1 *Set-Based Model.* We have developed a model that combines data mining and traditional information retrieval models [Pôssas et al. 2002; Pôssas et al. 2004; Pôssas et al. 2005]. It presents a new approach for ranking documents in the vector space model. Its novelties are: (*i*) patterns of term co-occurrence are considered and processed efficiently; (*ii*) term weights are generated using a data mining technique called association rules, which leads to a new ranking mechanism called *set-based vector model*. The components are no longer index terms but index *termsets* (sets of index terms). Termsets capture the intuition that semantically related terms appear close to each other in a document. They can be efficiently obtained by limiting the computation to small passages of text. Once termsets have been computed, the ranking is calculated as a function of the termset frequency in the document and its scarcity in the document collection.

The set-based model is the first IR model that exploits term correlations and term proximity effectively, providing significant gains in terms of precision, regardless of the sizes of the collection and the vocabulary. All known approaches for correlation among index terms were initially designed for processing only disjunctive queries. The set-based model advanced the state-of-the-art and provides a simple, effective, efficient and parameterized way to process disjunctive, conjunctive, and phrase queries. Experimental results show that the set-based vector model improves average precision for all collections and query types evaluated, while keeping computational costs small.

4.1.2 *Hypergraph Model.* Also related to web search, we proposed a representation of the web as a directed hypergraph, instead of a graph, where links can connect both pairs of pages and pairs of

disjoint sets of pages [Berlt et al. 2010]. Here, the web hypergraph is derived from the web graph by dividing the set of pages into non-overlapping sets and using the links between pages of distinct sets to create hyperarcs. Each hyperarc connects a set of pages to a single page and is created for providing more reliable information to link analysis methods. We used the hypergraph structure to compute the reputation of web pages by experimenting hypergraph versions of two link analysis methods, PageRank and Indegree [Brin and Page 1998]. Experimental results indicate that the hypergraph versions of PageRank and Indegree produce better results when compared to their original graph versions.

4.1.3 *Web Data Mining.* Another interesting work developed by our group is WIM – Web Information Mining [Pereira-Jr et al. 2009], a model for fast Web mining prototyping. The underlying conceptual model of WIM provides its users with a level of abstraction appropriate for prototyping and experimentation throughout the Web data mining task. Abstracting from the idiosyncrasies of raw Web data representations facilitates the inherently iterative mining process. WIM incorporates a set of conceptual modeling primitives, an associated algebra with a set of operators, and a software tool that implements the model. The experimentation of WIM in real use cases has shown to significantly facilitate Web mining prototyping. For example, in [Baeza-Yates et al. 2008] we use WIM to study about the evolution of textual content on the Web. That is, how some new pages are created from scratch while others are created using already existing content. We show that a significant fraction of the Web is a byproduct of the latter case.

4.1.4 *Learning to Rank.* Several empirical ranking methods such as boolean models, vector space models and probabilistic models have been proposed in the literature [Baeza-Yates and Ribeiro-Neto 2011]. Due to the difficulty in empirically tuning the parameters of the ranking functions that are obtained from the above methods, state-of-the-art search engines are recently adopting alternate methods which are derived from machine learning techniques. These methods automatically learn effective ranking functions and are regarded as *learning to rank* methods [Liu 2010; Trotman 2005].

We have been developing learning to rank algorithms and we have advanced the state-of-the-art in several ways: (i) producing collection-adapted ranking functions; (ii) devising associative learning to rank algorithms; (iii) devising active sampling strategies for ranking; and (iv) building rank aggregation approaches. Our initial research in this area was focused on discovering specialized ranking strategies for specific collections. We propose a method which is able to consider the important and unique characteristics of each collection so that the discovered function is more effective than any general solution [de Almeida et al. 2007]. To accomplish this, we use genetic programming (GP) to discover specific ranking functions for each document collection. Experiments were performed using the TREC-8 and WBR-99 collections. They show that our combined component approach improves the retrieval performance compared to standard TF-IDF, BM25 and other GP-based approaches [Fan et al. 2004] that use only basic statistical information derived from collections and documents.

We also proposed a new learning strategy for ranking called *associative learning to rank* [Veloso and Meira Jr. 2011]. Specifically, ranking models are composed of a structure called association rules [Agrawal et al. 1993], and the key advantage of using this structure is that ranking models can be built on-the-fly efficiently as new queries arrive. Using this strategy, we show that further improvements in learning performance are still possible by enabling the use of additional information while generating the model, namely the query terms [Veloso et al. 2008]. After, other researchers followed this trend by incorporating query-level information in order to learn better-quality ranking models [Bian et al. 2010; Lan et al. 2008; Geng et al. 2008]. We then defined other features related to the query, such as rule stability and ranking competence [Veloso et al. 2010], and use these query-level features in order to further improve ranking performance. We performed a deep analysis involving the state-of-the-art and showed that *associative algorithms* still offer superior ranking performance, while being extremely fast and highly practical [Alcântara et al. 2010]. We also used associative algorithms in other application scenarios, such as content-based IR [Faria et al. 2010].

We have developed active learning alternatives for reducing the human annotators' labeling effort by selectively sampling a set of unlabeled examples. Our algorithms select new documents to be labeled based on the number of association rules they demand, given the previously selected and labeled examples [Silva et al. 2011]. In contrast to previous work, our algorithms have a clear stop criterion and do not need to empirically discover the best configuration by iteration on validation sets.

4.2 Automatic Text Classification

Automatic text classification is one of the major information retrieval problems. Its goal is to create models capable of associating documents with semantically meaningful categories. Automatic text classification has been successfully employed, for instance, to design ad-matching systems, to organize digital libraries and web directories, for (personalized) recommendation, analysis of streams of news, spam filtering, among many other practical applications. Automatic text classification algorithms usually employ a supervised learning strategy, where a classification model is first built using a set of pre-classified documents, i.e., a training set, which is then used to classify unseen documents.

We have worked in automatic text classification for almost a decade and followed several directions, including: (i) exploiting link-based information to improve the classification effectiveness; (ii) estimating credibility of examples; and (iii) dealing with the temporal dynamics of textual collections. The first set of work focus on exploiting bibliographic metrics such as co-citation, bibliographic coupling and Amsler alone and in combination with traditional text-based classifiers for the classification task. Early work focused on web directories [Calado et al. 2003; Calado et al. 2006], producing high levels of classification effectiveness (around 90%) which is impressive for a noisy environment such as the Web. Later work demonstrated that similar results could be obtained in other scenarios where hyperlinked information also existed such as those provided by citations among scientific documents and references among encyclopedia articles [Couto et al. 2006; Couto et al. 2010]. A comparative study on the nature of these link-based classifiers and their respective strengths and weaknesses was also performed.

Automatic text classification algorithms usually assume that all examples in the training set are equally important for generating the classification model, which is not always the case. For instance, the contribution of a document may vary according to many factors, such as its actual content, citations, authorship, time of publication, among others. Accordingly, in [Palotti et al. 2010], we investigate how to estimate a credibility function for documents using their content, citations and authorship. Particularly for the case of document content, we propose a genetic programming algorithm to estimate the strength of term-class relationship based in the combination of a number of metrics (e.g., TF-IDF, dominance, information gain, chi-square, among others).

Another assumption of basically all automatic text classification algorithms is that the data used to learn a classification model are random samples independently and identically distributed from a stationary distribution that governs the test data. When this does not hold, the classification effectiveness may be compromised. We have characterized [Mourão et al. 2008] and quantified [Salles 2011] in real collections what we have called temporal effects that can negatively impact the automatic text classification task. Temporally-aware algorithms for automatic text classification [Salles et al. 2010; Salles et al. 2010] have been proposed in order to properly handle the temporal effects. The algorithms incorporate temporal information to document classifiers, aiming at improving their effectiveness by handling data governed by varying distributions. Extensive experimental evaluation with large real-world collections spanning more than 20 years of publications showed that these classifiers were able to significantly improve the classification effectiveness when facing with data varying distributions.

4.3 Web Search Engines

4.3.1 *Crawling the Web.* The Web has become a huge repository of pages and search engines allow users to find relevant information in this repository. Web crawlers are an important component of search engines. They find, download, parse content and store pages in a repository.

We proposed a new crawler architecture with the following main components: fetcher, URL extractor, URL uniqueness verifier and scheduler [Henrique 2011]. The fetcher is the component that sees the Web. It receives from the scheduler a set of URLs to download web pages. The downloaded pages are then sent to the extractor of URLs, which parses each one and obtains a set of URLs to be crawled. Among those, many URLs may be already crawled. Thus, we also need to develop a uniqueness verifier to check each URL against a large repository of unique URLs.

The crawler completes a cycle with the scheduler choosing a new set of URLs to be sent to the fetcher. The interval between two accesses to the same server must follow politeness rules, which might cause a significant slowdown in the whole process. We have developed new scheduling strategies that minimize delays related to politeness by using an extremely low cost scheduling algorithm. In this case the set of unique URLs are organized according to the server they belong to, exploiting a locality of reference property. We also proposed a new algorithm for verifying URL uniqueness in a large-scale web crawler [Henrique et al. 2011]. We performed experiments to compare our algorithm with a state-of-the-art algorithm found in the literature. The results up to now indicate that our new proposal yields a significant reduction in the time spent handling URL uniqueness verification.

4.3.2 *Indexing.* The usually large size of a search engine textual repository demands specialized indexing techniques for efficient retrieval. The two most important indexing methods are suffix arrays and inverted files. Suffix arrays are more adequate to applications where the problem of efficient substring searching arises, such as computational biology (protein sequences) or music databases. Inverted files are the preferred choice to implement IR systems. They are composed of two elements: the vocabulary (the set of all different words in the text) and the occurrences (for each word in the vocabulary the index stores the documents which contain the word).

Suffix Arrays. Suffix array [Manber and Myers 1993] or pat array [Gonnet et al. 1992] is a linear structure composed of index pointers to every suffix in the text. Each text position is considered as a text suffix (i.e., a string from there to the end of the text). In IR, since the user normally bases his queries upon words and phrases, it is customary to index only word beginnings. The index pointers are sorted according to a lexicographical ordering of their suffixes and each index point can be viewed simply as the offset (counted from text beginning) of its corresponding suffix in the text. To find the user patterns, binary search is performed on the array at $O(\log n)$ cost, where n is the text size.

We have presented an efficient implementation of pat arrays (or suffix arrays) when the database is stored on secondary storage devices such as magnetic or optical disks [Baeza-Yates et al. 1996]. The implementation uses additional index structures and searching algorithms that reduce the number of disk accesses. The index structures are: a two-level hierarchy model that uses main memory and one level of external storage (magnetic or optical devices) and a three-level hierarchy model that uses main memory and two levels of external storage (magnetic and optical devices). Performance improvement is achieved in both models by storing most of higher index levels in faster memories, thus reducing accesses in the slowest devices in the hierarchy.

In a more theoretical work, we studied the problem of minimizing the expected cost of binary searching for data where the access cost is not fixed and depends on the last accessed element, such as data stored in magnetic or optical disk [Barbosa et al. 1995; Navarro et al. 2000]. We have presented an optimal algorithm that finds the optimal search strategy in $O(n^3)$ time (i.e, the same time complexity of the simpler classical problem of fixed costs). Next, we presented two practical linear expected time algorithms, under the assumption that the access cost of an element is independent of its physical position. Both practical algorithms are online, that is, they find the next element to access as the search proceeds. We presented an application for our algorithms related to text retrieval using suffix arrays, where data access is provided through an indirect binary search on the text stored in magnetic disk or optical disk. Under this cost model we proved that the optimal algorithm cannot perform better than $\Omega(1/\log n)$ times the standard binary search. We also proved that the approximate

strategy cannot, on average, perform worse than 39% over the optimal one. Another theoretical work on binary search trees with costs depending on the access paths is in [Szwarcfiter et al. 2003].

We also have a series of works on parallel generation of suffix arrays [Kitajima et al. 1997; Navarro et al. 1997]. In [Navarro et al. 1997] we present an algorithm for the distributed computation of suffix arrays for large texts. The parallelism model is that of a set of sequential tasks which execute in parallel and exchange messages among them. The underlying architecture is that of a high bandwidth network of processors. Our algorithm builds the suffix array by quickly assigning an independent subproblem to each processor and completing the process with a final local sorting. We demonstrate that the algorithm has time complexity of $O(b \log n)$ computation and $O(b)$ communication in the average case, where b corresponds to the local text size on each processor (i.e., text size n divided by r, the number of processors). This is faster than the best known sequential algorithm and improves over previous parallel algorithms to build suffix arrays, both in time complexity and scaling factor.

Inverted Files. Inverted files have been traditionally the most popular indexing techniques. They are useful because their searching strategy is based mostly on the vocabulary which usually fits in main memory. Further, inverted files perform well when the pattern to be searched for is formed by conjunctions and disjunctions of simple words, common types of queries in search web systems.

In [Ribeiro-Neto et al. 1999] we present three distributed algorithms to build global inverted files for very large text collections. The distributed environment is a high bandwidth network of workstations with a shared-nothing memory organization. The text collection is assumed to be evenly distributed among the disks of the workstations. Our algorithms consider that the total distributed main memory is much smaller than the inverted file to be generated. The inverted file was compressed to save memory, disk space, and time for moving data in/out disk and across the network.

Detecting Replicated Web Sites. Web site replication occurs when multiple sites are similar in terms of content and structure. Replicas may appear because of several reasons, including: (i) the same for-sale items on e-commerce web sites, (ii) web sites that are pre-built and sold to multiple people, (iii) when the web site is moving to another hosting company, (iv) web sites getting both www and non-www versions indexed, or (v) web sites that are mirrored for the sake of load balancing. There is also a malicious reason for replicating web sites: many people assume that creating multiple or similar copies of the web site will either increase chances of getting listed in search engines or help them get multiple listings, due to the presence of more keywords.

However, being intentionally created or not, the fact is that replicated web sites challenge the effectiveness of search engines, either because (i) the same content should not be present in their search results, (ii) resources should not be spent in indexing web sites that are substantially similar, or (iii) replicas insert duplicated connectivity information in web collections, causing anomalies in ranking algorithms [Calado et al. 2003].

A general solution to the problem of detecting web site replicas is to adopt heuristics [Bharat et al. 2000] in order to select pairs of web sites that are suspect to be replicas, and then performing the detailed content/structure comparison only for these pairs. The quality of a replica detection method is thus directly related to the accuracy of heuristics in finding such suspect pairs.

Previous solutions to find replicated web sites, however, do not take content information into account, while selecting suspect pairs. The claim is that inspecting content turns the process unacceptably expensive due to the quadratic nature associated with pair-wise comparison. We propose an alternative method [da Costa Carvalho et al. 2007] that finds replica-candidate pairs by taking the advantages of using content, while not increasing processing time. Our method improves the quality of the replica-candidate detection task in 47.23% when compared to previously proposed methods, being extremely useful in practice.

Recently we reported impressive detection improvements with approaches based on the application of classification techniques [Guidolini 2011]. Our approaches have a series of advantages. The first one is related to the difficulty of precisely defining what is replica and what is not. Most definitions are based on the perceptual meaning of the web sites, being fuzzy by nature, in the sense that it is necessary to delimitate the amount of duplicate content necessary to characterize replication. On the other hand, following the classification strategy, we need only to provide sufficient training examples, and the classifier automatically learns to differentiate pairs that are replicas from those that are not. A second advantage is related to the difficulty of manually combining multiple evidence of replication. On the other hand, following the classification strategy, the classifier selects the best available features and accomplish example-driven ways to combine them. We propose a set of discriminative features, and show that the use of classifiers may improve detection performance by 48%.

4.3.3 *Query Processing.*

Query Expansion. One of the key difficulties in query searching is the fact that users usually submit very short and ambiguous queries, and they do not fully specify their information needs. That is, it is necessary to improve the query formation process if better answers are to be provided. In [Fonseca et al. 2005] we propose a concept-based query expansion technique, which allows disambiguating queries submitted to search engines. The concepts are extracted by analyzing and locating cycles in a special type of query relations graph. This is a directed graph built from query relations mined using association rules. The concepts related to the current query are then shown to the user who selects the one concept that he interprets is most related to his query. This concept is used to expand the original query and the expanded query is processed instead. Using a Web test collection, we show that our approach leads to gains in average precision figures of roughly 32%. Further, if the user also provides information on the *type* of relation between his query and the selected concept, the gains in average precision go up to roughly 52%. In Section 4.4.3 we discuss another query expansion method that exploits the entity semantics available in Wikipedia content and structure.

Distributed Query Processing. One of our main efforts is to develop new distributed query processing strategies for search engines. In [Badue et al. 2001] we present a real distributed architecture implementation that offers concurrent query service. In [Badue et al. 2005] we study three basic and key issues related to distributed web query processing: load balance, broker behavior, and performance by individual index servers. Our study reveals interesting tradeoffs: (1) load unbalance at low query arrival rates can be controlled by randomizing the distribution of documents among the index servers, (2) the broker is not a bottleneck, and (3) disk utilization is higher than CPU utilization.

In [Badue et al. 2006] we modeled workloads for a web search engine from a system performance point of view, analyzing both the distribution of the inter-arrival times of queries and the per-query execution time. This is crucial for performance evaluation and capacity planning purposes. We verified in practice that there is a high variability in both inter-arrival times of queries reaching a search engine and service times of queries processed in parallel by a cluster of index servers. We also showed that this highly variable behavior of inter-arrival times and service times is very well captured by hyper-exponential distributions.

The performance of parallel query processing in a cluster of index servers is crucial for modern web search systems. We found in [Badue et al. 2007] that even with a balanced distribution of the document collection among index servers, correlations between the frequency of a term in the query log and the size of its corresponding inverted list lead to imbalances in query execution times at these same servers, because these correlations affect disk caching behavior. Further, the relative sizes of the main memory at each server (with regard to disk space usage) and the number of servers participating in the parallel query processing also affect imbalance of local query execution times. These are relevant findings that have not been reported before and are of interest to the research community.

Predicting the response time of a vertical search engine is usually done empirically through experimentation, requiring a costly setup. An alternative is to develop a model of the search engine for predicting performance. In [Badue et al. 2010] we propose a methodology for analyzing the performance of vertical search engines. Applying the proposed methodology, we present a capacity planning model based on a queueing network for search engines with a scale typically suitable for the needs of large corporations. The model is simple and yet reasonably accurate and, in contrast to previous work, considers the imbalance in query service times among homogeneous index servers. We discuss how we tune up the model and how we apply it to predict the impact on the query response time when parameters such as CPU and disk capacities are changed. This allows a manager of a vertical search engine to determine a priori whether a new configuration of the system might keep the query response under specified performance constraints.

Link Analysis. Information derived from the cross-references among the documents in a hyperlinked environment, usually referred to as link information, is considered important since it can be used to effectively improve document retrieval. Depending on the retrieval strategy, link information can be local or global. Local link information is derived from the set of documents returned as answers to the current user query. Global link information is derived from all the documents in the collection.

In [Silva et al. 2000; Calado et al. 2003] we investigate how the use of local link information compares to the use of global link information. For the comparison, we run a series of experiments using a large document collection extracted from the Web. For our reference collection, the results indicate that the use of local link information improves precision by 74%. When global link information is used, precision improves by 35%. However, when only the first 10 documents in the ranking are considered, the average gain in precision obtained with the use of global link information is higher than the gain obtained with the use of local link information. This is an interesting result since it provides insight and justification for the use of global link information in major web search engines, where users are mostly interested in the first 10 answers. Further, global information can be computed in the background, which allows speeding up query processing.

4.3.4 *Efficiency Issues in Web Search.* IR systems need to be not only highly effective but also extremely efficient in terms of time and space, since query throughput and a huge number of documents are central problems in these systems.

Text Compression. Our discussion here focuses on text compression methods that are suitable for use in an IR environment. By suitable means to access text randomly and allow searching the compressed directly and faster than the uncompressed text, something that was not possible to perform efficiently until the work presented in [de Moura et al. 1998; 2000]. We presented a fast compression and decompression technique for natural language texts. The novelties are that (i) decompression of arbitrary portions of the text can be done very efficiently, (ii) exact search for words and phrases can be done on the compressed text directly, using any known sequential pattern matching algorithm and (iii) word-based approximate and extended search can also be done efficiently without any decoding. The compression scheme uses a semi-static word-based model and a Huffman code where the coding alphabet is byte-oriented rather than bit-oriented. We compress typical English texts to about 30% of their original size, against 40% and 35% for *Compress* and *Grip*, respectively. Compression time is close to that of *Compress* and approximately half the time of *Gzip*, and decompression time is lower than that of *Gzip* and one third of that of *Compress*.

We present in [de Moura et al. 2000] three algorithms to search the compressed text. When searching for simple words, the experiments show that running our algorithms on a compressed text is twice as fast as running the best existing software on the uncompressed version of the same text. When searching complex or approximate patterns, our algorithms are up to 8 times faster than the search on uncompressed text. We also discuss the impact of our technique in inverted files pointing to logical blocks and argue for the possibility of keeping the text compressed all the time, decompressing only for displaying purposes, as discussed in the following section on index compression.

Index and Text Compression. We also have studied in past years alternatives to inverted index compression and text compression considered simultaneously. For instance, in [Navarro et al. 2000; Ziviani et al. 2000], we combine index compression to block addressing and sequential search on compressed text. For the problem of creating a large text database and providing fast access through keyword searches, compressing both the index and the text cuts the total space in half. The time required to build the index and answer a query is far less than if the index and text had not been compressed. This illustrates a rare case where there is no space-time trade-off.

Thus, inverted index compression obtains significant reduction of their original size at the same processing speed. Block addressing makes the inverted lists point to text blocks instead of exact positions and pay the reduction in space with some sequential text scanning. We combined these ideas in a single scheme, presenting a compressed inverted file that indexes compressed text and uses block addressing. This block addressing scheme allows fast and flexible search on large textual databases.

Caching. In [Saraiva et al. 2001] we present an effective caching scheme that reduces the computing and I/O requirements of a Web search engine without altering its ranking characteristics. The novelty is a two-level caching scheme that simultaneously combines cached query results and cached inverted lists on a real case search engine. A set of log queries are used to measure and compare the performance and the scalability of the search engine with no cache, with the cache for query results, with the cache for inverted lists, and with the two-level cache. Experimental results show that the two-level cache is superior, and that it allows increasing the maximum number of queries processed per second by a factor of three, while preserving the response time. These results are new, have not been reported before, and demonstrate the importance of advanced caching schemes for real case search engines.

Perfect Hashing Methods. The design of new hashing methods for *static sets of keys* is strongly related to the generation of indexes for IR systems, since a significant portion of the time is spent in hash operations. A *hash function* is a mapping from a key universe U to a range of integers, i.e., $h: U \mapsto \{0, 1, \ldots, m-1\}$, where m is the size of this range. A *perfect hash function* for some set $S \subseteq U$ is a hash function that is one-to-one on S, where $m \geq |S|$. A *minimal perfect hash function* (MPHF) for some set $S \subseteq U$ is a perfect hash function with a range of minimum size, i.e., $m = |S|$.

A MPHF totally avoids the common problem of wasted time (that is, any key is found in one probe) and space (that is, there are no empty entries in the hash table). For applications with only successful searches (i.e., when the queried key is always found in the key set), a key is simply represented by the value of a MPHF and the key set is not needed to locate information related with the key. MPHFs are used for memory efficient and fast retrieval of items from static sets, such as words in natural languages, reserved words in programming languages or interactive systems, universal resource locations in web search engines, or itemsets in data mining techniques.

We proposed a construction for MPHFs that combines theoretical analysis, practical performance, expected linear constructing time and nearly optimal space consumption for the data structure [Botelho et al. 2005; Botelho et al. 2007; Botelho and Ziviani 2007; Botelho et al. 2008]. For n keys and $m = n$ the space consumption ranges from $2.62n$ to $3.3n$ bits, and for $m = \lceil 1.23n \rceil$ it ranges from $1.95n$ to $2.7n$ bits. This is within a small constant factor from the theoretical lower bounds of $1.44n$ bits for $m = n$ and $0.89n$ bits for $m = \lceil 1.23n \rceil$.

The methods proposed in [Botelho et al. 2007] and [Botelho and Ziviani 2007] give a mathematical basis for (minimal) perfect hashing and also leads to real constructions for key sets of size in the billions that use much less space than all previously known algorithms. Such practical constructions were not known before. Experimental results show that we can construct MPHFs for 1.024 billion keys in 46 minutes using a commodity PC. A distributed and parallel version of the algorithm presented in [Botelho et al. 2008] constructs a MPHF for the same set of 1.024 billion URLs using 14 commodity PCs in approximately 4 minutes.

After the new results presented in [Botelho et al. 2007], we show in [Botelho et al. 2011] that MPHFs are a good option to index internal memory when static key sets are involved and both successful and unsuccessful searches are allowed. They provide the best tradeoff between space usage and lookup time when compared with other open addressing and chaining hash schemes such as linear hashing, quadratic hashing, double hashing, dense hashing, cuckoo hashing, sparse hashing, hopscotch hashing, chaining with move-to-front heuristic and exact fit.

An open source implementation of the algorithms is available in the C Minimal Perfect Hashing Library (CMPH)[12] under the GNU Lesser General Public License (LGPL). The CMPH code has been downloaded more than 6,500 times by August 2011 and was incorporated by the Debian and Ubuntu Linux distributions, which indicates how useful the results are in practice.

4.4 Semantic Enrichment in IR

Traditionally, several IR problems have been addressed by using a matching approach, in which objects to be matched are represented as *bags of words* (*BOWs*). The BOWs are used to build TF-IDF weighted term vectors and a *ranking* of results is generated based on the similarity between objects. This similarity can be computed using the cosine measure or one of its variations. The list of relevant answers corresponds to the top-K items in the ranking.

There are many situations, however, for which the simplistic representation based on BOW leads to poor results. This problem occurs in cases where the semantic relationship between objects to be matched cannot be properly captured by simply matching the terms used to describe them. For instance, in a content-based recommender system, the user profile might include the word "soccer", whereas a description of a book about soccer would miss the word "soccer" but includes the synonym "football" or the related word "league". Thus, the simple matching of the terms used to describe profiles and books may not be enough to evaluate similarity. A common strategy to deal with such a problem is to enrich the representation of the objects to be matched using additional sources of information. Our group has been exploiting this idea in different contexts, as described below.

4.4.1 *Content-Based Advertising.* An important topic related to web information retrieval and query processing is the problem of selecting ads in content-based advertisement systems. Content-based advertising constitutes a key web monetization strategy nowadays. In an initial research work related to advertisement selection, we present approaches in which external sources of information are used to improve ranking on content-based targeted advertising [Ribeiro-Neto et al. 2005]. We propose and evaluate several methods for matching pages and advertisements, and determine how accurate they are in picking the most relevant advertisements to the content of a web page.

In a more recent work, we propose a new framework for associating ads with web pages based on genetic programming [Lacerda et al. 2006]. Our method aims at learning functions that select the most appropriate ads, given the contents of a Web page. These ranking functions are designed to optimize overall precision and minimize the number of misplacements.

4.4.2 *Category-Based Recommendation.* In [Matos-Junior 2011], we investigate how to take advantage of information encoded in a *taxonomy* to improve product recommendation. Using taxonomies opens the opportunity to incorporate domain-specific and common-sense knowledge compiled by humans, something that could not be obtained otherwise from neither the target nor the items alone. We propose three distinct strategies for using taxonomies in content-based recommender systems. Experimental results indicate that these strategies can yield gains close to 20% in average precision.

4.4.3 *Exploiting Entity Semantics.* In [Brandão et al. 2011], we present a new query expansion method that uses knowledge acquired from Wikipedia, exploiting the entity semantics available in its

[12] http://cmph.sf.net

content and structure. The main appeal in our method is that, differently from methods previously presented in the literature, it extensively uses information from attribute-value pairs available in infoboxes in a principled way. Such information is not only closely related to entities, but it is also continuously refined by human editors. Thus, they are valuable sources of semantic knowledge to obtain terms to be used in query expansion. In addition, our method uses infoboxes to associate entities identified in queries with categories. By doing so, we leverage previously proposed term-selection functions, adapting them to deal properly with entities, ultimately improving the accuracy of such functions in selecting the best terms to be used for query expansion.

An obstacle to this method is the lack of an infobox on the entity-related page from which attribute-value pairs could be obtained. To overcome it, in [Brandão et al. 2010] we propose a self-supervised approach for autonomously extracting attribute-value pairs from the text of Wikipedia articles. By doing this, we effectively create a *pseudo-infobox* that can be used by our query expansion method.

5. PERSPECTIVES AND FUTURE DIRECTIONS

Developing core technologies for managing and processing information on electronic documents has been the focus of our group. Several algorithms and techniques proposed represent the state-of-the-art in information retrieval solutions for web search engines and recommender systems. The key for the development of state-of-the-art technology has been the production of significant research results, which can be assessed by the quality of the publications produced by our group. Several challenging problems have been served as research topics for MSc and PhD students. Also important is the collaboration with other research groups, being worth stressing the connections of the group with start-up companies in the past and today.

Future work will include topics such as focused crawling, web data mining, efficiency issues in crawling, indexing and query processing, semantic enrichment in IR and recommender systems. The notion of autonomous search, "to tell me things I did not know but am probably interested in, is the next great stage of search", in the view of Google CEO Eric Schmidt[13]. The UFMG IR group is concentrating research efforts in this direction, e.g, [Menezes et al. 2010]. Recommender systems are unanimously pointed as the most effective tool to face the abundance of available information in the digital era. Contrary to traditional search systems, recommender systems have as their main task the autonomously delivery of relevant pieces of information (news, videos, books, etc) to users. This task is particularly challenging in environments like the Web and on-line social networks, where the information flows in an uncontrolled fashion, unpredictably, fast and in huge volumes.

REFERENCES

Agrawal, R., Imielinski, T., and Swami, A. Mining association rules between sets of items in large databases. In *ACM SIGMOD Int'l Conf. on Management of Data*. pp. 207–216, 1993.

Ahnizeret, K., Fernandes, D., Cavalcanti, J. M. B., de Moura, E. S., and da Silva, A. S. Information retrieval aware web site modelling and generation. In *Int'l Conf. on Conceptual Modeling*. pp. 402–419, 2004.

Alcântara, O., Jr., A. P., de Almeida, H., Gonçalves, M., Middleton, C., and Baeza-Yates, R. Wcl2r: A benchmark collection for learning to rank research with clickthrough data. *Journal of Information and Data Management* 1 (3): 551–566, 2010.

Badue, C. S., Almeida, J. M., Almeida, V., Baeza-Yates, R., Ribeiro-Neto, B., Ziviani, A., and Ziviani, N. Capacity planning for vertical search engines. *CoRR* vol. abs/1006.5059, 2010.

Badue, C. S., Baeza-Yates, R., Ribeiro-Neto, B., Ziviani, A., and Ziviani, N. Modeling performance-driven workload characterization of web search systems. In *ACM Int'l Conf. on Information and Knowledge Management*. pp. 842–843, 2006.

Badue, C. S., Baeza-Yates, R., Ribeiro-Neto, B., Ziviani, A., and Ziviani, N. Analyzing imbalance among homogeneous index servers in a web search system. *Information Processing and Management* 43 (3): 592–608, 2007.

Badue, C. S., Baeza-Yates, R., Ribeiro-Neto, B., and Ziviani, N. Distributed query processing using partitioned inverted files. In *Int'l Symp. on String Processing and Information Retrieval*. pp. 10–20, 2001.

[13]http://www.readwriteweb.com/archives/google_ceo_next_great_stage_of_search_is_automatic.php

Badue, C. S., Barbosa, R., Golgher, P. B., Ribeiro-Neto, B., and Ziviani, N. Basic issues on the processing of web queries. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 577–578, 2005.

Baeza-Yates, R., Gonnet, G. H., and Ziviani, N. Expected behaviour analysis of avl trees. In *2nd Scandinavian Workshop on Algorithm Theory*. pp. 143–159, 1990.

Baeza-Yates, R., Pereira-Jr, A., and Ziviani, N. Genealogical trees on the web: a search engine user perspective. In *Int'l World Wide Web Conf.*. pp. 367–376, 2008.

Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval (First edition)*. Addison-Wesley, 1999.

Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval (Second edition)*. Pearson, 2011.

Baeza-Yates, R. A., Barbosa, E. F., and Ziviani, N. Hierarchies of indices for text searching. *Information Systems* 21 (6): 497–514, 1996.

Barbosa, E. F., Navarro, G., Baeza-Yates, R. A., Perleberg, C. H., and Ziviani, N. Optimized binary search and text retrieval. In *3rd European Symp. on Algorithms*. pp. 311–326, 1995.

Berlt, K., de Moura, E. S., da Costa Carvalho, A. L., Cristo, M., Ziviani, N., and Couto, T. Modeling the web as a hypergraph to compute page reputation. *Information Systems* 35 (5): 530–543, 2010.

Bharat, K., Broder, A. Z., Dean, J., and Henzinger, M. R. A comparison of techniques to find mirrored hosts on the www. *IEEE Data Engineering Bulletin* 23 (4): 21–26, 2000.

Bian, J., Liu, T., Qin, T., and Zha, H. Ranking with query-dependent loss for web search. In *ACM Int'l Conf. on Web Search and Data Mining*. pp. 141–150, 2010.

Botelho, F., Galinkin, D., Meira-Jr., W., and Ziviani, N. Distributed perfect hashing for very large key sets. In *Int'l Conf. on Scalable Information Systems*, 2008.

Botelho, F., Kohayakawa, Y., and Ziviani, N. A practical minimal perfect hashing method. In *Int'l Workshop on Efficient and Experimental Algorithms*. pp. 488–500, 2005.

Botelho, F., Pagh, R., and Ziviani, N. Simple and space-efficient minimal perfect hash functions. In *Workshop on Algorithms and Data Structures*. pp. 139–150, 2007.

Botelho, F. and Ziviani, N. External perfect hashing for very large key sets. In *ACM Int'l Conf. on Information and Knowledge Management*. pp. 653–662, 2007.

Botelho, F. C., Lacerda, A., Menezes, G. V., and Ziviani, N. Minimal perfect hashing: A competitive method for indexing internal memory. *Information Sciences* 181 (13): 2608–2625, 2011.

Brandão, W. C., de Moura, E. S., Silva, A. S., and Ziviani, N. A self-supervised approach for extraction of attribute-value pairs from wikipedia articles. In *Int'l Symp. on String Processing and Information Retrieval*. pp. 279–289, 2010.

Brandão, W. C., de Moura, E. S., Silva, A. S., and Ziviani, N. Exploiting entity semantics for query expansion. In *IADIS Int'l Conf. WWW/Internet*, 2011.

Brin, S. and Page, L. The anatomy of a large-scale hypertextual web search engine. In *Int'l World Wide Web Conf.*. pp. 107–117, 1998.

Calado, P., Cristo, M., Gonçalves, M. A., de Moura, E. S., Ribeiro-Neto, B., and Ziviani, N. Link-based similarity measures for the classification of web documents. *Journal of the American Society for Information Science and Technology* 57 (2): 208–221, 2006.

Calado, P., Cristo, M., Moura, E., Ziviani, N., Ribeiro-Neto, B., and Gonçalves, M. A. Combining link-based and content-based methods for web document classification. In *ACM Int'l Conf. on Information and Knowledge Management*. pp. 394–401, 2003.

Calado, P., Moura, E. S., Ribeiro-Neto, B., Reis, I., and Ziviani, N. Local versus global link information. *ACM Transactions on Information Systems* 21 (1): 1–22, 2003.

Calado, P., Ribeiro-Neto, B., Ziviani, N., de Moura, E. S., and Silva, I. Local versus global link information in the web. *ACM Transactions on Information Systems* 21 (1): 42–63, 2003.

Coelho, T., Calado, P., Souza, L., Ribeiro-Neto, B., and Muntz, R. Image retrieval using multiple evidence ranking. *IEEE Transactions on Knowledge and Data Engineering* 16 (4): 408–417, April, 2004.

Couto, T., Cristo, M., Gonçalves, M. A., Calado, P., Ziviani, N., Moura, E., and Ribeiro-Neto, B. A comparative study of citations and links in document classification. In *ACM/IEEE Joint Conf. on Digital Libraries*. pp. 75–84, 2006.

Couto, T., Ziviani, N., Calado, P., Cristo, M., Gonçalves, M. A., de Moura, E. S., and Brandão, W. C. Classifying documents with link-based bibliometric measures. *Information Retrieval* 13 (4): 315–345, 2010.

da Costa Carvalho, A. L., de Moura, E. S., da Silva, A. S., Berlt, K., and de Souza Bezerra, A. J. A cost-effective method for detecting web site replicas on search engine databases. *Data Knowledge and Engineering* 62 (3): 421–437, 2007.

de Almeida, H. M., Gonçalves, M. A., Cristo, M., and Calado, P. A combined component approach for finding collection-adapted ranking functions based on genetic programming. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 399–406, 2007.

de Moura, E. S., Navarro, G., Ziviani, N., and Baeza-Yates, R. Fast searching on compressed text allowing errors. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 298–306, 1998.

de Moura, E. S., Navarro, G., Ziviani, N., and Baeza-Yates, R. Fast and flexible word searching on compressed text. *ACM Transactions on Information Systems* 18 (2): 113–139, 2000.

Fan, W., Gordon, M., and Pathak, P. A generic ranking function discovery framework by genetic programming for information retrieval. *Information Processing and Management* 40 (4): 587–602, 2004.

Faria, F., Veloso, A., de Almeida, H., Valle, E., Torres, R., Gonçalves, M., and Meira Jr., W. Learning to rank for content-based image retrieval. In *Multimedia Information Retrieval*. pp. 285–294, 2010.

Fernandes, D., de Moura, E. S., Ribeiro-Neto, B., da Silva, A. S., and Gonçalves, M. A. Computing block importance for searching on web sites. In *ACM Int'l Conf. on Information and Knowledge Management*. pp. 165–174, 2007.

Fonseca, B., Golgher, P., Moura, E. S., Pôssas, B., and Ziviani, N. Discovering search engine related queries using association rules. *Journal of Web Engineering* 4 (2): 215–227, 2004.

Fonseca, B. M., Golgher, P. B., Pôssas, B., Ribeiro-Neto, B., and Ziviani, N. Concept-based interactive query expansion. In *ACM Int'l Conf. on Information and Knowledge Management*. pp. 696–703, 2005.

Geng, X., Liu, T., Qin, T., Arnold, A., Li, H., and Shum, H. Query dependent ranking using k-nearest neighbor. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 115–122, 2008.

Gonçalves, M. Text Classification. In R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval (Second edition)*. Pearson, pp. 281–335, 2011a.

Gonçalves, M. Digital Libraries. In R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval (Second edition)*. Pearson, pp. 711–735, 2011b.

Gonnet, G. H., Baeza-Yates, R., and Snider, T. New indices for text: Pat trees and pat arrays. In *Information Retrieval: Data Structures & Algorithms*. pp. 66–82, 1992.

Guidolini, R. *Detecção de Réplicas de Sítios Web em Máquinas de Busca Usando Aprendizado de Máquina*. M.S. thesis, Universidade Federal de Minas Gerais, Belo Horizonte, 2011. Advisor Nivio Ziviani.

Henrique, W. F. *Verificação de unicidade de URLs em Coletores de Páginas Web*. M.S. thesis, Universidade Federal de Minas Gerais, Belo Horizonte, 2011. Advisor Nivio Ziviani.

Henrique, W. F., Ziviani, N., Cristo, M., Carvalho, C., de Moura, E. S., and Silva, A. S. A new approach for verifying url uniqueness in web crawlers. In *Int'l Symp. on String Processing and Information Retrieval*, 2011.

Kitajima, J. P., Ribeiro-Neto, B., Resende, M. D., and Ziviani, N. Distributed parallel generation of indices for very large databases. In *IEEE Int'l Conf. on Algorithms and Architectures for Parallel Processing*. pp. 745–752, 1997.

Lacerda, A., Cristo, M., Gonçalves, M. A., Fan, W., Ziviani, N., and Ribeiro-Neto, B. Learning to advertise. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 549–556, 2006.

Laender, A. H. F., de Lucena, C. J. P., Maldonado, J. C., de Souza e Silva, E., and Ziviani, N. Assessing the research and education quality of the top brazilian computer science graduate programs. *SIGCSE Bulletin* 40 (2): 135–145, 2008.

Lan, Y., Liu, T., Qin, T., Ma, Z., and Li, H. Query-level stability and generalization in learning to rank. In *Int'l Conf. on Machine Learning*. pp. 512–519, 2008.

Liu, T. Learning to rank for information retrieval. In *ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 904, 2010.

Manber, U. and Myers, E. W. Suffix arrays: A new method for on-line string searches. *SIAM Journal on Computing* 22 (5): 935–948, 1993.

Matos-Junior, O. *Uso de Taxonomias na Recomendação de Produtos*. M.S. thesis, Universidade Federal de Minas Gerais, Belo Horizonte, 2011. Advisor Nivio Ziviani.

Menezes, G. V., Almeida, J. M., Belém, F., Gonçalves, M. A., Lacerda, A., de Moura, E. S., Pappa, G. L., Veloso, A., and Ziviani, N. Demand-driven tag recommendation. In *European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. pp. 402–417, 2010.

Mourão, F., Rocha, L., Araújo, R., Couto, T., Gonçalves, M., and Meira, Jr., W. Understanding temporal aspects in document classification. In *ACM Int'l Conf. on Web Search and Data Mining*. pp. 159–170, 2008.

Navarro, G., Baeza-Yates, R. A., Barbosa, E. F., Ziviani, N., and Cunto, W. Binary searching with nonuniform costs and its application to text retrieval. *Algorithmica* 27 (2): 145–169, 2000.

Navarro, G., de Moura, E. S., Neubert, M. S., Ziviani, N., and Baeza-Yates, R. Adding compression to block addressing inverted indexes. *Information Retrieval* 3 (1): 49–77, 2000.

Navarro, G., Kitajima, J. P., Ribeiro-Neto, B., and Ziviani, N. Distributed generation of suffix arrays. In *Annual Symp. on Combinatorial Pattern Matching*. pp. 102–115, 1997.

Navarro, G. and Ziviani, N. Documents: Languages and Properties. In R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval (Second edition)*. Pearson, pp. 203–254, 2011.

Palotti, J. R. M., Salles, T., Pappa, G. L., Arcanjo, F., Gonçalves, M. A., and Jr., W. M. Estimating the credibility of examples in automatic document classification. *Journal of Information and Data Management* 1 (3): 439–454, 2010.

Pereira-Jr, A., Baeza-Yates, R. A., Ziviani, N., and Bisbal, J. A model for fast web mining prototyping. In *ACM Int'l Conf. on Web Search and Data Mining*. pp. 114–123, 2009.

Pôssas, B., Ziviani, N., Meira, W., and Ribeiro-Neto, B. An efficient approach for correlation-based ranking. *ACM Transactions on Information Systems* 23 (4): 397–429, 2005.

Pôssas, B., Ziviani, N., Meira Jr., W., and Ribeiro-Neto, B. Set-based model: A new approach for information retrieval. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 230–237, 2002.

Pôssas, B., Ziviani, N., Ribeiro-Neto, B., and Meira Jr., W. Processing conjunctive and phrase queries with the set-based model. In *Int'l Symp. on String Processing and Information Retrieval*. pp. 171–183, 2004.

Ribeiro, V. F. *A Família Miner de Agentes para a World Wide Web*. M.S. thesis, Universidade Federal de Minas Gerais, Belo Horizonte, 1998. Advisor Nivio Ziviani.

Ribeiro-Neto, B., Cristo, M., Golgher, P. B., and de Moura, E. S. Impedance coupling in content-targeted advertising. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 496–503, 2005.

Ribeiro-Neto, B., de Moura, E. S., Neubert, M. S., and Ziviani, N. Efficient distributed algorithms to build inverted files. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 105–112, 1999.

Salles, T. *Classificação automática de documentos temporalmente robusta*. M.S. thesis, Universidade Federal de Minas Gerais, Belo Horizonte, 2011. Advisor Marcos Andre Goncalves.

Salles, T., da Rocha, L. C., Mourão, F., Pappa, G. L., Cunha, L., Gonçalves, M. A., and Jr., W. M. Automatic document classification temporally robust. *Journal of Information and Data Management* 1 (2): 199–212, 2010.

Salles, T., Rocha, L., Pappa, G. L., Mourão, F., Meira, Jr., W., and Gonçalves, M. Temporally-aware algorithms for document classification. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 307–314, 2010.

Saraiva, P. C., de Moura, E. S., Fonseca, R. C., Jr., W. M., Ribeiro-Neto, B., and Ziviani, N. Rank-preserving two-level caching for scalable search engines. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 51–58, 2001.

Silva, I., Ribeiro-Neto, B., Calado, P., de Moura, E. S., and Ziviani, N. Link-based and content-based evidential information in a belief network model. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 96–103, 2000.

Silva, R., Veloso, A., and Gonçalves, M. Rule-based active sampling for learning to rank. In *European Conf. on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2011.

Szwarcfiter, J. L., Navarro, G., Baeza-Yates, R. A., Oliveira, J., Cunto, W., and Ziviani, N. Optimal binary search trees with costs depending on the access paths. *Theoretical Computer Science* 290 (3): 1799–1814, 2003.

Trotman, A. Learning to rank. *Information Retrieval* 8 (3): 359–381, 2005.

Vale, R. F., Ribeiro-Neto, B., Lima, L. R. S., Laender, A. H. F., and Freitas Jr., H. R. Improving text retrieval in medical collections through automatic categorization. In *Int'l Symp. on String Processing and Information Retrieval*. pp. 197–210, 2003.

Veloso, A., de Almeida, H. M., Gonçalves, M. A., and Meira, W. Learning to rank at query-time using association rules. In *Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval*. pp. 267–274, 2008.

Veloso, A., Gonçalves, M., Meira Jr., W., and de Almeida, H. Learning to rank using query-level rules. *Journal of Information and Data Management* 1 (3): 567–582, 2010.

Veloso, A. and Meira Jr., W. *Demand-Driven Classification*. Springer, 2011.

Ziviani, N. *Projeto de Algoritmos e Estruturas de Dados*. Editora Unicamp, 1986.

Ziviani, N. A system for efficient full-text retrieval. In *RIAO*. pp. 586–606, 1991.

Ziviani, N. *Projeto de Algoritmos com Implementações em Pascal e C (First edition)*. Pioneira Thomson, 1993.

Ziviani, N. Text Operations. In R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval (First edition)*. Addison-Wesley, pp. 163–190, 1999.

Ziviani, N. *Projeto de Algoritmos com Implementações em Pascal e C (Second edition)*. Thomson learning, 2004.

Ziviani, N. *Projeto de Algoritmos com Implementações em Java e C++*. Thomson Learning, 2007.

Ziviani, N. *Projeto de Algoritmos com Implementações em Pascal e C (Third edition)*. Cengage Learning, 2010.

Ziviani, N. and Albuquerque, L. C. Um novo método eficiente para recuperação em textos. In *Congress of the Brazilian Computer Society*. pp. 175–187, 1987.

Ziviani, N., de Moura, E. S., Navarro, G., and Baeza-Yates, R. Compression: A key for next-generation text retrieval systems. *IEEE Computer* 33 (11): 37–44, 2000.