Information Search in Web Archives

Miguel Ângelo Leal da Costa

Orientador:
Mário Jorge Costa Gaspar da Silva

Prova de Qualificação

Universidade de Lisboa
Faculdade de Ciências

Contents

1 Motivation

No one knows the real size of the world wide web. According to Google, in 2008 the web had more than a trillion of unique web pages[1]. Recently, Google's ex-CEO Eric Schmidt said that we create as much data in two days, around five exabytes, as we did from the dawn of man up until 2003[2]. The fast development of ICT (Information and Communication Technology) had a great impact on this growth. In the last decade, the world population with access to Internet grew more than 1,000% in some regions[3]. Computer-based devices and mobile phones with Internet connectivity are now about 5 billion[4], much of which equipped with technology that empowers people to easily create data. Moreover, tools such as social networks, blogs and CMSs (Content Management Systems) made it easier for people to publish and share data. This combination of factors resulted in the largest source of information ever created.

The web has a democratic nature, where everyone can publish all kinds of information. News, blogs, wikis, encyclopedias, interviews and public opinions are just a few examples of this enormous list. Part of this information is unique and historically valuable. However, since the web is too dynamic, a large amount of information is lost everyday. Ntoulas et al. discovered that 80% of the web pages are not available after one year [Ntoulas et al., 2004]. In a few years they are all likely to disappear, creating a knowledge gap for future generations. Most of what has been written today will not persist and, as stated by UNESCO[5], this constitutes an impoverishment of the heritage of all nations.

Several initiatives of national libraries, national archives and consortia of organizations started to archive parts of the web to cope with this problem[6]. Some country code top-level domains and thematic collections are being archived reg-

[1] see http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html
[2] see http://techcrunch.com/2010/08/04/schmidt-data/
[3] see http://www.internetworldstats.com/stats.htm
[4] see http://www.networkworld.com/news/2010/081610-5billion-devices-internet.html
[5] see portal.unesco.org/ci/en/files/13367/10700115911Charter_en.pdf/Charter_en.pdf
[6] see http://www.nla.gov.au/padi/topics/92.html

ularly[7]. Other collections related to important events, such as September 11th, are created at particular points in time[8]. In total, billions of web documents are already archived and their number is increasing as time passes. The historic interest over the documents is also growing as they age, becoming an unique source of past information for widely diverse areas, such as sociology, history, anthropology, politics, journalism or marketing. However, for making historical analysis possible, web archives must turn from mere document repositories into living archives. They need innovative solutions to search and explore past information.

Current web archives are built on top of web search engine technology. This seems like the logical solution, since the web is the main focus of both systems. They both collect and mine the web to create special indexes to search it. However, web archives reproduce the stored document versions as close as possible to the original at discrete points in time. Web search engines only redirect users to web servers hosting the documents. Web archives enable searching over multiple web snapshots of the past, while web search engines only enable searching over one snapshot of the close present. The web archives' mission is to preserve the web, while the only concern of web search engines is searching.

Given the above differences between both systems, I formulate the hypothesis that users from web search engines and web archives have different information needs that should be handled differently. The goal of this thesis is to understand these differences and take a step forward in developing IR (information retrieval) approaches for web archives, that better satisfy their users' information needs.

[7] see `http://www.archive.org/`
[8] see `http://www.loc.gov/minerva/`

2 Proposal Context

2.1 Projects

This work follows two projects, one concerned with searching the Portuguese web and another with its preservation. The Portuguese web is broadly considered the part of the web of interest to the Portuguese.

Tumba![9] was a web search engine optimized for the Portuguese web, which was available as a public service from 2002 to 2006 [Costa, 2004; Costa and Silva, 2010a]. Several experiments were conducted on the different data processing phases of this project, spanning from the crawling of documents to the presentation of results.

Tomba was a web archive prototype for the Portuguese web operated between 2006 and 2007 [Gomes et al., 2006]. The main difference from the Tumba! web search engine was that Tomba provided support for the storage and access to several versions of documents from consecutive snapshots of the web. These snapshots came from Tumba! and included only the textual part of the crawled documents. The prototype was publicly available with 57 million documents searchable by URL.

The Portuguese Web Archive (PWA) is Tomba's successor [Gomes et al., 2008]. It continues to archive the Portuguese web, which is now defined as the set of documents satisfying one of the following rules: (1) hosted on a site under a .PT domain; (2) hosted on a site under other domain, but embedded in a document under the .PT domain; (3) suggested by the users and manually validated by the PWA team. The PWA team has also integrated web collections from several other sources, such as the Internet Archive[10] and the Portuguese National Library. The indexed documents are now more than 180 million, ranging from 1996 to 2010, and searchable by full-text and URL. The documents can then be accessed and navigated as they were in the past. The experimental version

[9] see `http://www.tumba.pt`
[10] see `http://www.archive.org/`

Figure 1: Archived document from 1996.

of the PWA has been available as a service to the general public since 2010 at `http://arquivo.pt/`. It contains some of the first documents of the Portuguese web, such as the Faculty of Sciences' homepage shown in Figure 1.

The PWA serves other purposes beyond the preservation of historical and cultural aspects, such as the characterization of the Portuguese web [Miranda and Gomes, 2009] and the aggregation of special contents for research communities. Another important aspect is its contribution to the dissemination of the Portuguese language on the web, which is used by 247 million people and considered the fifth most popular language on the Internet[11]. My work takes place within the PWA,

[11] see `http://www.internetworldstats.com/stats7.htm`

Figure 2: Web archive workflow.

which is coordinated by the FCCN (National Foundation for Scientific Computing).

2.2 Workflow

The web data passes through several phases where it is transformed in a pipeline until it is presented to the user. Figure 2 illustrates the following workflow:

Acquisition: the web data can be acquired by several paths, such as from an entity that archived it previously or from the digitalization of publications in paper (e.g. The Times archive[12]). However, the most usual path is to crawl portions of the web. Crawling is the process of seeking and collecting data. It starts with the downloading of a set of URLs, that are then parsed to extract the URLs they link to. This process is continuously repeated for the extracted URLs that have not been downloaded yet, until a stop condition is met. The decision of what to archive is difficult, since there is not enough storage space to save everything and the web is permanently growing. Thus, some web archives prefer a more granular selection to exhaustively crawl a limited number of websites, such as the ones related to elections [Paynter et al., 2008]. Others prefer a wider selection of the web, but shallower, such as a top-level domain [Gomes et al., 2008].

Storage: the web data is persistently stored on secondary memory. Usually, web archives concatenate sequences of compressed web documents into long

[12] see `http://archive.timesonline.co.uk/tol/archive/`

files close to 100MB, where each document is proceeded by a small header. This format is called ARC and was originally developed by the Internet Archive [Burner and Kahle]. It offers an easier way to manage and speed up access to documents, since file systems have difficulty to handle billions of files. Recently, ARC was extended to the new WARC format that supports relations between contents [ISO 28500:2009]. The web documents and their sites can undergo several processes during or after storage. For instance, they can be enriched with descriptive metadata or their quality can be evaluated with a completeness measure.

Indexing: the web data is read from storage, uncompressed, broke up into words (tokenized) and syntactically analyzed (parsed). Parsing is necessary to distinguish text from metadata and analyze to which part of the document the text belongs. It is challenging because there are hundreds of file formats that must be handled and continue to evolve, such as HTML, PDF or new formats. Other processes can be applied, such as the morphological reduction of words to their root form (stemming) or the elimination of high frequent words called stopwords, that have no impact in discriminating the text (e.g. *the* and *of*). Then, index structures over the words and the metadata are created for efficient search.

Searching: the index structures are used to lookup the documents that match a received query. This match depends of the implemented retrieval model. Usually, for large scale collections such as the web, a model is chosen where all query terms must occur on the matching documents. The documents are then sorted by their relevance scores that measures how well they satisfy a user's information need. This need is formally represented by a query. Since millions of documents can match a query, ranking documents by relevance is essential to effectively find the desired information. This ranking is computed with a set of heuristics, based on several features such as the terms proximity or the number of links a document receives.

Presentation: the results are formated and displayed in ranked lists for end user consumption. Usually, each result is augmented with metadata, such as the title, URL and timestamp of when it was archived. Results can also be clustered by time for an easier perception of their temporal distribution or displayed along a timeline to support exploration tasks [Alonso et al., 2009]. When an archived document is shown, all of its hyperlinks are changed so that the references will point to the archive instead of the live web. This enables users to interactively browse the web as it was in the past.

Preservation: is a parallel process in this workflow, to guarantee that the web documents are accessible for long-term. Hence, data must be replicated within the data center and between data centers spread by different geographic locations, to prevent all sorts of failures. Data must also be stored in a tamper-proof manner to prevent someone from rewriting history. Malicious persons could try to take advantage of this fact for their own benefit. The monitoring of potential obsolescence in file formats and technology must be constant for a timely migration of the data before it is no longer accessible or usable.

This thesis focus mainly on the indexing and searching processes, despite all the other processes in the workflow have influence in the final outcome. These two processes usually encompass two systems working in tandem. One is the **searching system**, whose goal is to create indexes over the stored data and use them to speedup the matching of documents satisfying a query. The **ranking system** then uses the indexed data of the matching documents to estimate their query relevance. Documents matching the query are thus sorted in descending order by their relevance score, which enables users to find information effectively and efficiently.

2.3 Challenges

Much of the current effort on web archives development focuses on acquiring, storing, managing and preserving data [Masanès, 2006]. However, this is just the beginning. The data must be accessible for the public to see and exploit them.

My perspective is that web archives fail to support users' information needs. This is the result of manifold causes. First, there is not a clear understanding of who are the users and what they actually need. Second, web archives' IR is based on web search engines' IR. However, both are not designed to handle the temporal dimension created by the successive web snapshots as a first-class citizen. Third, there are no tools to agilize the collection and exploiting of information from the archived contents. Fourth, there is no evaluation of the current web archive IR technology. Each of these causes represent a research challenge that should be addressed.

Understanding what potential users need is the first step to the success of any IT (Information Technology) system. How can we offer what users want if we do not know what it is? Hence, I began by the study of users' information needs, i.e. the goals/intents behind their queries [Costa and Silva, 2010b]. An initial survey confirmed my hypothesis: users from web archives and web search engines have different needs. Web archive users mostly intend to:

- see how a web page or site, that they know, was in the past.
- see the evolution over time of a web page or site.
- collect information about a subject written in the past.

Time is present in all these needs, but not properly supported by the search technology available in today's web archives. The prevalent access in web archives is based on URL search, which returns a list of chronologically ordered versions of that URL. This type of search is limited, as it forces the users to remember the URLs, some of which may no longer exist for many years. The most desired web archive functionality is full-text search [Ras and van Bussel, 2007].

However, supporting full-text poses tremendous efficiency, effectiveness and scalability challenges. This is the reason why just a very few web archives support full-text and in a limited way. These web archives index at most two orders of magnitude less than the 150 billion documents served by the Internet Archive's Wayback Machine[13], which amounts to three petabytes of data or about 150 times the content of the Library of Congress. As web collections continue to grow, sooner or later web archives will have to face this data dimension, which is already an order of magnitude larger than the number of documents covered by the largest web search engines.

Moreover, the full-text search that these web archives support is based on the Lucene search engine[14]. Cohen et al. showed that the out-of-the-box Lucene produces low quality results, with a MAP (Mean Average Precision) of 0.154, which is less than half when compared with the best systems participating in the TREC Terabyte track [Cohen et al., 2007]. In addition, many of the specific characteristics of web archive collections are not handled by Lucene, degrading the quality of results even more. For instance, Lucene does not contemplate any temporal attribute, such as the crawl date or last-modified date of documents, despite my initial survey showing that users prefer the oldest documents over the newest [Costa and Silva, 2010b].

This general tendency of adapting web search engine IR technology to provide full-text search for web archives raises several questions that require a dedicated testbed to be studied. The elaboration of this testbed towards the evaluation of IR over web archive collections, is essential to demonstrate the superior effectiveness and robustness of some retrieval approaches and to produce sustainable knowledge for future development cycles. Appendix A presents and compares paradigms that can be adopted to create an IR test collection that meet the needs of web archives.

[13] see `http://www.archive.org/web/web.php`
[14] see `http://lucene.apache.org/java/docs/index.html`

3 Objectives and contributions

This PhD thesis intends to contribute toward the advance of IR in web archives. Its main objective is to pursue effective and efficient access mechanisms for users to unfold the full potential of web archives. In order to achieve this goal, my research should entail the use of:

- usage data collecting methods, such as surveys, search log mining and laboratory studies, to better understand the information needs, expectations and search patterns of its users. A clear understanding of the users is fundamental for the development of useful search functionalities that could imply new lines of research. The analysis of the users' search patterns and behaviors will support the architectural design decisions for a state-of-the-art web archive.

- IR and machine learning approaches to support time-travel queries, i.e. full-text search on the state of the web within a user-specified time interval. This can be considered a *killer application* for web archives, making historical analysis possible. The long time spans covered by web archives bring new challenges for IR. The models must adapt to the successive periods, where the tendencies of writing, design and publication of documents differ [Mota, 2009]. The models must also adapt to the popularity and authoritativeness of the documents throughout time.

- distributed and scalable IR architectures designed according to the temporal dimension, where for instance, indexes should be partitioned by time. Web search engines face many challenges related to scalability and information overload [Baeza-Yates et al., 2007]. Web archives face a greater challenge, because they accumulate previous documents and indexes, unlike web search engines that drop the old versions when new ones are discovered. Even so, web archives have a much smaller budget, which leads

them to find solutions that provide satisfactory results in *Google time* with much less resources.

These objectives will be assessed by:

- measuring the overall effectiveness and efficiency of the novel web archive IR system and comparing it with state-of-the-art alternatives. Effectiveness will be measured as how well the system retrieves the relevant documents along with the position of these documents within a ranked list. Common measures include precision and recall [Manning et al., 2008]. Efficiency will be measured as the elapsed time, throughput and the computational resources consumed to complete a task. A classic IR evaluation will be applied, using for this purpose a test collection that I will create, composed by a corpus, a set of topics (queries) and relevance assessments.
- measuring the usefulness of the developed technology deployed on the PWA, i.e. how appropriate is for the tasks and needs of the users. The monitoring of users interactions with the system, users self-reporting and laboratory studies will provide the quantitative and qualitative data to evaluate the users' satisfaction about the system.

The expected results of this work are:

- a deeper knowledge of web archive users about why, what and how do they search. These answers are essential to point out directions for developing technology that can better satisfy the users.
- a web archive IR system that provides effective and efficient mechanisms for real users to find and explore past information. A significant gain in performance is expected when compared with state-of-the-art. This system will be integrated in the PWA architecture, contributing directly to the preservation of the Portuguese web and all of its knowledge.

The relevant scientific contributions to research in web archiving, IR and web mining will be published in major conferences or journals of these research areas. Despite this research being focused in web archives, results can have interest to other domains, such as web search engines and digital libraries.

The modules of the web archive IR system will become publicly available under the LGPL license.

4 Work Plan

4.1 Completed Work

Web archives pose tremendous efficiency and effectiveness challenges, mostly because Google became the norm for users in terms of response speed and results quality. Users are not willing to wait more than a few seconds or browse through many results pages to find relevant information. Hence, I have put a great effort on the response speed, quality of results and scalability over the large and heterogeneous snapshots of the Portuguese web. This heterogeneity includes not only the different languages, formats and versions of documents along the years, but also the size and link sparseness of collections. The response speed was increased through several techniques, such as sorting indexes by a measure of relevance, which requires matching and ranking less documents. Several caches were also created in different system tiers and indexes were replicated to distribute load evenly among a group of servers. All these improvements were implemented in the search system.

The ranking system was completely recreated inside Lucene to enable selecting a ranking model at runtime, i.e. to select the ranking functions and the weight they have in a combination between them, after a query is submitted. This enables a great versatility on adjusting the ranking for different users' profiles, different types of searches and specially on testing and tuning ranking models. There are several difficulties in deciding which ranking model is best for web archive users

and this framework supports this decision based on L2R (learning to rank) algorithms [Liu, 2009].

L2R algorithms optimize search relevance by tuning the weights among large pools of ranking features (see Appendix A). L2R algorithms were applied to create and optimize the PWA ranking model. Due to the lack of explicit (manually) or implicit (e.g clicks on search results) relevance assessments over web archive collections, I conducted an experimental investigation using two TREC datasets summarized in Table 1. The TD2003 and TD2004 datasets are the topic distillation tasks from TREC 2003 and TREC 2004 web tracks. Both datasets include the .gov collection, composed by a crawl of .gov web sites in early 2002, containing 1,053,110 HTML documents. The datasets also include 50 and 75 queries, respectively. Triples <document, query, relevance judgment> are provided to test the IR system, together with evaluation metrics and evaluation tools.

I indexed these two datasets and for each query I extracted values from 30 ranking features for all top 1000 documents returned by BM25 [Robertson et al., 1995], with the respective binary judgments (relevant or nonrelevant). The features are for instance, the number of in-links, TFxIDF [Manning et al., 2008] and BM25 functions over different fields (URL, title, anchor and body). From the 30 features, 4 were selected with ranking feature selection algorithms to remove irrelevant and redundant features. These features were then weighted with L2R algorithms to tune the ranking model. This experiment resulted in a significant improvement of the quality of results for these two datasets. However, the existent evaluation frameworks such as TREC do not reflect the information needs of web archive users, nor consider the temporal dimension of collections. A dataset that considers the special characteristics of web archives must be created for a proper evaluation.

The PWA is being developed since 2008 [Gomes et al., 2008] and an experimental version of the system is publicly available since 2010 at `http://www.arquivo.pt`. As far as I know, the PWA supports the largest web archive collection searchable by full-text, with 180 million documents, and has

dataset	collection	q	f	rel	instances
TD2003	.gov	50	64	2	49058
TD2004	.gov	75	64	2	74146

q=number of queries; f=number of features; rel=levels of relevance

Table 1: Datasets used in the L2R experiment.

one of the largest time spans. The documents range between 1996 and 2010. Before the PWA was publicly available, people had great difficulty in suggesting any functionality or information need without seeing the system working. Showing similar systems from other countries, helped them to understand the concept of the project. Nevertheless, without real information needs over past documents and subjects they could remember and explore, the responses were too vague. Only now with the system working, I could conduct an exploratory study to analyze the users' information needs [Costa and Silva, 2010b]. This study answered why and what users search. Users perform mostly navigational searches, i.e. intend to reach a web page or site in mind. Nearly half of the informational needs, i.e. to collect information about a topic, are focused on names of people, places or things. Overall, web archives fail in supporting some needs, such as exploring the evolution of a web page throughout time.

Another study was conducted to identify how users search, which resulted in a search behavior characterization of web archive users [Costa and Silva, 2011]. Users do not spend much time and effort searching the past. They prefer short sessions, composed of short queries and few clicks. Full-text search is preferred to URL search, but both are frequently used. There is a strong evidence that users prefer the oldest documents over the newest, but mostly search without any temporal restriction. Overall, users search in web archives as they search in web search engines.

4.2 Developed software

I started by using some of the sub-projects of the Archive Access project[15], developed by members of the IIPC (International Internet Preservation Consortium). The IIPC has the goal of aggregating efforts to produce common tools and standards. NutchWAX is one of these tools, which is an extension of the Nutch search engine with Web Archive eXtensions. Nutch in turn is built on top of the Lucene search engine. NutchWAX runs in a Hadoop[16] cluster for distributed computing.

I quickly faced limitations in searching speed, quality of results and scalability of the IR system. To cope with this, I modified the software in the following manner:

- Nutchwax and Nutch's code were adapted to the web archive IR requirements. Several optimizations had to be added, from simplifications in the way document versions were searched and accessed to the inclusion of a search results page cache working in the web server. Several bottlenecks were resolved and the indexes replicated in multiple servers to scale-out the system. In fact, the whole system is replicated and accessed with a load balancing solution to enable a highly scalable and highly available service. All interactions of the users with the system are now registered and a search log mining module was created to analyze what and how users search. These data can be used to detect problems and envision more efficient searching solutions for the users.

  Figure 3 illustrates a typical session supported by the PWA, where the user submits a textual query and receives ten results ranked by relevance to the query. The user can then click on the results to see and navigate in the web pages as they were in the past. Each result has also an associated link to see all versions of the respective page. When clicked, the PWA presents the same results page as when a user submits that URL. Figure 4 depicts the interface, which is a table where each column contains all versions of a year

[15] see `http://archive-access.sourceforge.net/`
[16] see `http://hadoop.apache.org/`

Figure 3: Search interface after a full-text search.

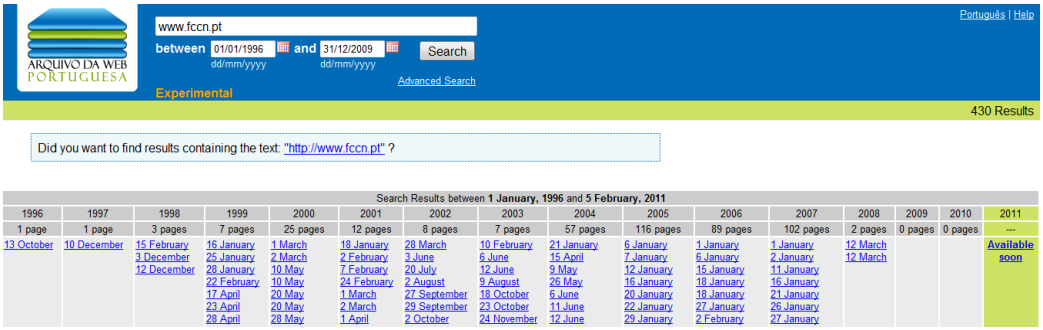

Figure 4: Search interface after an URL search.

sorted by date. The user can then click on any version to see it as it was on that date.

- Lucene's code for searching and ranking documents was completely rewritten. This gave me the freedom to redesign and tune search operators, such as the sort by date or relevance, which are now much more faster. Figure 5 shows the available operators supported by my version of Lucene. The addition of metadata caches, such as the documents' timestamps, or caches with index statistics constantly used at runtime, such as the title or document average length, also led to a great performance boost. The indexes were restructured by ordering the entries with an importance measure [Costa and Silva, 2004] and removing unnecessary data to reduce index

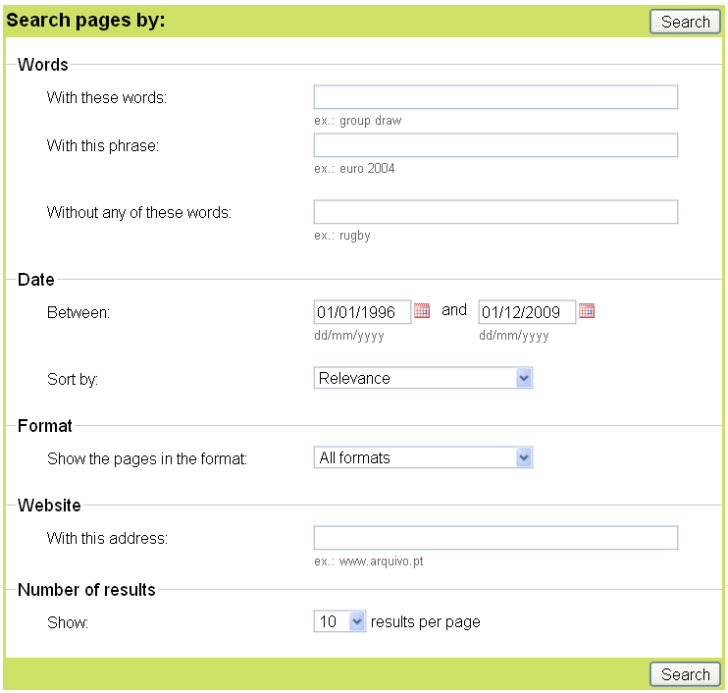

Figure 5: Advanced Search Interface.

size. This allowed the return of good quality results, without reading the full index entries (i.e. posting lists).

I also extended Lucene as a framework for evaluating ranking algorithms and models aimed to improve the quality of results. A set of ranking functions was first implemented, based on heuristics, such as the term frequency [Salton and Buckley, 1988; Robertson et al., 1995], term distance [Monz, 2004; Tao and Zhai, 2007], URL division [Kraaij et al., 2002] and web graph connectivity [Page et al., 1998; Abiteboul et al., 2003]. The Lucene's code was then changed to enable another module to select these functions and their weights at runtime, adjusting for instance the model to the users' query type. L2R algorithms will work over this framework to engineer rank-

ing models for different purposes (e.g. finding pages or collecting information).

These two modules are the fundamental building blocks upon which the PWA IR system was built.

4.3 Publications

These are the publications accepted during this thesis:

- Daniel Gomes, João Miranda and Miguel Costa, *A Survey on Web Archiving Initiatives*. Submitted to the International Conference on Theory and Practice of Digital Libraries 2011, Berlin, Germany.
- Miguel Costa and Mário J. Silva, *Characterizing Search Behavior in Web Archives*. In the 1st International Temporal Web Analytics Workshop, Hyderabad, India. March 2011.
- Miguel Costa and Mário J. Silva, *Understanding the Information Needs of Web Archive Users*. In the 10th International Web Archiving Workshop, Vienna, Austria. September 2010.
- Miguel Costa and Mário J. Silva, *A Search Log Analysis of a Portuguese Web Search Engine*. In INForum - Simpósio de Informática, Braga, Portugal. September 2010.
- Miguel Costa and Mário J. Silva, *Towards Information Retrieval Evaluation over Web Archives*. In the SIGIR 2009 Workshop on the Future of IR Evaluation, Boston, U.S. July 2009.
- Daniel Gomes, André Nogueira, João Miranda, Miguel Costa, *Introducing the Portuguese web archive initiative*. In the 8th International Web Archiving Workshop, Aaarhus, Denmark. September 2008.

4.4 Work ahead

The work plan foresees completion of the PhD work in four years and four months of which the first two years and eight months are complete, involving the creation and evaluation of the PWA IR system. The work plan to achieve the proposed objectives envisions:

- the creation of an IR test collection, considering the special characteristics of web archives. One or more of the following techniques will be chosen for this goal: pooling, implicit measures from logs and crowdsourcing (see Appendix A). The effectiveness of the ranking system will be evaluated with this collection. An attempt to join the research community towards the creation of an evaluation initiative over web archives was already made at a SIGIR workshop [Costa and Silva, 2009].

- the engineering of ranking models optimized for web archives, using the IR test collection as a fundamental piece in this process. The engineering is decomposed in a pipeline of four steps as explained in Appendix A. All steps will be researched and implemented as part of the ranking system to achieve the best effectiveness.

- the measurement of the usefulness of the PWA IR system. The system may be effective and efficient, but not appropriated for the users tasks and needs. The monitoring of the users interactions with the system, the users self-reporting and laboratory studies will provide the quantitative and qualitative data to evaluate the users' satisfaction about the system. All results will be analyzed to refine the PWA IR system and to validate the thesis.

- an architecture designed to continuously evolve toward a better fulfillment of web archive requirements. This architecture is described in Appendix A and must cope with the inevitably growth of data from web collections that will be integrated in the PWA. The architecture must be flexible to enable the deployment of different configurations to respond to the different needs

of web archives. The performance of the IR system designed and assembled according this architecture will be evaluated over a realistic set of queries extracted from the PWA' search logs.

- the continuous improvement of the PWA IR system to ensure satisfaction of the users' requirements. A regular effort must be employed to maintain the system robust to failures and fast enough for real users to use. Problems and bottlenecks need to be constantly monitored and fixed timely to guarantee the quality of service supported by the PWA, which is an essential condition to assure a significant number of users for further analysis.

4.5 Calendar

Figure 6 presents the calendar of the planned work. Three milestones are defined:

1. The experimental version of the PWA IR system is publicly available since 2010.

2. An IR test collection will be created and distributed to the web archiving research community at the end of June 2011. This is a step towards the normalization of experiments.

3. The improved version of the PWA IR system based on the findings of this thesis, will be made publicly available at the end of June 2012.

5 Acknowledgments

This work could not be done without the help and infrastructure of the Portuguese Web Archive team. I thank FCT (Portuguese research funding agency) for its Multiannual Funding Programme.

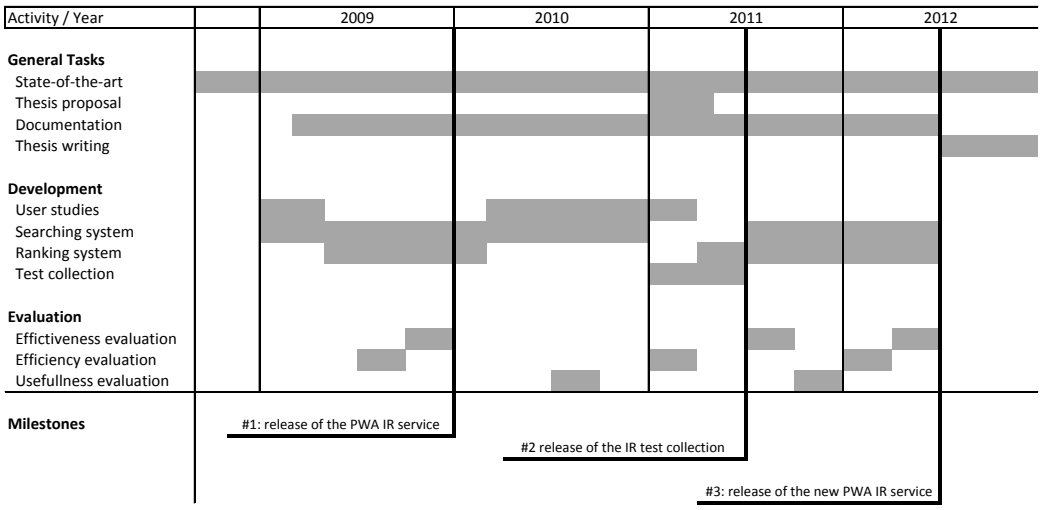

Figure 6: Calendar for the planned tasks of the PhD thesis.

References

S. Abiteboul, M. Preda, and G. Cobena. Adaptive on-line page importance computation. In *Proc. of the 12th International Conference on World Wide Web*, pages 280–290, 2003.

O. Alonso, M. Gertz, and R. Baeza-Yates. Clustering and exploring search results using timeline constructions. In *Proc. of the 18th ACM Conference on Information and Knowledge Management*, pages 97–106, 2009.

R. Baeza-Yates, C. Castillo, F. Junqueira, V. Plachouras, and F. Silvestri. Challenges in distributed information retrieval (invited paper). In *Proc. of the 23rd International Conference on Data Engineering*, 2007.

M. Burner and B. Kahle. The Archive File Format. URL `http://www.archive.org/web/researcher/ArcFileFormat.php`.

D. Cohen, E. Amitay, and D. Carmel. Lucene and Juru at Trec 2007: 1-million queries track. In *Proc. of the 16th Text REtrieval Conference*, 2007.

M. Costa. Sidra: a flexible web search system. Master's thesis, University of Lisbon, Faculty of Sciences, November 2004.

M. Costa and M. J. Silva. Optimizing ranking calculation in web search engines: a case study. In *Proc. of the 19th Simpósio Brasileiro de Banco de Dados, SBBD'2004*, 2004.

M. Costa and M. J. Silva. Towards information retrieval evaluation over web archives. In *Proc. of the SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 37–40, 2009.

M. Costa and M. J. Silva. A search log analysis of a Portuguese web search engine. In *Proc. of the 2nd INForum - Simpósio de Informática*, pages 525–536, 2010a.

M. Costa and M. J. Silva. Understanding the information needs of web archive users. In *Proc. of the 10th International Web Archiving Workshop*, pages 9–16, 2010b.

M. Costa and M. J. Silva. Characterizing search behavior in web archives. In *Proc. of the 1st International Temporal Web Analytics Workshop*, 2011.

D. Gomes, S. Freitas, and M. J. Silva. Design and selection criteria for a national web archive. In *Proc. of the 10th European Conference on Research and Advanced Technology for Digital Libraries*, pages 196–207, 2006.

D. Gomes, A. Nogueira, J. Miranda, and M. Costa. Introducing the Portuguese web archive initiative. In *Proc. of the 8th International Web Archiving Workshop*, 2008.

ISO 28500:2009. Information and documentation - WARC file format. URL `http://www.iso.org/iso/catalogue_detail.htm?csnumber=44717`.

W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proc. of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34, 2002.

T. Liu. *Learning to Rank for Information Retrieval*, volume 3 of *Foundations and Trends in Information Retrieval*. Now Publishers Inc., 2009.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

J. Masanès. *Web Archiving*. Springer-Verlag New York Inc., 2006.

J. Miranda and D. Gomes. Trends in Web characteristics. In *Proc. of the 7th Latin American Web Congress*, pages 146–153, 2009.

C. Monz. Minimal Span Weighting Retrieval for Question Answering. In *Proc. of the SIGIR 2004 Workshop on Information Retrieval for Question Answering*, pages 23–30, 2004.

C. Mota. *How to keep up with Language Dynamics: A case-study on Named Entity Recognition*. PhD thesis, Instituto Superior Técnico, May 2009.

A. Ntoulas, J. Cho, and C. Olston. What's new on the web?: the evolution of the web from a search engine perspective. In *Proc. of the 13th International Conference on World Wide Web*, pages 1–12, 2004.

L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

G. Paynter, S. Joe, V. Lala, and G. Lee. A Year of Selective Web Archiving with the Web Curator Tool at the National Library of New Zealand. *D-Lib Magazine*, 14(5):2, 2008.

M. Ras and S. van Bussel. Web archiving user survey. Technical report, National Library of the Netherlands (Koninklijke Bibliotheek), 2007.

S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proc. of the 3rd Text REtrieval Conference*, pages 109–126, 1995.

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

T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 295–302, 2007.

Appendix A: A survey on IR in Web Archives

Contents

Information retrieval (IR) is a broad interdisciplinary research field that draws on many other disciplines, such as computer science, mathematics, user interfaces, cognitive psychology, linguistics and library science. It studies the computational search of information within collections of data with little or no structure [Manning et al., 2008; Baeza-Yates and Ribeiro-Neto, 1999]. Often, IR deals with the matching of natural language text documents with users' queries, but it also studies other forms of content, such as the web and its search engines. These have become the dominant form of information access.

Web archiving is a research field concerned with the preservation of the web for future generations [Masanès, 2006]. The dynamic and ephemeral nature of the web means that web sites are continually evolving or disappearing. Web archiving mitigates this problem by studying strategies to select, acquire, store and manage portions of the web. These strategies must handle the rapid obsolescence of technologies for contents to remain accessible and usable for as long as they are needed. The effective use of these archived contents is also object of research, including IR and analytical tools to extract knowledge from them.

This Appendix presents an overview of information retrieval in web archives. Section 1, starts by addressing the index structures designed for search engines. These structures optimized for keyword matching, handle sub-optimally the queries restricted with a specific period of interest that web archives receive. Moreover, the indexes are not thought to be partitioned in a way that enable web archive architectures to scale and be easily manageable as new collections are added to the system. Given the very long life expectancy of web archives, it is expected that the data size will increase several orders of magnitude, surpassing largely the size indexed by top commercial web search engines. Section 2, presents the existing web archive architectures that face the enormous challenge of making all this data searchable.

A clear understanding of the users' information needs and how they search is fundamental to support the architectural design decisions. However, user studies are far-reaching. As shown in Section 3, they provide the insight to develop new

search technologies and the knowledge to improve existing ones. For instance, users studies enable to create a representative test collection for realistically simulating retrieval tasks. Which paradigm to employ in the creation of this test collection, is an important problem discussed in Section 4. This test collection is the missing cornerstone necessary to demonstrate the superior effectiveness and robustness of some retrieval approaches. Very little is known about how to provide the most relevant results in a ranked list for web archives. However, being time present in all the processes and foreseen solutions over a web archive, it should be considered as a first-class citizen when ranking results. Which temporal features to select and how to automatically combine them with non-temporal features, still needs to be researched as pointed out in Section 5.

Other important issues about web archiving are not addressed in this Appendix. The decision of what to archive is difficult, since there is not enough storage space to save everything and the web is permanently growing. Capturing web documents and sites as faithful as the original is also challenging, since it is necessary to interact with millions of web servers beyond control. I refer the interested reader to the following publications about collection selection [Gomes et al., 2006; Masanès, 2006] and web crawling [Castillo, 2005; Olston and Najork, 2010]. Also not addressed in this Appendix is the preservation of digital contents to guarantee their long-term access [Masanès, 2006; Strodl et al., 2007]. The topic includes the strategies based on the migration of contents to more standard formats or the emulation of the environment necessary to present and interact with the contents.

1 Index Structures

1.1 Inverted Files

The large size of web collections demands specialized techniques for efficient IR. The inverted index (a.k.a. inverted file) is the most efficient index structure for

textual search and can be easily compressed to reduce space [Zobel and Moffat, 2006]. It is composed of two elements: a lexicon and posting lists. The lexicon is the set of all terms that occur on the documents collection. Each term of the lexicon points to a posting list, which is a list of identifiers of documents where the term occurs. Each of these identifiers, id, has usually associated a payload, p, that may include the frequency of the term into the document and information about where the term occurs (e.g. in title). The pair $< id, p >$ is called a posting. For a query to match the potential relevant documents, each query term is searched on the lexicon and the posting list it points to is fetched. The result is the combination of the identifiers on the posting lists.

Web collections continue to grow as new snapshots are crawled and archived. A centralized index for all these data, if possible, would be inefficient. A distributed approach requires partitioning the index and spreading it by several computers to parallelize searching. Inverted files are mainly partitioned in two ways as illustrated in Figure 1. They can be partitioned by document or term [Zobel and Moffat, 2006]. Document-based partition (DP) refers to splitting the index per document (vertically). Actually, each computer creates a sub-index of a disjoint subset of documents. A query response is the merging of the results produced by all computers using their sub-indexes. Term-based partition (TP) refers to splitting the index per term (horizontally) and allocating each posting list to a computer. This requires that computers execute pairwise exchanges of posting lists after they create their sub-indexes. A query response is the joining of results produced by the computers that have at least one query term in their sub-indexes. The major differences between both partitions are that:

- in DP, all computers are devoted to the processing of each query, achieving a higher parallelism that reduces response time. In TP, at most one computer per query term responds to a query. The others are free to respond to other requests, achieving a higher concurrency. This results in fewer disk seeks, which is a dominant factor in the search time.

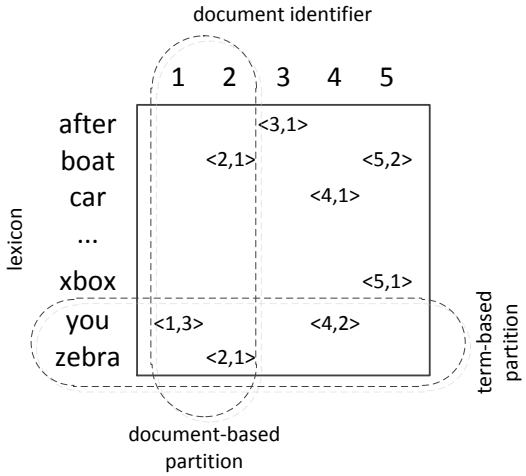

Figure 1: The two main ways to partition an inverted file.

- in DP, all information for matching a document is available locally, avoiding communication between computers. This results in a nearly linear scale out. In TP, the posting lists allocated in different computers must be joined at some point. Usually, a broker joins postings by repeatedly requesting batches of them until it has enough.

- DP does not require rebuilding of the indexes when new documents are indexed, while TP does by adding the new documents to the existent posting lists. Rebuilding the indexes each time a new web collection is integrated into a web archive can be prohibitive.

Usually, commercial web search engines, such as Google, use the document-based partition [Barroso et al., 2003]. Results show that this partition achieves superior query throughput [Zobel and Moffat, 2006].

1.2 Temporal Inverted Files

Web archives receive a significant number of queries with a specific time interval of interest, denoted time-travel queries [Costa and Silva, 2011]. Thus, partitioning

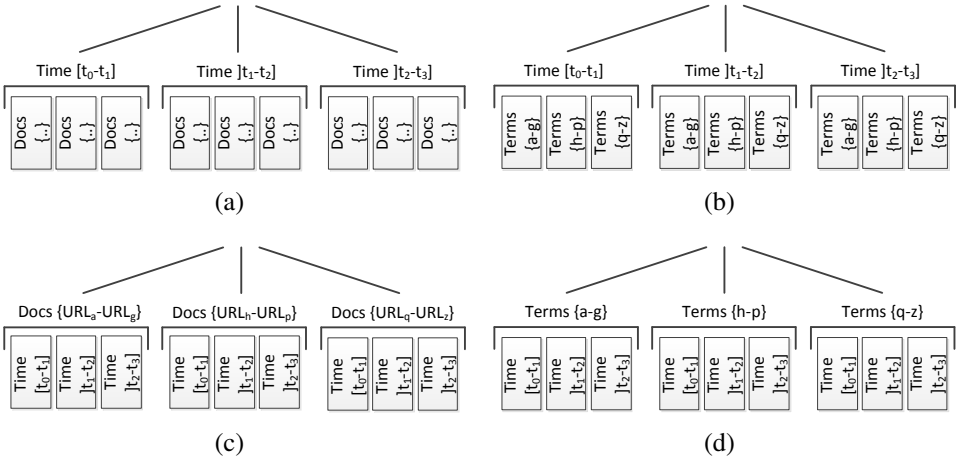

Figure 2: Index partitions using three dimensions: time, documents and terms.

the index by time enables to search only the sub-index corresponding to that interval. In this Section, we refer to the time when the web collections were crawled.

We can have at least four types of partitions, where the index is split per time and then per document or term, or the opposite. When partitioning first by time, subsets of documents are allocated to computer clusters according to their timestamps. Then, each subset can have a document-based or term-based partition within the cluster, as illustrated in Figures 2(a) and 2(b), respectively. This offers a simple way to scale web archive indexes, since new clusters are added to the system as new web snapshots are indexed. Another advantage is that time-travel queries will be sent only to the clusters handling that time interval.

When partitioning first by document and then by time, subsets of documents are allocated to computer clusters according to their identifiers (e.g. URLs), and then each subset within a cluster is partitioned according to the timestamps of the documents. Figure 2(c) depicts this partition. In the same way, when partitioning first by term and then by time, subsets of posting lists are allocated to computer clusters according to their terms, and then the postings of each subset are partitioned according to their documents' timestamps. See Figure 2(d). The advantage

of these last two partitions is that the sub-indexes of each cluster can be overlapped to reduce space. The sub-indexes have documents that span multiple temporal partitions and are replicated across posting lists. Berberich et al. extended postings with a validity time interval $[t_b, t_e]$, having the form $< id, t_b, t_e, p >$ [Berberich et al., 2007]. On the other hand, the overlapping increases the size of posting lists and consequently the response time of the IR system. It is a trade-off between space and speed that must be chosen.

1.3 Discussion

The document-based partition of the index offers superior query throughput and does not require rebuilding of the indexes each time new documents are indexed, contrary to the term-based partition. These are two important aspects of decision, specially the last one, because rebuilding the indexes involves a great computational effort and complexity. The decision of partitioning the index first or after per time, presents itself as a trade-off between speed and space, but partitioning first by time has the additional advantage of offering a simple way to manage and scale web archive indexes, since new clusters are added to the system as new web snapshots are indexed.

2 Web Archive Architectures

The specification of a web archive architecture comprises the definition of the necessary components to support the service, how these components interact and the requirements governing those interactions. From the user's perspective there are three main requirements: good quality search results, short response times, and a large coverage of the web throughout time. These requirements are key drivers for the architectural design decisions. Other requirements, mostly non-functional, are also important. They include high availability, i.e. how often the system is accessible to users, and high scalability, i.e. the ability to cope with growing amounts of load or data as we add more resources to the system. Given

the very long life expectancy of the service, it is expected that the data size will increase several orders of magnitude and that many technological changes will occur. Hence, the architecture must be created without the need to include any special hardware or proprietary software. Instead, it should be modular enough to easily exchange core modules as technology evolves. Next, I present the three known web archive architectures.

2.1 Wayback Machine

The open source Wayback Machine (WM) is a set of loosely coupled modules that can be exchanged and configured according to the web archive needs [Tofel, 2007]. This architecture supports the Internet Archive's WM that serves tens of millions of daily requests, over 500 terabytes of web documents archived since 1996 [Jaffe and Kirkpatrick, 2009]. This search is, however, limited by users having to know beforehand which URL to search. In the Internet Archive's WM, queries are routed to a broker via HTTP, which in turn routes it to several index servers. Each index server responds over a subset of documents, meaning that this architecture uses a document-based index partition. The broker then merges the results containing a list of all URL's versions chronologically ordered, and sends them back in XML to the WM. To access one of these versions, the WM uses as index a flat file sorted alphabetically by URL and divided in similar size buckets. The buckets are distributed across web servers and map URLs to the ARC files storing them [Burner and Kahle]. Thus, a URL request is routed to the web server handling that range of URLs to identify the respective ARC. Then, the ARC file is located by sending a UDP broadcast to all storage servers. There are more than 2500 storage servers in the primary data center, each with up to four commodity disks. The storage server containing the ARC file replies to the broadcast with the local file path and its identification. Finally, the client browser is redirected to the respective storage server, which runs a lightweight web server to deliver the ARC file.

Since all storage server replicas respond to each broadcast request, a computer failure does not interrupt the service. However, by having all replicas responding to the same request, a lot of work is replicated unnecessarily. The main advantage of this architecture is that it enables the system to be easily extended or replicated, just by adding more computers to the cluster.

2.2 Portuguese Web Archive

The Portuguese Web Archive (PWA) is based on the WM, but uses NutchWAX as its full-text and URL search engine [Gomes et al., 2008]. NutchWAX was developed by the International Internet Preservation Consortium (IIPC). It is an extension of the Nutch search engine with Web Archive eXtensions. There are some older descriptions of Nutch experiments on the Internet Archive [Stack, 2005]. Nutch is a search engine built on top of Lucene [Hatcher and Gospodnetic, 2004] that follows a document-based partition of the index. We extended it to partition first by time and only then by documents. As result, the PWA has a scheme where a broker only routes queries to index servers serving collections within the interval of interest. Queries are also balanced between several replicas of the system with the Linux Virtual Server (see `http://www.linuxvirtualserver.org/`). Hadoop is an important piece in scaling this architecture [White, 2010]. Hadoop is a framework that provides distribution, parallelization, load-balancing and fault-tolerance services to software programmed according to the MapReduce model [Dean and Ghemawat, 2008]. It is especially well-suited to process large datasets. The PWA currently supports tens of full-text queries per second over a web collection of more than 180 million documents created since 1996.

2.3 Everlast

Everlast is a web archive with a peer-to-peer (P2P) architecture for storing and searching past documents with indexes partitioned by term and then by time [Anand et al., 2009]. P2P architectures have loosely-coupled computers called

	Wayback Machine	PWA	Everlast
Scalability	High	High	Very High
Reliability	High	High	Medium
Time-aware	No	Yes	Yes
Performance	High	Very High	Medium

Table 1: Comparison between web archive architectures' characteristics.

peers, where each peer operates as both a server and a client aiming to share resources. The power of P2P lies in their capability to provide services with practically unlimited scalability of resources. Some existing P2P systems are formed by millions of peers connected via the Internet. These peers belong to people participating in the effort to provide a common service, which diminishes drastically the cost of the system. However, the peers could be unreliable and transient due to their autonomy, which result in data loss. This tends to be mitigated by replicating data in several peers. Another problem is that P2P systems are based on decentralized object location and routing schemes, such as Chord [Stoica et al., 2001] or Pastry [Rowstron and Druschel, 2001]. The expected number of routing steps to find an object (e.g. document) is $O(log\ n)$, where n is the number of peers in the network. This number of steps and the communication latency via the Internet can degrade the performance to a time longer than the users are willing to wait.

2.4 Comparison

The architectures of the Wayback Machine and the PWA are very similar. The main difference is the type of index partition, where the PWA takes the time of collections in consideration and uses it to improve performance (i.e. response time and query throughput).

The Everlast enables a higher scalability of storage space and load, but its decentralized coordination over the Internet degrades performance and diminishes reliability. These two factors are quite prohibitive for users that search in web

archives as they search in web search engines and expect the same behavior [Costa and Silva, 2011]. Table 1 contrasts the architectures' characteristics.

3 Web Archive User Studies

3.1 Scenarios and Usability

There are several web archiving initiatives currently harvesting and preserving the web heritage, but very few studies about the behavior of web archive users. The International Internet Preservation Consortium (IIPC) reported a number of possible user scenarios over a web archive [Group, 2006]. The scenarios are related to professional scopes and have associated the technical requirements necessary to fulfill them. These requirements include a wide variety of search and data mining applications that have not been developed yet, but could play an important role. However, the hypothetical scenarios did not come directly from web archive users.

The National Library of the Netherlands conducted an usability test on the searching functionalities of its web archive [Ras and van Bussel, 2007]. Fifteen users participated on the test. One of the results was a compiled list of the top ten functionalities that users would like to see implemented. Full-text search was the first one, followed by URL search. Strangely, time was not considered in none of the top ten functionalities, despite being present in all the processes on a web archive. The users' choices can be explained by web archives being mostly based on web search engine technology. As a result, web archives offer the same search functionalities. This inevitably constrains the users' behaviors. Another explanation is that Google became the norm, influencing the way users search in other settings.

3.2 Characterizations

Costa and Silva studied the information needs of web archive users [Costa and Silva, 2010]. They used three instruments to collect quantitative and qualitative data, namely search logs, an online questionnaire and a laboratory study. The data collected by the three instruments led to coincident results. Users perform mostly navigational searches without a temporal restriction. Other findings show that users prefer full-text over URL search, the oldest documents over the newest and many information needs are expressed as names of people, places or things. Results also show that users from web archives and web search engines have different information needs, which cannot be effectively supported by the same technology.

In a subsequent study, Costa and Silva characterized the search behavior of web archive users [Costa and Silva, 2011]. Their results showed that users did not spend much time and effort searching the past. Users prefer short sessions, composed of short queries and few clicks. This study confirmed that users prefer full-text over URL search, but both are significantly used. It also presented stronger evidences that users prefer the oldest documents over the newest. The main conclusion was that users of web archives and web search engines have similar behaviors. In general, web search technology can be adopted to work on web archives. However, this behavior seems to be the consequence of the Portuguese Web Archive having offered a similar interface, leading users to search in a similar way. They suggested that new types of interfaces must be experimented, such as the temporal distribution of documents matching a query or timelines, which could create a richer perception of time for the user and eventually trigger different search behaviors.

4 IR Assessment Paradigms

Modern IR evaluations are based on the Cranfield paradigm established in the 1960s by Cleverton [Cleverdon, 1967]. This paradigm defines the creation of test collections for evaluating and comparing retrieval results. The test collections are

composed by three parts: (1) a corpus representative of the documents that will be encountered in a real searching environment; (2) a set of topics (also referred as queries) simulating the users' information needs; and (3) relevance judgments (or assessments) indicating which documents are relevant and nonrelevant for each topic. The performance of an IR system is then measured by comparing the result lists against the known correct answers for each topic.

A representative corpus of a web archive collection could be compiled straightforward from the available collections. Some particularities of these collections are pointed out in [Costa and Silva, 2009]. The previous section addresses the users's information needs and how they are represented as queries. Next, I present the three most used assessment paradigms, that can be used to judge the relevance of web archive documents. Note that due to the size of web collections, assessing all documents is impractical. These paradigms try to diminish the human effort, while increasing the assessment coverage.

4.1 Pooling

The pooling paradigm, first suggested by Sparck Jones and Van Rijsbergen in 1975, is usually adopted to build ad-hoc collections [Jones and Van Rijsbergen, 1975]. Pooling is based on the assumption that the top-ranked documents of many and diversified IR systems aggregate most of the relevant documents. For that, each participant (group or individual) submits several runs, where each run corresponds to a list of the top-ranked documents for each topic (usually 1000). Participants also indicate the order of preference for runs to be added to the pool. The pool aggregates the top-ranked documents (usually 100 or 50) for each of the selected runs, which are then judged with relevancy degrees (usually binary or ternary) by several assessors. All unpooled documents are considered nonrelevant. The pool is considered the ground truth and is used to evaluate all the submitted runs. Looking to previous research, a total of 50 topics and a pool depth (the number of documents evaluated per topic) of 100 showed to be sufficient for reliable comparisons of retrieval systems [Zobel, 1998; Buckley and Voorhees, 2000;

Sanderson, 2005]. The diversity of the runs is also important to find new techniques that present good results. Thus, the use of diverse techniques to increase the diversity of the pool is encouraged, such as pseudo-relevance feedback and other query expansion techniques. Manual runs tend to add relevant documents that are not found by automatic systems using current technology. The related literature points to adding at least 50 varied runs to create the pool [Zobel, 1998].

By guaranteeing the minimal requirements, the pooling methodology showed to fairly compare the IR systems, i.e. the ranking of performances of the evaluated systems is maintained even if their performance scores vary after changing the pool [Zobel, 1998; Voorhees, 2000]. There are some modifications to pooling that reduce the human effort in assessing test collections, but even so, a significant effort by the research community is required [Aslam et al., 2006].

4.2 Implicit Feedback

Logs of search engines can be analyzed to model user interactions [Jansen and Spink, 2006; Markey, 2007] and to improve their ranking quality [Joachims, 2002; Radlinski and Joachims, 2005]. Many studies follow this approach, because it makes it possible to record and extract a large amount of implicit feedback at low cost. Top commercial web search engines receive hundreds of millions of queries per day. Logs also have the advantage of being a non-intrusive mean of collecting user data about the searching process. Most users are not aware that their interactions are being logged, which leads them to behave as if they were not under observation. Another use of this feedback is that it can be used to produce relevance judgments over the specific collections being served, in contrast to more general collections made available for testing.

On the other hand, search logs are limited to what can be registered. In public search engines, there is often no contextual information about the users, such as their demographic characteristics, the motivations that lead them to start searching, and their degree of satisfaction with the system. The major disadvantage of collecting implicit feedback is that the gathered data is noisy, thus being hard to

interpret. For instance, Fox et al. discovered that the viewing time of a document is an indicator of relevance [Fox et al., 2005]. However, the amount of time the document is open after selected, does not necessary correspond to the reading time by the user.

Clicks on the result lists provide important feedback about the users choices. However, this data is also problematic because it contains many false positives (clicks on nonrelevant documents due to misleading snippets) or false negatives (relevant documents that are not clicked because they are placed too low in the ranking or have poor snippets). The noise can be mitigated by considering a large number of replicated feedback. According to Joachims et al., the clicks are reasonably accurate if they are used as relative judgments between documents on ranked lists of results [Joachims et al., 2005]. For instance, if the second result is clicked and the first is not, then we can conclude that the second tends to be more relevant. Users are unlikely to click on a result they consider less relevant than another they first observed on a higher rank.

4.3 Crowdsourcing

Crowdsourcing emerged as an alternative to conduct relevance evaluations [Alonso et al., 2008] and user studies [Kittur et al., 2008] by taking advantage of the power of millions of people connected through the Internet. The idea is to post tasks on the web in the form of an open call, which are outsourced by a large group of online users in exchange of a small payment. These are easy tasks for people, but hard for computers. For instance, assessing the relevance of documents.

There are several online labour markets for crowdsourcing. Amazon Mechanical Turk (`https://www.mturk.com`) has been adopted in many of the crowdsourcing relevance evaluations [Alonso et al., 2008; Kittur et al., 2008; Alonso and Mizzaro, 2009]. It accepts just about anyone possessing basic literacy. Its use requires splitting large tasks into smaller parts for people willing to complete small amounts of work for a minimal amount of money.

There are other applications exploiting the power of crowdsourcing by presenting the tasks as a game to motivate participants [von Ahn and Dabbish, 2008]. A successful example is the Google's Image Labeler, where players label images for free while they play [von Ahn and Dabbish, 2004]. Besides entertainment, user participation can be pursued by promoting their social status, for instance using leader boards. Another approach is to engage persons from the same group who will benefit from the outcome of the data, such as in information retrieval communities. Hybrid approaches may consider aspects of entertainment and social status, but also monetary rewards to winners.

This paradigm substitutes expert judges by non-experts, which creates doubts about the assessments' reliability. Bailey et al. study concluded that there is a low level of agreement between both groups. As consequence, this produces small variations in performance that can affect the relative order between the assessed systems [Bailey et al., 2008]. Snow et al. showed the opposite [Snow et al., 2008]. A few non-experts can produce just as good or even better judgments as one expert. Alonso and Mizzaro used the TREC data to demonstrate in a small scale (for 29 documents of a topic) that Mechanical Turk users were accurate in assessing relevance and in some cases more precise than the original experts [Alonso and Mizzaro, 2009].

4.4 Comparison

Each paradigm has advantages and disadvantages. The pattern drawn from Table 2 shows that the more *human participants* (researchers and users) cooperate, the higher the *number* of assessments and the faster the *speed* to complete them. Pooling aggregates a large quantity of topic-document pair assessments. For instance, the average number of documents assessed per topic on the first eight years of the TREC's ad-hoc tracks was 1,464 [Voorhees and Harman, 1999]. Multiplying by the 50 topics, 73,200 judgments were necessary per year. This number will hardly scale up, since the number of involved researchers tends to be small in the order of the tens. On the other hand, implicit feedback can exploit the knowledge

	Pooling	Implicit feedback	Crowdsourcing
Human participants	Low	High	Medium
Assessment number	Medium	High	Medium
Assessment speed	Low	High	Medium
Assessment scale up	Low	High	Medium
Monetary Costs	High	Low	Medium
Assessment quality	High	Medium	Medium
Assessment coverage	High	Low	Medium
Topic coverage	Medium	High	Medium
Identification of needs	High	Low	High

Table 2: Comparison between assessment paradigms' characteristics, where High or Low is a good or bad indicator, respectively.

from millions of users and with that surpass the other paradigms in number of assessments, speed and with less effort. Crowdsourcing can engage hundreds of participants and with that reach the same number of assessments as pooling, but faster and with less effort per participant. Specifically, the crowdsourcing experiments tend to have a smaller number of assessments (thousands) completed in a couple of days. The *monetary costs* associated to assess a corpus are higher in pooling, because experts need to be contracted and researchers have to spend many days of work. In crowdsourcing a small amount in the order of hundreds of dollars is paid, while in implicit feedback the assessments are free.

Pooling loses in speed, but gains in *quality*, because there is a high *assessment coverage* executed by a set of experts following guidelines to reduce bias. Experts assess much more documents, typically 100 for each run, while the participants in other paradigms assess only a few, mostly from the first page of results. Pooling increases the diversity of results by assessing ranked lists from several IR systems that use different techniques. The other paradigms typically assess only one ranking list that the users have access to, which tend to be insufficient to detect most of the relevant documents. However, for crowdsourcing is easy to include this diversity by assessing the top documents from different ranking models.

All paradigms achieve sufficient *topic coverage*, but implicit feedback can reach a higher coverage by analyzing all the queries submitted to the IR system. Lastly, implicit feedback cannot perform the *identification of information needs* from queries, while in pooling and crowdsourcing the topics are created from information needs [Broder, 2002].

It is very hard to aggregate enough efforts of the research community to create a pooling based IR evaluation for web archives. Currently, it seems that the IR community is not highly aware and sensitized to the problems of the web archiving community and the web archiving community has given priority to other issues beyond IR, such as preservation. Using a standard relevance judgment benchmark such as TREC is not useful, because the collection is not time-related and the judgments do not contemplate the temporal information needs. Implicit feedback presents itself as a good choice, but most web archives do not have a large and varied set of users from which relevance judgments can be derived. If this is not the case, the best solution is crowdsorcing.

5 Ranking

Test collections have been used for decades to evaluate the many ranking models proposed by IR researchers. The ranking models estimate document relevance with regards to a query [Baeza-Yates and Ribeiro-Neto, 1999]. Documents matching the query can thus be sorted in descending order by their relevance score as a mean for users to find information effectively and efficiently. Hence, the creation of these models is a central problem in IR and for the systems that depend on them, such as web archives. Next, I present some of the ideas about the existent models and how to create them.

5.1 Conventional Ranking Models

Ranking models can be classified as query-dependent or query-independent. In query-dependent models, documents are ranked according to the occurrences of

query terms within them. Early models, such as the Boolean model, consider a document relevant if it matches the query terms and irrelevant otherwise [Manning et al., 2008]. On the other hand, the Vector Space model (VSM) computes degrees of relevance [Manning et al., 2008]. Both documents and queries' terms are represented as vectors in an Euclidean space, where the dimensionality of the vectors is the number of distinct terms in the collection. The inner product between the vectors measures the query and document similarity. The TF-IDF is one of the well-known functions that compute term weights for a document vector [Salton and Buckley, 1988]. The weight increases proportionally to the number of times a term occurs in the document and decreases with the frequency of the term in the collection. There are other weighting functions that provide good results, such as the probabilistic ranking model BM25, which additionally decreases the weight as a document length gets larger [Robertson et al., 1995]. Its variants, such as BM25F, take the document structure into account [Zaragoza et al., 2004]. For instance, BM25F scores a term differently if it occurs on the title, URL or anchor texts of other documents linking to the document.

All the above models assume that terms are independent. For example, a document would have the same relevance for the query *European Union*, whether the query terms occurred together or far apart. Some models overcome this by considering the terms proximity [Tao and Zhai, 2007]. Language models estimate the probability of a document generating the terms in the query [Song and Croft, 1999]. They handle the dependency between query terms by taking into consideration the fact that the probability of a term depends on the probability of previous adjacent terms.

Another type of data that can be explored is social annotations from sites such as *del.icio.us*, since annotations provide a good summary of the key aspects of the document [Bao et al., 2007].

In query-independent models, the documents are ranked according to an importance, quality or popularity measure computed independently of the query. The source most used to compute importance values is the hyperlink structure of the

web. One of the most well-known algorithms that uses it is PageRank, because it is partially responsible for the Google's initial success [Page et al., 1998]. PageRank relies on the assumption that the importance of a document depends on the number and the importance of the documents linking to it. There are other algorithms taking use of the web link structure, such as HITS [Kleinberg, 1999] or HostRank [Xue et al., 2005], but there are also algorithms analyzing different sources. For instance, analyzing the number of times a document was visited, the number of terms of a document or the degree of conforming to W3C standards [Richardson et al., 2006].

5.2 Temporal Ranking Models

Several works extended the language modeling approach to integrate temporal features. One of the most common ideas is to bias the document's prior probability of being relevant to favor the most recent documents [Li and Croft, 2003]. Boosting the most recent documents is desirable for queries where the user intends to find recent events or breaking news, such as in news search engines. Another idea is to favor more dynamic documents, since there is a strong relationship between the amount and frequency of content change in documents over time, and their relevance [Elsas and Dumais, 2010].

The distribution of the documents' dates reveals time intervals that are likely to be of interest to the query. For instance, when searching for *tsunami*, the peaks in the distribution may indicate when these events occurred. Thus, some studies explored the distribution of the publication dates of the top-k query matches to boost documents published withing relevant intervals [Jones and Diaz, 2007; Dakka et al., 2010]. However, identifying the dates of web documents is not straightforward. The metadata from the document's HTTP header fields, such as Date, Last-Modified and Expires are not always available, nor reliable. For instance, servers often send an invalid Last-Modified date of when the content was changed [Clausen, 2004]. Studies estimate that 35% to 80% of web documents have valid last-modified dates [Amitay et al., 2004; Gomes and Silva,

2006]. However, these percentages can be significantly improved by using the dates of the web document's neighbors, specially of the web resources embedded in the selected document (e.g. images, CSS, javascript) [Nunes et al., 2007].

The content is a valuable source of temporal information, but is likely to be the most difficult to handle. Temporal expressions can be extracted from text with the help of NLP and information extraction technology [Alonso et al., 2007]. Their inherent semantic is then mapped into the corresponding time intervals, which are used to measure the temporal distance to the search period of interest. Thus, instead of treating temporal expressions as common terms, they can be integrated in the language model to estimate the probability of a document generating the temporal part of the query [Irem Arikan and Berberich, 2009; Berberich et al., 2010]. Note however, that these expressions may refer to a time completely different from the publication date of the document. For instance, they can refer to an event in the past or future.

Query logs are another source that can be explored, for instance, to detect temporal implicit intends in queries [Metzler et al., 2009]. If a query is likely to contain years then, it may have a temporal intent. In this case, the documents with the years embedded in their content should be boosted.

Time awareness can also improve link-based ranking algorithms. A known problem in these algorithms is that they underrate recent documents, because the indegree used to compute the popularity of web documents, such as in PageRank [Page et al., 1998], favors older documents that have already accumulated a significant number of references over time. This problem can be overcome by weighing higher the in-links of the sources updated more recently [Yu et al., 2004; Amitay et al., 2004]. The idea is to reflect the freshness of source documents on the importance of the document they link to. Additionally, the update rates of the sources can also be considered [Berberich et al., 2005] or the obsolete links that point to documents that are no longer accessible [Bar-Yossef et al., 2004]. This gives a clear indication that the documents have not been maintained and contain out-of-date information.

5.3 Learning to Rank Models

All these conventional and temporal ranking models are just a few examples of the large number proposed over the years. They explore different features to determine whether a page is relevant for a query. The question now is, which ones are better suited for web archives?

Previous IR evaluations showed that combinations of ranking models tend to provide better results than any single model [Brin and Page, 1998; Craswell et al., 2005; Liu et al., 2007]. An individual model is also more susceptible to influences caused by the lack or excess of data (e.g. spam). Therefore, it is advantageous to use different aspects of data to build a more precise and robust ranking model. By robust, I mean a model capable of coping well with variations in data. For instance, a document can receive a low relevance score due to a small query term frequency, but a high number of in-links can identify the document as important. All these factors must be properly balanced by the model.

I decompose the generation of a ranking model in a pipeline of four steps: (1) extraction of low-level ranking features, such as term frequency or document length, which are then (2) assembled in high-level ranking features (a.k.a. ranking functions), such as BM25 [Robertson et al., 1995]; (3) the most suitable features for a retrieval task are selected and (4) combined in a way to maximize the results' relevance. For simplicity, this combination can be linear, i.e. for a document d with a vector of low-level ranking features associated, $\vec{d}$, the values produced by the n selected ranking features are added after each feature f_i is weighted by a coefficient λ_i and adjusted with a value b_i:

$$\text{rankingModel}(\vec{d}) = \sum_{i=1}^{n} \lambda_i f_i(\vec{d}) + b_i$$

However, the best combination between features can be non-linear. There are several ways to combine them non-linearly. One solution is to map features from its original space into a high-dimensional space, $\vec{d} \mapsto \Phi(\vec{d})$. Then by the means of the *kernel trick* it is possible to apply linear methods to nonlinear data [Schölkopf and Smola, 2002]. Another solution is to combine features in a non-linearly way,

such as in the case of genetic programming through the use of crossover and mutation operations [Yeh et al., 2007].

The first two steps of the model generation are well studied and some ranking features, such as BM25, are good ranking models by themselves [Manning et al., 2008]. Combining them manually is not trivial. There are search engines using a large number of features, such as Google that uses more than 200 (see `http://www.google.com/corporate/tech.html`). Manual tuning can lead to overfitting, i.e. it fits training data closely, but fails to generalize to unseen test data. Hence, in the last few years the fourth step has been concentrating attention. Supervised learning algorithms have been employed to tune the weights between combined ranking features, resulting in significant improvements [Liu, 2009]. Additionally, this offers the capability of rapidly adding new features to the model as advances are presented by the research community. This research is called learning to rank (L2R).

There are some benchmark datasets for ranking, such as LETOR [Liu et al., 2007; Qin et al., 2010], which aggregate IR test collections, including corpora, query sets, relevance judgments, evaluation metrics and evaluation tools. In addition to that, it extracts different low-level and high-level feature values for each <document, query> pair, eliminating the usual parsing and indexing difficulties. The results of some state-of-the-art L2R algorithms are also provided for a direct comparison. LETOR provides the means to experiment ranking models as illustrated in Figure 3. A training set is composed by n queries, where each query q has associated a set of p feature vectors $\vec{d}$ extracted from documents and a set of relevance judgments y. The learning system uses the training set to learn a model that tries to maximize the accordance between its predictions and the relevance judgments. The model is then tested by using a test set similar to the training set, but without the judgments. The ranking system uses the model to compute a relevance value for each document according to the query. The results are then sorted by these values and evaluated using the judgments of the tested documents (ground truth). The average measure over the entire test set represents the over-

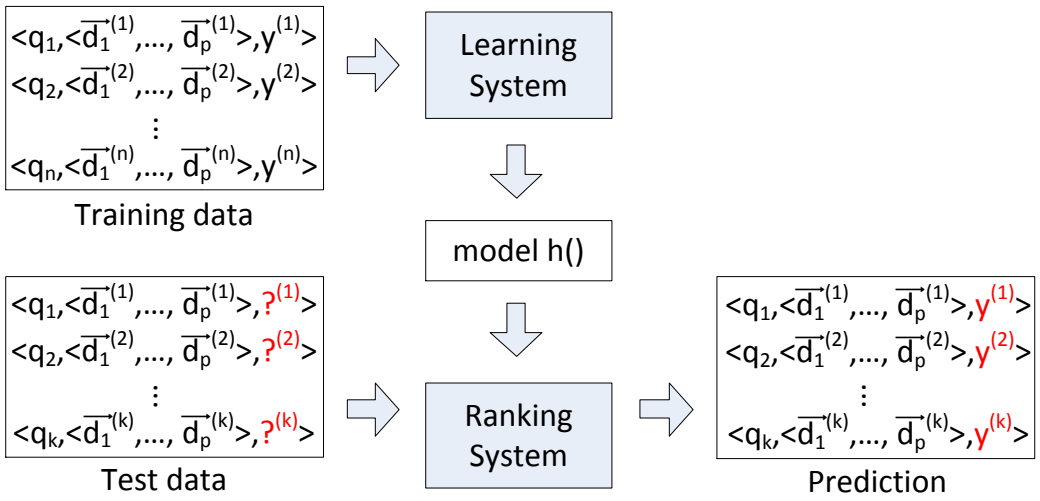

Figure 3: A general paradigm of L2R (copy from).

all ranking performance. Usually measures such as the Mean Average Precision (MAP) [Manning et al., 2008] and the Normalized Discount Cumulative Gain (NDCG) [Järvelin and Kekäläinen, 2002] are used for this evaluation.

L2R algorithms learn models by trying to minimize the difference between judgments and its prediction. The way they learn can be categorized into three approaches. The *pointwise* approach uses the documents' feature vectors as input to learn a model that predicts a relevance degree for each document (e.g. d is relevant or not). An example is PRank that learns through ordinal regression [Crammer and Singer, 2002]. The *pairwise* approach uses feature vectors of pairs of documents as input to learn a model that predicts relevance relations between pairs of documents (e.g. d_1 is more relevant than d_2). This approach enables to use clickthrough data from search engines as relevance judgments [Joachims, 2002]. The *listwise* approach uses as input a group of documents associated with a query to learn a model that minimizes the permutations between pairs of documents of ranked lists [Xia et al., 2008] or a model that minimizes IR measures used in evaluation, such as MAP [Yue et al., 2007].

While the pointwise approach only focuses on one document at a time, the

pairwise considers the dependency between documents. Even so, there is a gap between IR evaluation measures used to evaluate the model and measures used to learn the model. To overcome this, the listwise approach considers the position of the documents in the ranked list and their association with a query. As result, listwise algorithms tend to produce the best models and pointwise the worse. On the other hand, the listwise approach is more complex, while the first approaches that were developed, which were the pointwise and then the pairwise, can take advantage of existing theories and algorithms more easily. The three approaches present a trade-off between complexity and performance.

5.4 Discussion

Imagine what it would be like to find anything in an unsorted list of million of documents. Ranking models are an essential key for users to find information effectively and efficiently. They are one of the major reasons behind the search engines success. However, to maximize their performance they need to be tailor-made, depending on the IR system, the type of users and even on the type of queries. Thus, engineering the best ranking models for web archives is a challenging task that requires state-of-the-art technology in IR and learning to rank. Still, we should not only combine existent models, but also understand how users search and what data features they consider relevant to encode this knowledge into new models.

6 Conclusions

Every single day millions of web documents are created, updated and deleted. Some contain unique information that will be as valuable as books are today in our libraries. They contain knowledge for future generations that should be preserved and made accessible to everyone. In fact, this information is already being used currently for historical research, but also to improve technology, such as assessing the trustworthiness of statements [Yamamoto et al., 2007], detecting web spam

[Erdélyi and Benczúr, 2011] or improving current web information retrieval [Elsas and Dumais, 2010].

Giving access to all this information will enable to unfold web archives' full potential. However, this also brings new challenges: What index structures and architectures are more suitable for web archives? Can web search engine technology be used in web archives or is it just a good starting point? What services do users need? How do users search? How to provide the best ranked list of results and how to evaluate it? All these are open issues addressed in this Appendix that require further research.

References

O. Alonso and S. Mizzaro. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In *Proc. of the SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 15–16, 2009.

O. Alonso, M. Gertz, and R. Baeza-Yates. On the value of temporal information in information retrieval. *ACM SIGIR Forum*, 41(2):35–41, 2007.

O. Alonso, D. E. Rose, and B. Stewart. Crowdsourcing for relevance evaluation. *SIGIR Forum*, 42(2):9–15, 2008.

E. Amitay, D. Carmel, M. Herscovici, R. Lempel, and A. Soffer. Trend detection through temporal link analysis. *American Society for Information Science and Technology*, 55(14):1270–1281, 2004.

A. Anand, S. Bedathur, K. Berberich, R. Schenkel, and C. Tryfonopoulos. EverLast: a distributed architecture for preserving the web. In *Proc. of the 2009 Joint International Conference on Digital Libraries*, pages 331–340, 2009.

J. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In *Proc. of the 29th Annual International ACM SI-*

GIR Conference on Research and Development in Information Retrieval, pages 541–548, 2006.

R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co. Inc., 1999.

P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, and E. Yilmaz. Relevance assessment: are judges exchangeable and does it matter. In *Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 667–674, 2008.

S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *Proc. of the 16th International Conference on World Wide Web*, pages 501–510, 2007.

Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins. Sic transit gloria telae: towards an understanding of the web's decay. In *Proc. of the 13th International Conference on World Wide Web*, pages 328–337, 2004.

L. A. Barroso, J. Dean, and U. Hölzle. Web search for a planet: the Google cluster architecture. *IEEE Micro Magazine*, pages 22–28, 2003.

K. Berberich, M. Vazirgiannis, and G. Weikum. Time-aware authority ranking. *Internet Mathematics*, 2(3):301–332, 2005.

K. Berberich, S. Bedathur, T. Neumann, and G. Weikum. A time machine for text search. In *Proc. of the 30th SIGIR Conference on Research and Development in Information Retrieval*, 2007.

K. Berberich, S. Bedathur, O. Alonso, and G. Weikum. A language modeling approach for temporal information needs. *Advances in Information Retrieval*, pages 13–25, 2010.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In *Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, 2000.

M. Burner and B. Kahle. The Archive File Format. URL `http://www.archive.org/web/researcher/ArcFileFormat.php`.

C. Castillo. Effective web crawling. In *ACM SIGIR Forum*, volume 39, pages 55–56, 2005.

L. Clausen. Concerning etags and datestamps. In *Proc. of the 4th International Web Archiving Workshop*, volume 16, 2004.

C. Cleverdon. The Cranfield tests on index language devices. In *Aslib proceedings*, volume 19, pages 173–193, 1967.

M. Costa and M. J. Silva. Towards information retrieval evaluation over web archives. In *Proc. of the SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 37–40, 2009.

M. Costa and M. J. Silva. Understanding the information needs of web archive users. In *Proc. of the 10th International Web Archiving Workshop*, pages 9–16, 2010.

M. Costa and M. J. Silva. Characterizing search behavior in web archives. In *Proc. of the 1st International Temporal Web Analytics Workshop*, 2011.

K. Crammer and Y. Singer. Pranking with Ranking. *Advances in Neural Information Processing Systems*, 1(14):641–647, 2002.

N. Craswell, S. Robertson, H. Zaragoza, and M. Taylor. Relevance weighting for query independent evidence. In *Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 416–423, 2005.

W. Dakka, L. Gravano, and P. Ipeirotis. Answering general time-sensitive queries. *IEEE Transactions on Knowledge and Data Engineering*, 2010.

J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

J. Elsas and S. Dumais. Leveraging temporal dynamics of document content in relevance ranking. In *Proc. of the 3rd ACM International Conference on Web Search and Data Mining*, pages 1–10, 2010.

M. Erdélyi and A. Benczúr. Temporal analysis for web spam detection: An overview. In *Proc. of the 1st International Temporal Web Analytics Workshop*, pages 17–24, 2011.

S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems*, 23(2):147–168, 2005.

D. Gomes and M. Silva. Modelling information persistence on the web. In *Proc. of the 6th International Conference on Web Engineering*, page 200, 2006.

D. Gomes, S. Freitas, and M. J. Silva. Design and selection criteria for a national web archive. In *Proc. of the 10th European Conference on Research and Advanced Technology for Digital Libraries*, pages 196–207, 2006.

D. Gomes, A. Nogueira, J. Miranda, and M. Costa. Introducing the Portuguese web archive initiative. In *Proc. of the 8th International Web Archiving Workshop*, 2008.

A. T. W. Group. Use cases for access to Internet Archives. Technical report, Internet Preservation Consortium, 2006.

E. Hatcher and O. Gospodnetic. *Lucene in Action*. Manning Publications Co., 2004.

S. B. Irem Arikan and K. Berberich. Time Will Tell: Leveraging Temporal Expressions in IR. In *Proc. of the 2nd ACM International Conference on Web Search and Data Mining*, 2009.

E. Jaffe and S. Kirkpatrick. Architecture of the Internet Archive. In *Proc. of SYSTOR 2009: The Israeli Experimental Systems Conference*, pages 1–10, 2009.

B. Jansen and A. Spink. How are we searching the World Wide Web? A comparison of nine search engine transaction logs. *Information Processing and Management*, 42(1):248–263, 2006.

K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 154–161, 2005.

K. Jones and C. Van Rijsbergen. Report on the need for and provision of an 'ideal' information retrieval test collection. British Library Research and Development Report 5266. *Computer Laboratory, University of Cambridge*, 6, 1975.

R. Jones and F. Diaz. Temporal profiles of queries. *ACM Transactions on Information Systems (TOIS)*, 25(3), 2007.

A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with Mechanical Turk. In *Proc. of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems*, pages 453–456, 2008.

J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

X. Li and W. B. Croft. Time-based language models. In *Proc. of the 12th International Conference on Information and Knowledge Management*, pages 469–475, 2003.

T. Liu. *Learning to Rank for Information Retrieval*, volume 3 of *Foundations and Trends in Information Retrieval*. Now Publishers Inc., 2009.

T. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proc. of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, 2007.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

K. Markey. Twenty-five years of end-user searching, Part 1: Research findings. *American Society for Information Science and Technology*, 58(8):1071–1081, 2007.

J. Masanès. *Web Archiving*. Springer-Verlag New York Inc., 2006.

D. Metzler, R. Jones, F. Peng, and R. Zhang. Improving search relevance for implicitly temporal queries. In *Proc. of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 700–701, 2009.

S. Nunes, C. Ribeiro, and G. David. Using neighbors to date web documents. In *Proc. of the 9th Annual ACM International Workshop on Web Information and Data Management*, WIDM '07, pages 129–136, 2007.

C. Olston and M. Najork. Web Crawling. *Information Retrieval*, 4(3):175–246, 2010.

L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

T. Qin, T. Liu, J. Xu, and H. Li. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374, 2010.

F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *Proc. of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 239–248, 2005.

M. Ras and S. van Bussel. Web archiving user survey. Technical report, National Library of the Netherlands (Koninklijke Bibliotheek), 2007.

M. Richardson, A. Prakash, and E. Brill. Beyond PageRank: machine learning for static ranking. In *Proc. of the 15th International Conference on World Wide Web*, pages 707–715, 2006.

S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proc. of the 3rd Text REtrieval Conference*, pages 109–126, 1995.

A. I. T. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proc. of the IFIP/ACM International Conference on Distributed Systems Platforms*, pages 329–350, 2001.

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

M. Sanderson. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 162–169, 2005.

B. Schölkopf and A. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. the MIT Press, 2002.

R. Snow, B. O'Connor, D. Jurafsky, and A. Ng. Cheap and fast - but is it good?: evaluating non-expert annotations for natural language tasks. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263, 2008.

F. Song and W. Croft. A general language model for information retrieval. In *Proc. of the 8th International Conference on Information and Knowledge Management*, pages 316–321, 1999.

M. Stack. Full text searching of web archive collections. In *Proc. of the 5th International Web Archiving Workshop*, 2005.

I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proc. of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 149–160, 2001.

S. Strodl, C. Becker, R. Neumayer, and A. Rauber. How to choose a digital preservation strategy: Evaluating a preservation planning procedure. In *Proc. of the 7th ACM/IEEE-CS Joint Conference on Digital libraries*, pages 29–38, 2007.

T. Tao and C. Zhai. An exploration of proximity measures in information retrieval. In *Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 295–302, 2007.

B. Tofel. 'Wayback' for Accessing Web Archives. In *Proc. of the 7th International Web Archiving Workshop*, 2007.

L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 319–326, 2004.

L. von Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, 2008.

E. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management*, 36(5):697–716, 2000.

E. Voorhees and D. Harman. Overview of the eighth text retrieval conference (TREC-8). In *Proc. of the 8thd Text REtrieval Conference*, volume 8, pages 1–24, 1999.

T. White. *Hadoop: The Definitive Guide*. Yahoo Press, 2010.

F. Xia, T. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *Proc. of the 25th International Conference on Machine learning*, pages 1192–1199, 2008.

G. Xue, Q. Yang, H. Zeng, Y. Yu, and Z. Chen. Exploiting the hierarchical structure for link analysis. In *Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 186–193, 2005.

Y. Yamamoto, T. Tezuka, A. Jatowt, and K. Tanaka. Honto? search: Estimating trustworthiness of web information by search results aggregation and temporal analysis. *Advances in Data and Web Management*, pages 253–264, 2007.

J. Yeh, J. Lin, H. Ke, and W. Yang. Learning to rank for information retrieval using genetic programming. In *Proc. of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, 2007.

P. S. Yu, X. Li, and B. Liu. On the temporal dimension of search. In *Proc. of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*, pages 448–449, 2004.

Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 271–278, 2007.

H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S. Robertson. Microsoft Cambridge at TREC-13: Web and Hard Tracks. In *Proc. of the 13th Text REtrieval Conference*, 2004.

J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–314, 1998.

J. Zobel and A. Moffat. Inverted files for text search engines. *ACM Computing Surveys (CSUR)*, 38(2):6, 2006.