Creating a Billion-Scale Searchable Web Archive

Daniel Gomes, Miguel Costa, David Cruz, João Miranda, Simão Fontes
Foundation for National Scientific Computing
Av. Brasil, 101
1700-066 Lisboa, Portugal
{daniel.gomes, miguel.costa, david.cruz, joao.miranda, simao.fontes}@fccn.pt

ABSTRACT

Web information is ephemeral. Several organizations around the world are struggling to archive information from the web before it vanishes. However, users demand efficient and effective search mechanisms to access the already vast collections of historical information held by web archives. The Portuguese Web Archive is the largest full-text searchable web archive publicly available. It supports search over 1.2 billion files archived from the web since 1996. This study contributes with an overview of the lessons learned while developing the Portuguese Web Archive, focusing on web data acquisition, ranking search results and user interface design. The developed software is freely available as an open source project. We believe that sharing our experience obtained while developing and operating a running service will enable other organizations to start or improve their web archives.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process; H.3.7 [**Digital Libraries**]: Systems issues

General Terms

Web, Archive, Search, Preservation

Keywords

Portuguese Web Archive, Temporal Search

1. INTRODUCTION

Human knowledge has been incrementally built for thousands of years. The new generations augment knowledge transmitted by the previous ones. Inventions such as writing, press and recently the web, deeply improved this process. However, after a short period of time, the information published on the web becomes unavailable and commonly is lost forever. Several studies showed that only 20% of the pages available today last for more than one year [13].

Besides loosing our collective memory, such as important scientific and historical information, we are loosing our individual memories too. For instance, everyday people take photos and share them exclusively on the web without having the most elementary preservation concerns. In the future these people will have difficulties in showing portraits of their memories as their ancestors did.

For centuries, cultural heritage organizations, such as archives and libraries, ensured the preservation of information published through printed media. Since 1996, several web archiving initiatives were created worldwide [12]. Web archives acquire, store, preserve and provide access to information published on the web across time, which also includes content created before the digital era that was digitized and published online. This content includes official documents, such as those traditionally kept by libraries or museums, but also other kinds of publications that are valuable descriptions of recent history, such as blogs and online forums.

Archiving and preserving data from the web is not enough to make web archives useful for societies. Historical information must be searchable efficiently and effectively for enabling users to explore it. Web users expect performance similar to the one provided by live web search engines [28]. However, achieving this goal raises new challenges, because web search engine technology cannot be directly applied to web archives. There are several differences between both types of systems. For instance, web search engines redirect users to content hosted on their original servers. There is no concern with content preservation across time. On their turn, web archives preserve content and reproduce it as close as possible to its original format. Web archives must address the temporal dimension of data on all steps of the processing workflow.

The Portuguese Web Archive (PWA) began in 2008 and it aims to preserve web content of interest to the Portuguese community. It was based on the Archive-Access project tools [17], which are used by most web archives worldwide [12]. However, we observed that these tools did not fulfill users' requirements. Thus, we researched and developed a new web archive search engine. In January 2010, we released a beta version of a search service over the PWA available at `http://archive.pt`. In December 2012, the service provided public access to 1.2 billion (10^9) files. This paper presents the main lessons learned while developing our service, the research that sustained the adopted design decisions and the experience obtained from operating the system in a production environment. Its main contributions are an analysis of the application of deduplication mechanisms during data acquisition from the live web, the analysis of ranking models to support search over historical web collections and the redesign of a standard search engine interface to be applicable in the context of web archives. The developed software is available as an open source project at `http://code.google.com/p/pwa-technologies/`.

The remainder of this paper is organized as follows. In Section 2, we cover the related work. Section 3 presents

WWW 2013 Companion, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2038-2/13/05.

our strategies to select and acquire content, while Section 4 describes how search results are ranked. Section 5 explains how we adapted web search user interfaces to support web archives, and Section 6 finalizes with the conclusions.

2. RELATED WORK

Ideally, web archives should acquire and preserve every content published on the web. In this sense, web archives are similar to traditional printed media archives and selection policies must be applied to acquire the most relevant information according to the available resources [11]. Capturing web documents to be later reproduced is challenging, because it is necessary to interact with millions of web servers beyond our control [27].

Web archives face many challenges related to scalability and information overload because they accumulate previous files and indexes, unlike web search engines that tend to drop the old versions when new ones are discovered [2]. Web archives already hold more than 282 billion[1] files and this number continues to grow as new initiatives continue to arise. This data dimension is one order of magnitude larger than the number of files indexed by the largest web search engine and 150 times more than the content of the Library of Congress.

About 89% of web archives provide URL search [12], mostly supported by the open source Wayback Machine [33], which returns a list of chronologically ordered versions of a given URL. However, this type of search forces users to know the URL of the file that contains the required information, which may have disappeared many years ago.

The National Library of the Netherlands conducted a usability test on the searching functionalities of its web archive and derived a list of the top 10 functions that users would like to see implemented [28]. Full-text search was the first ranked, followed by URL search. At least 67% of web archives support full-text search for a part of their collections [12]. The large majority of full-text search that these web archives support is based on the Lucene search engine [14], which is the core of NutchWAX [17]. However, its performance was considered unsatisfactory by stakeholders [12, 31]. Cohen et al. showed that the out-of-the-box Lucene produces low quality results, presenting half of the precision of the best systems participating in the TREC Terabyte track [4]. In addition, Costa and Silva proposed an evaluation methodology based on the Cranfield paradigm and measured the effectiveness of state-of-the-art information retrieval (IR) technology employed in web archives [8]. Results confirmed the poor quality of search results retrieved with this state-of-the-art technology based on Lucene and NutchWAX. The authors also achieved statistically significant improvements by leveraging temporal information intrinsic to web archives.

Other works leveraged temporal information, but to improve results of web search engines. One of the most common ideas is extending language models to bias the document's prior probability of being relevant to favor the most recent documents [20]. Boosting the most recent documents is desirable for queries where the user intends to find recent events or breaking news. 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 [10]. Different works explored the distribution of the documents' dates, since it reveals time intervals that are likely to be of interest to the query [18]. For instance, when searching for *tsunami*, the peaks in the distribution may indicate when tsunamis occurred.

Costa and Silva studied the information needs and search behaviors of web archive users [6, 7]. The main conclusions were that users from web archives and web search engines have different information needs. However, they maintain the same search behavior when using both types of systems, ignoring that they are searching historical collections. This behavior is a motivation to research user interfaces for searching historical web collections. New interfaces have been mainly focused on the exploitation of historical curated collections using elements, such as timelines or information clusters [1]. However, this research does not address the specific requirements of searching historical web collections. The analyzed data was obtained from curated online news archives, which is not representative of the heterogeneity of data addressed by web archives. Hearst's book presents a comprehensive analysis of user interface design to search the live web, but searching over historical web collections is not addressed [16].

Web archiving research projects have been receiving a growing support from the European Commission. The Living Web Archives (LiWA) aimed to provide contributions to make archived information accessible and not just stored [22]. It addressed problems shared with other IR areas, such as web spam detection, terminology evolution, capture of stream video, and assuring temporal coherence of archived content. LiWA was followed by the Longitudinal Analytics of Web Archive data (LAWA), which aims to build an experimental testbed for large-scale data analytics [34]. Particular emphasis is given to web data analysis that has been crawled over extended time periods.

3. ACQUIRING WEB DATA

Web archives must acquire content before it disappears. Data acquisition is typically made through web crawling.

3.1 Crawling policy

Crawling is a data acquisition method widely used by web archives to capture information for preservation and later access. Besides the textual content to support full-text search, web archives must exhaustively gather embedded files (e.g. CSS, JavaScript or images) to enable the reproduction of archived pages. Thus, crawling for archiving imposes a higher workload on visited web servers in comparison to search engines. Still, the crawler must be polite to the visited web servers. The Heritrix crawler v.1.14.3 was used to acquire content from the live web [24]. It was configured based on the information derived from past web characterizations and crawling experiences [23].

The objective of the PWA is to preserve web content relevant to the Portuguese community. This selection criteria is highly subjective and must be transposed to machine-understandable rules. Currently, the PWA performs two types of crawls from the live web. *Trimestral broad crawls* are performed every 3 months and include a broad set of approximately 500 thousand seed URLs derived from the national top-level domain listings, user submissions, web directories of Portuguese speaking countries and home pages

[1] `http://en.wikipedia.org/wiki/List_of_Web_archiving_initiatives`

of sites successfully harvested on the previous crawl. On average, 78 million files are downloaded each crawl (5.9 TB). The objective of trimestral crawls is to archive exhaustive snapshots of the national web. The *daily selective crawls* are performed every day and include a set of 359 online publications selected in collaboration with the National Library of Portugal, typically online newspapers and magazines. As these types of publications receive heavier load during work hours, our crawl begins at 16:00 and reaches 90% of the URLs to visit at 7:00 of the next day. On average, 764 thousand files are downloaded each day (42 GB).

3.2 Crawling pitfalls

The crawler got trapped several times in infinite sites, while performing trimestral broad crawls. There are millions of sites available on the web, but they tend to be supported by a small number of publishing platforms. This fact enabled the automatic detection of infinite sites through regular expression matching applied to URLs. We also avoid following links that originate data insertion, such as comments on well known platforms. Our black list and exclusion rules are available at `http://arquivo.pt/crawlfilters` to be reused and validated by the community.

During our crawls we respect the access rules established by authors through the Robots Exclusion Protocol (REP). This best practice avoids unwanted access complaints by site owners and harvesting infinite parts of a site, such as online calendars. However, it raised unexpected problems. Social network platforms, such as Facebook, are commonly used by organizations and individuals to publish information in replacement of traditional sites. The problem is that Facebook is very restrictive regarding crawlers and these rules cannot be changed by users. A formal authorization request to Facebook is required to crawl its content[2]. Nonetheless, the crawl of some types of content, such as photos, is always forbidden. Even with the obtained authorization Facebook pages are difficult to crawl, because they are strongly based on AJAX technology and chained redirects, which make it difficult to discover and acquire embedded or linked content.

Some popular Content Management Systems, such as Joomla [3], also present default REP restrictions that do not allow crawlers to harvest all files. As web search engines just need to crawl textual content to present results from a site, the default REP rules forbid the crawl of embedded files that are mandatory to archive the entire web page. This situation had a significant impact on the daily crawls. Hence, we contacted via email or contact forms the webmasters of the sites that had REP rules to raise awareness about this issue. Only 10% of the sent messages originated a reaction by the webmasters, suggesting that default access rules tend to prevail.

3.3 Incremental crawling and deduplication

We adopted the DeDuplicator plug-in for Heritrix to detect and avoid the storage of duplicate content [30]. This plug-in analyses the log of the previous crawl and builds an index containing the cryptographic digest of each URL. Heritrix downloads a content from a URL and compares it with the digest of the version harvested on the previous crawl. If it remains unchanged, it is discarded. To measure the impact of adding the DeDuplicator, we measured the download rate for daily and trimestral crawls. The obtained results show that the addition of the DeDuplicator did not have significant impact on the crawler performance.

The amount of data archived is incremental. Thus, the waste of resources caused by the storage of duplicates is significant. Among the files downloaded in a daily crawl, 46% were duplicates. In trimestral crawls the level of duplication was 30%. Eliminating duplicates in web archives saves a considerable amount of disk space, but also reduces the amount of data to be indexed. The indexes size decreased 20% in the daily crawls and 23% in the trimestral crawls.

We analyzed the distribution of files stored per media type and observed that images dropped from 32.5% to 14% when duplicates were discarded. The HTML pages increased from 60.1% to 78.8%. This means that images tend to be more persistent and originate more duplicates.

The elimination of duplicates may be dangerous from a preservation perspective. For instance, if a frequently updated page includes an image that remains unchanged across time, the image is crawled and stored just once, while several new versions of the page are archived. Thus, if that image is lost, all versions of the page that contain it will become incomplete. To minimize this preservation problem, we perform a crawl without deduplication in the beginning of each year.

4. RANKING THE PAST WEB

Ranking models order results based on how well they match user queries. Since millions of document versions archived across time can match a query, ranking models are crucial for users to find the desired information. This section describes and compares experimental approaches to create ranking models for supporting search over historical web collections.

4.1 Ranking with TREC test collections

When we started developing the PWA, there were not suitable evaluation methodologies and test collections for web archive information retrieval (WAIR). These resources have been a driver of research and innovation in information retrieval (IR), which enable to compare systems and measure their progress. Thus, we initially used the available test collections closer to our goal, which were built for evaluating web search engine technology [15]. We used the TD2003 and TD2004 datasets from the TREC 2003 and TREC 2004 web tracks.

For each query, we extracted the 1 000 top documents ranked by Okapi's BM25 [29]. The top documents were then scored by combinations of 30 ranking features. These features included term-weighting features, such as the term frequency or Okapi's BM25, and term-distance features, such as the Minimal Span Weighting [25]. These features operate over different document fields: URL, title, text body and anchor text of incoming links.

We used a feature selection algorithm to remove irrelevant and redundant features. This selection algorithm selects in each iteration, the feature f_i that leads to the highest gain when combined with the previously selected features S. The algorithm iterates until the gain of adding a new feature f_i is lower than a defined threshold. The weights among the subset of features $S + f_i$ were tuned in each iteration by the learning to rank (L2R) algorithm SVM-MAP [35]. Thus,

[2] `http://www.facebook.com/apps/site_scraping_tos_terms.php`

the computational cost of processing many ranking features was balanced against the ranking quality.

Four features were selected from the 30, which compose the model denoted as MdTREC. Lucene[3] was the only term-weighting function selected, being preferred to BM25 and TFxIDF. Adding the latter also did not improve significantly the quality of results. On the other hand, adding functions that quantify the distance between query terms did improve. The selected ranking model also includes the MinSpanCov$_{unord}$ function over the title and content fields. The function returns the length of the shortest segment of text containing two query terms. This length is then transformed by an exponential decay function based on [32]:

$$\text{MinSpanCov}(q, d, span) = \ln(1 + e^{-span-length(q)+1}) \quad (1)$$

A variant of this function, denoted MinSpanCov$_{ord}$, was selected over the anchor field. In this, the shortest segment of text contains the terms in the same order as in the query.

4.2 Ranking with a WAIR test collection

We proposed an evaluation methodology and built a test collection for WAIR, because existing evaluation methodologies and test collections from evaluation campaigns, such as TREC, are not suitable for web archives [8]. These collections have different task goals and characteristics. For instance, previous test collections did not have a temporal dimension, where each document may have several versions throughout time and their relevance depends of the user's period of interest. Our test collection is named PWA9609[4].

As proof of concept, we leveraged temporal information intrinsic to web archives to build a better ranking model. Our assumption was that the more relevant documents are maintained longer. We implemented two time-aware models under this assumption that give a higher score to: (1) documents with more versions; (2) documents with a larger time span between the first and last archived versions. Both models are defined by the same function:

$$f(v^d_{t_i}) = \log_y(x) \quad (2)$$

where, for the first case, x is the number of versions of document d and, for the second case, x is the number of days between the first and last versions of document d. y is the maximum possible x for normalization. Each of these functions was linearly combined with the NutchWAX's term-weighting function [8]. We generally denote these linearly combined models by TVersions and TSpan, respectively.

4.3 Ranking with a L2R dataset

Supervised learning algorithms have been employed to tune the weights between combined ranking features resulting in significant improvements [21]. This research area is called learning to rank (L2R) and was mostly driven by web IR. Typical ranking models from web search engines combine hundreds, or even thousands, of features. For instance, a L2R dataset released by Yahoo! includes 700 features[5].

We created a L2R dataset for researching learning to rank applied to WAIR. This dataset named L2R4WAIR, is composed by 39 608 vectors of 68 features extracted from <*query, URL, timestamp*> triplets, along with relevance judgment labels. The relevance judgments have a three-level scale: very relevant, relevant, and not relevant. Both features and relevance judgments were obtained from the PWA9609 test collection, which contains 50 navigational queries [8]. One third of these queries have temporal restrictions.

The L2R4WAIR dataset contains different types of features, some time-unaware and others time-aware. The time-unaware include term-weighting features and term-distance features, such as the ones described above. The time-aware features include the age of a version measured from the query issued time and the features described above, such as TVersions and TSpan.

We tuned the weights between the features with the RankBoost algorithm available in the RankLib open-source Java package[6]. RankBoost produces linear models that are supported by the PWA search system. We could also have used any other L2R algorithm with the L2R4WAIR dataset.

4.4 Results

We compared all ranking models described above using the PWA9609 test collection built to evaluate WAIR. Table 1 presents the results of the ranking models measured with the Normalized Discount Cumulative Gain at cut-off k (nDCG@k), the Precision at cut-off k (P@k), and the Success at rank k (S@k). The bold entries indicate the best result for each IR measure. We can see that the state-of-the-art typically used in web archives (i.e. Lucene and NutchWAX) present poor results and their effectiveness is similar. Despite the MdTREC model tuned with the TREC collections had presented good results when evaluated with the same TREC collections, it presented results close to the state-of-the-art when evaluated with the PWA9609 test collection. This indicates that IR test collections built for web search engines are not suitable for web archives.

A clear pattern is that the time-aware models (i.e. with temporal features) give better results than the time-unaware, showing statistical significant improvements of $p<0.05$ against NutchWAX in most IR measures, when using a two-tailed paired Student's t-test. This strongly indicates that the use of temporal information improves the search effectiveness in web archives.

The model derived from the L2R4WAIR dataset and optimized with the RankBoost algorithm presents the best results. The improvements are very significant. For instance, when compared with NutchWAX, the MdRankBoost model achieved nDCG@1, nDCG@5 and nDCG@10 values of 30%, 26.5% and 20.2% higher, respectively. These preliminary results show that L2R is a promising technology to use in WAIR.

5. DESIGNING THE USER INTERFACE

5.1 Layout & Functionalities

Most web archives were created and are managed by libraries [12]. The digital library user interfaces address the temporal dimension of data and are based on faceted search (e.g. search by metadata, such as title and author). This

[3] `http://lucene.apache.org/java/2_9_0/api/all/org/apache/lucene/search/Similarity.html`

[4] `http://code.google.com/p/pwa-technologies/wiki/TestCollection`

[5] `http://learningtorankchallenge.yahoo.com`

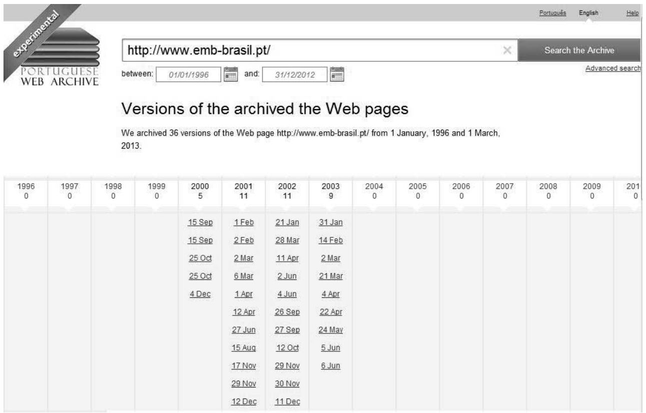
[6] `http://www.cs.umass.edu/~vdang/ranklib.html`

Metric	time-unaware			time-aware		
	Lucene	NutchWAX	MdTREC	TVersions	TSpan	MdRankBoost
nDCG@1	0.220	0.250	0.250	0.430 †	0.450 †	**0.550** †
nDCG@5	0.157	0.215	0.209	0.266 †	0.263 †	**0.510** †
nDCG@10	0.133	0.174	0.177	0.202 †	0.193	**0.555** †
P@1	0.280	0.320	0.320	0.500 †	0.520 †	**0.600** †
P@5	0.164	0.236	0.232	0.264	0.256	**0.276**
P@10	0.132	0.168	0.170	0.172	0.158	**0.194**
S@1	0.280	0.320	0.320	0.500 †	0.520 †	**0.600** †
S@5	0.500	0.680	0.600	**0.780** †	0.760	**0.780** †
S@10	0.600	0.780	0.780	**0.840**	0.760	**0.840**

† shows a statistical significance of $p<0.05$ against NutchWAX

Table 1: Results obtained for the tested ranking models.

leads to complex user interfaces composed by user interface (UI) elements that require strong contextualization and decisions by the users to provide relevant search results. On its turn, the typical web search engine interface is simpler and more familiar to users [9]. The downside is that it usually does not consider the temporal dimension of data.



http://www.emb-brasil.pt/

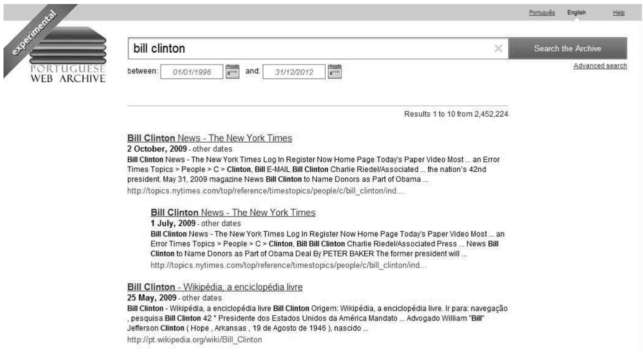
between: 01/01/1996 and: 31/12/2012

Versions of the archived the Web pages

We archived 36 versions of the Web page http://www.emb-brasil.pt/ from 1 January, 1996 and 1 March, 2013.

1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	201
0	0	0	0	5	11	11	9	0	0	0	0	0	0	0
				15 Sep	1 Feb	21 Jan	31 Jan							
				15 Sep	2 Feb	28 Mar	14 Feb							
				25 Oct	2 Mar	11 Apr	2 Mar							
				25 Oct	6 Mar	2 Jun	21 Mar							
				4 Dec	1 Apr	4 Jun	4 Apr							
					12 Apr	26 Sep	22 Apr							
					27 Jun	27 Sep	24 May							
					15 Aug	12 Oct	5 Jun							
					17 Nov	29 Nov	6 Jun							
					29 Nov	30 Nov								
					12 Dec	11 Dec								

Figure 1: Result page for a URL search on the Portuguese Web Archive.



bill clinton

between: 01/01/1996 and: 31/12/2012

Results 1 to 10 from 2,452,224

Bill Clinton News - The New York Times
2 October, 2009 - other dates
Bill Clinton News - The New York Times Log In Register Now Home Page Today's Paper Video Most ... an Error Times Topics > People > C > **Clinton**, **Bill** E-MAIL **Bill Clinton** Charlie Riedel/Associated ... the nation's 42nd president. May 31, 2009 magazine News **Bill Clinton** to Name Donors as Part of Obama ...
http://topics.nytimes.com/top/reference/timestopics/people/c/bill_clinton/ind...

> **Bill Clinton** News - The New York Times
> **1 July, 2009** - other dates
> **Bill Clinton** News - The New York Times Log In Register Now Home Page Today's Paper Video Most ... an Error Times Topics > People > C > **Clinton**, **Bill Bill Clinton** Charlie Riedel/Associated Press ... News **Bill Clinton** to Name Donors as Part of Obama Deal By PETER BAKER The former president will ...
> http://topics.nytimes.com/top/reference/timestopics/people/c/bill_clinton/ind...

Bill Clinton - Wikipédia, a enciclopédia livre
25 May, 2009 - other dates
Bill Clinton - Wikipédia, a enciclopédia livre **Bill Clinton** Origem: Wikipédia, a enciclopédia livre. Ir para: navegação , pesquisa **Bill Clinton** 42 ° Presidente dos Estados Unidos da América Mandato ... Advogado William "**Bill**" Jefferson **Clinton** (Hope , Arkansas , 19 de Agosto de 1946), nascido ...
http://pt.wikipedia.org/wiki/Bill_Clinton

Figure 2: Result page for a full-text search on the Portuguese Web Archive.

We decided to use a typical web search engine interface as starting point and iteratively enhanced it to address the temporal dimension of web archive search. A web archive UI must address temporal search restrictions (e.g. definition of date interval), versioning of URLs on search results and reproduction of archived content with metadata for temporal contextualization (e.g. crawl date). The PWA interface is available at `http://archive.pt` and it is composed by:

Archived content view: presents the archived content along with the original URL and crawl date. It supports link navigation within the archive;

URL versions list (Figure 1): a chronology of crawled versions from a given URL in a yearly grid. Each date links to the archived content view;

Full-text search results list (Figure 2): a list of typically 10 results, where each result shows the title that links to the archived content, its crawl date, a *other dates* link to the URL versions list, and a snippet of the content containing the query terms;

Search form (Figures 1 and 2): is presented on the top of the URL versions list and full-text search results list. It is composed by a search box that receives the query terms and two datepickers to restrict the crawl dates of the content to be searched;

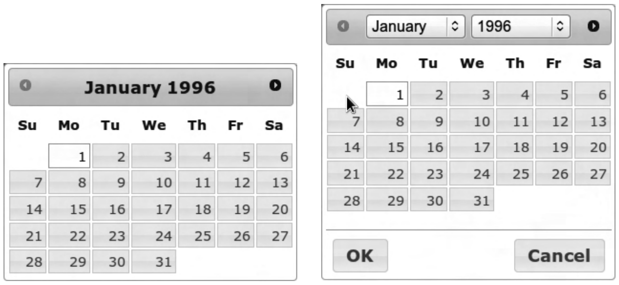
Advanced search form: enables users to refine search by phrase, term negation, crawl dates, file format, site and support for choosing the number of presented results, and sorting by date and relevance.

5.2 Usability Tests & Improvements

During the development of our search UI we performed several iterations of laboratory usability tests to identify interaction problems and measure progress. Each testing round consisted of 10 tasks performed by 6 users following the *think aloud method* [26]. Each of the users executed the test individually in the presence of a usability expert. The audio and screen captures of the user sessions were recorded for later analysis. Each user filled a pre-questionnaire to establish a profile and a post-questionnaire for measuring their satisfaction [19].

We observed that users compared the behavior of their favorite search engine with our web archive and expected the same response speed and search result quality due to the UI similarity. They did not understand the difference between searching the live web and historical web collections. Using a web archive to access pages that are no longer available on the live web is a confusing concept to most users, which requires technical knowledge about the functioning of the Internet. Only 20% of the users answered that they knew what a web archive was.

Users typically ignored the dates and links to the URL versions list on the full-text search results list. However, when



Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

(a)

Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

(b)

Figure 3: Datepicker (a) of JQuery JavaScript library (b) adapted to web archive interface.

in the URL versions list, the grid layout enabled users to perceive the version distribution by year. Initially, the date presented on the full-text search results list and archived content view was labeled as *crawl date*. However, the technical meaning of *crawl* was not understandable to the users. After several tries, we found that the best option was to omit any label and let the users loosely interpret the meaning of the presented date.

Throughout usability testing we observed that users did not have difficulties with the search form. It consists of elements that users were used to find in search engines: a text field and a submit button. However, it also contain two datepickers for delimiting the timespan of the search. The introduction of the datepickers raised unexpected challenges, because the adoption of a conventional datepicker, obtained from the jQuery UI library (Figure 3(a)), did not meet the web archive users expectations. Conventional datepickers are meant to specify days or short intervals of time, but in web archives the time intervals can be very small, such as a specific day, or very broad, spanning several years. Unexpectedly, the datepickers became a problematic UI element that required several design and evaluation iterations. Figure 3(b) presents the final version of the datepicker. Left and right arrows are useful for month navigation, but did not work to define date ranges of several years. Thus, dropdown lists for month and year selection were added. The datepickers were complemented with text fields to enable direct typing of dates.

The PWA supports full-text and URL search. Our first UI prototype was composed by two distinct search forms: one for full-text search and another one for URL search. This approach failed, because users did not understand the difference between search types. They inserted full-text queries on the URL search form, and vice-versa. The solution was to present a single textfield that receives any query. If the query is composed by a URL, the corresponding URL versions list is presented to the user. If the query includes a URL and other terms, which we observed in 9.6% of the cases [7], the web archive does a full-text search with the query terms, but also presents a suggestion link to the versions list of the queried URL. Otherwise, it performs a full-text search for the query terms. The URL queries are expanded to find results crawled with different URL aliases, that are likely to refer the same content. For instance, with and without *www.* prefix or *index.html* suffix.

Users frequently mistyped queries and blamed the web archive for poor search results. The addition of a query spellchecker had great impact on the perceived quality of the web archive, which led to fewer negative comments [5].

The presented changes increased the overall user satisfaction from 51% on the first version of the UI to 71% on the last one.

6. CONCLUSIONS

This study shares the experience obtained while creating a fully searchable web archive from data acquisition to user interface design. We concluded that excluding files that remain unchanged across time significantly saves storage space for archiving files and for index structures.

Finding the desired information in a web archive containing billions of files with several versions acquired throughout time is challenging. Using supervised learning algorithms to combine temporal features implicitly hidden in the archive, along with the regular topical features, provides better results than the state-of-the-art IR typically used in web archives. Moreover, web search user interfaces must be adapted to web archives. Users have well defined expectations about search interfaces and they are not receptive to new UI elements, even if they are conventional elements such as datepickers. However, the usability of web archive search interfaces can be significantly improved by tweaking conventional UI elements.

Web archives have been storing information for years. However, search over historical web collections is giving its first steps. We believe that the provided contributions represent a breakthrough in web archiving and are a baseline to develop more sophisticated searchable web archives in the future. The source code of the developed web archive is available as a free open source project at `http://code.google.com/p/pwa-technologies/`.

7. ACKNOWLEDGMENTS

We thank the Internet Archive for supplying us web collections from the .PT domain and for its continuous efforts to preserve and grant access to human knowledge. We acknowledge Marco de Sá and Rui Lopes for their precious collaboration in the user interface design.

8. REFERENCES

[1] 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.

[2] O. Alonso, J. Strötgen, R. Baeza-Yates, and M. Gertz. Temporal Information Retrieval: Challenges and Opportunities. In *Proc. of the 1st International Temporal Web Analytics Workshop*, pages 1–8, 2011.

[3] S. Burge. *The Joomla SEO Book*. Alledia Inc., 2007.

[4] 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.

[5] M. Costa, J. Miranda, D. Cruz, and D. Gomes. Query suggestion for web archive search. Technical report, Foundation for National Scientific Computing, 2012.

[6] 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.

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

[8] M. Costa and M. J. Silva. Evaluating web archive search systems. In *Proc. of the 13th International Conference on Web Information Systems Engineering*, pages 440–454, 2012.

[9] C. De Rosa, J. Cantrell, J. Hawk, and A. Wilson. *College Students' Perceptions of Libraries and Information Resources: A Report to the OCLC Membership.* OCLC, 2006.

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

[11] 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.

[12] D. Gomes, J. Miranda, and M. Costa. A survey on web archiving initiatives. In *Proc. of the International Conference on Theory and Practice of Digital Libraries*, 2011.

[13] D. Gomes and M. Silva. Modelling information persistence on the web. In *Proc. of the 6th International Conference on Web Engineering*, pages 193–200, 2006.

[14] E. Hatcher and O. Gospodnetic. *Lucene in action.* Manning Publications Co., 2004.

[15] D. Hawking and N. Craswell. Very large scale retrieval and web search. In E. Voorhees and D. Harman, editors, *The TREC Book.* MIT Press, 2004.

[16] M. Hearst. *Search User Interfaces.* Cambridge University Press, 2009.

[17] Internet Archive. Nutchwax - Home Page. `http://archive-access.sourceforge.net/`, March 2008.

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

[19] J. R. Lewis. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *Int. J. Hum.-Comput. Interact.*, 7(1):57–78, 1995.

[20] 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.

[21] T. Liu. *Learning to rank for information retrieval*, volume 3 of *Foundations and Trends in Information Retrieval.* Now Publishers Inc., 2009.

[22] J. Masanès. LiWA news #3: Living web archives. `http://liwa-project.eu/images/videos/Liwa_Newsletter-3.pdf`, March 2011.

[23] J. Miranda and D. Gomes. An Updated Portrait of the Portuguese Web. In *14th Portuguese Conference on Artificial Intelligence (EPIA)*, 2009.

[24] G. Mohr, M. Kimpton, M. Stack, and I. Ranitovic. Introduction to Heritrix, an archival quality web crawler. In *Proc. of the 4th International Web Archiving Workshop*, 2004.

[25] 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.

[26] J. Nielsen and H. Loranger. *Prioritizing Web Usability.* New Riders, 2006.

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

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

[29] 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.

[30] K. Sigurdsson. Managing duplicates across sequential crawls. In *Proc. of the 6th International Web Archiving Workshop*, 2006.

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

[32] 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.

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

[34] G. Weikum, N. Ntarmos, M. Spaniol, P. Triantafillou, A. A. Benczur, S. Kirkpatrick, P. Rigaux, and M. Williamson. Longitudinal analytics on web archive data: It's about time! In *Proc. of the 5th Conference on Innovative Data Systems Research*, pages 199–202, 2011.

[35] 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.