Summer 2011

Using the Web Infrastructure for Real Time Recovery of Missing Web Pages

Martin Klein
Old Dominion University

Recommended Citation
Klein, Martin. "Using the Web Infrastructure for Real Time Recovery of Missing Web Pages" (2011). Doctor of Philosophy (PhD), dissertation, Computer Science, Old Dominion University, DOI: 10.25777/jdht-6564
https://digitalcommons.odu.edu/computerscience_etds/20

USING THE WEB INFRASTRUCTURE FOR REAL TIME RECOVERY OF MISSING WEB PAGES

by

Martin Klein
Diploma November 2002, University of Applied Sciences Berlin, Germany

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
August 2011

Approved by:

Michael L. Nelson (Director)

Yaohang Li

Michele C. Weigle

Mohammad Zubair

Robert Sanderson

Herbert Van de Sompel

ABSTRACT

USING THE WEB INFRASTRUCTURE FOR REAL TIME RECOVERY OF MISSING WEB PAGES

Martin Klein
Old Dominion University, 2011
Director: Dr. Michael L. Nelson

Given the dynamic nature of the World Wide Web, missing web pages, or "404 Page not Found" responses, are part of our web browsing experience. It is our intuition that information on the web is rarely completely lost, it is just missing. In whole or in part, content often moves from one URI to another and hence it just needs to be (re-)discovered. We evaluate several methods for a "just-in-time" approach to web page preservation. We investigate the suitability of lexical signatures and web page titles to rediscover missing content. It is understood that web pages change over time which implies that the performance of these two methods depends on the age of the content. We therefore conduct a temporal study of the decay of lexical signatures and titles and estimate their half-life. We further propose the use of tags that users have created to annotate pages as well as the most salient terms derived from a page's link neighborhood. We utilize the Memento framework to discover previous versions of web pages and to execute the above methods. We provide a workflow including a set of parameters that is most promising for the (re-)discovery of missing web pages.
We introduce *Synchronicity*, a web browser add-on that implements this workflow. It works while the user is browsing and detects the occurrence of 404 errors automatically. When activated by the user Synchronicity offers a total of six methods to either rediscover the missing page at its new URI or discover an alternative page that satisfies the user's information need. Synchronicity depends on user interaction which enables it to provide results in *real time*.

Dedicated to a plate, may it always be full of shrimp!

ACKNOWLEDGMENTS

This is the part where you thank God if you are Christian, your spouse if you are married and your parents in case you get along.

However, first and foremost I would like to thank my advisor Dr. Michael L. Nelson for his eternal support and patience as well as superb guidance throughout my time at Old Dominion University. It goes without saying that this work would not have been possible without his inspiration and mentoring. He provides a prolific research environment enabling students to reach their top performance. Simply put, Michael is my role model (even though I could not care less for old cars and like dogs better than cats).

I am an unmarried atheist but there are others that I would like to mention here. I am very grateful to my dissertation committee for their support and the input they have provided. In particular Dr. Mohammad Zubair who suggested the use of web pages' titles for our purpose which inarguably added a new dimension to this work. Along the way numerous outstanding researchers have directly and indirectly contributed to this work. Early on I was in the fortunate position to work with Terry Harrison, Frank McCown and Joan A. Smith who all taught me how to survive in the US, how to be a good student and they all led by example for how to become a successful researcher. I am very thankful to Dr. Gary Marchionini (University of North Carolina Chapel Hill) and Dr. C. Lee Giles (Pennsylvania State University) who provided valuable feedback during my dissertation. Working with fellow students such as Charles Cartledge, Moustafa Emara, Jeb Ware and Jeffery Shipman was also a very rewarding experience.

I would like to especially thank Dr. Michael C. Overstreet who enabled me to join the ODU family and made me feel very welcome.

This work would not have been possible without financial support. I am grateful to the Library of Congress and the National Digital Information Infrastructure and Preservation Program (NDIIPP) as well as the National Science Foundation, grant IIS 0643784. I further received support from the Computer Science Department and the ODU College of Sciences. The ACM SIGWEB provided generous travel support enabling me to attend various international conferences and present my work.

I do have a great relationship to my parents and so they will not be left unmentioned. When they always have a place for you to withdraw and recharge, distract you when needed and push you when you are in danger of slipping away then you know you could not be more thankful to your parents.

I owe a lot to Linda, she is my everything.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

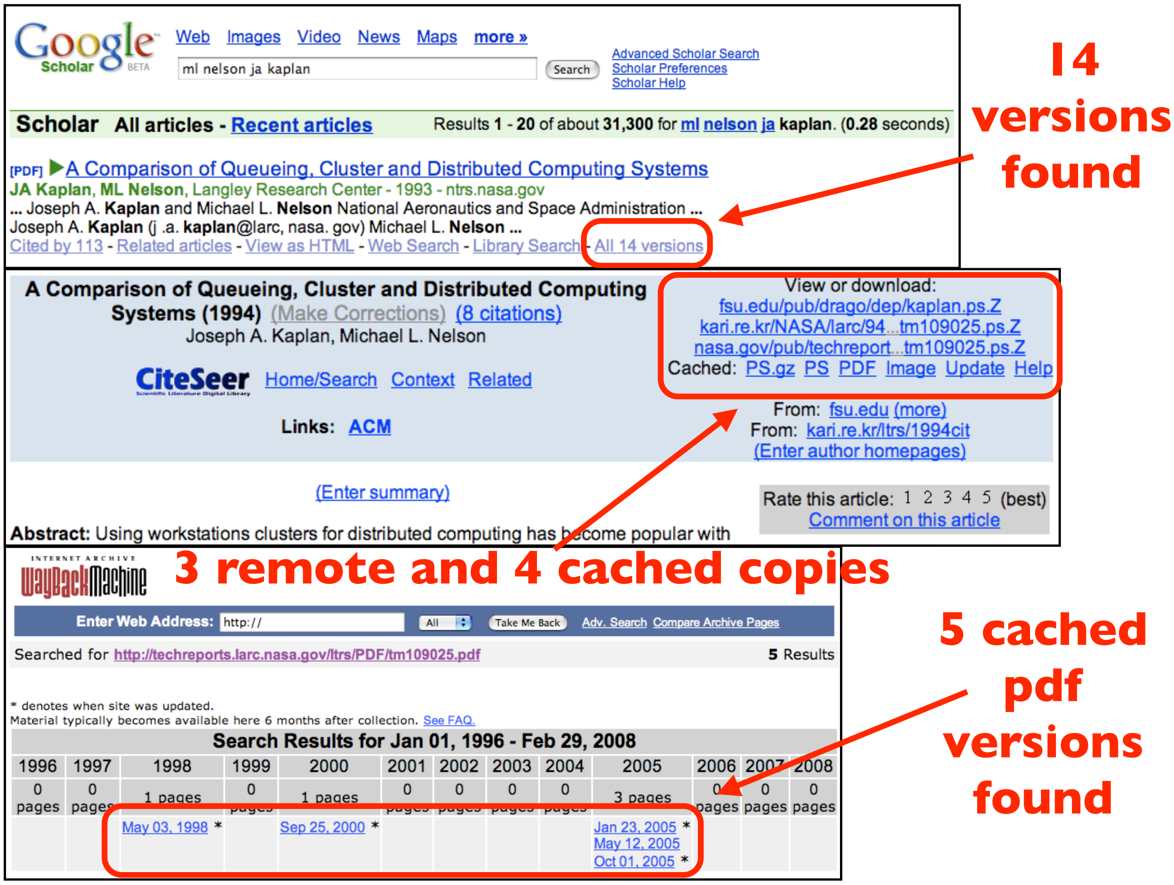
CHAPTER I

INTRODUCTION

Digital data preservation is often seen as an institutional effort. That means archives, libraries and other for-profit or non-profit organizations spend a lot of time and money on applying preservation services to their own collections. The commonly known preservation approaches are refreshing (copying bits to different systems), migration (transferring data to newer system environments) and emulation (replicating the functionality of obsolete systems) [229, 228]. The underlying assumption is that digital data will eventually become unavailable and inaccessible meaning devices used to access the data will become inoperable. Therefore the argument is to invest in a preservation effort now to yield persistent data available and accessible for humanity far in the future. This approach follows a "just-in-case" philosophy. The World Wide Web, somewhat as a by-product, provides what can be called "just-in-time" preservation [196]. It occurs naturally but still randomly in the "living web" where people create, copy and move content. This rather passive kind of preservation is enabled due to search engine services, web archival efforts and digital library research projects. It is neither supervised nor does it follow quality control guidelines but it has the invaluable benefit of millions of users contributing worldwide around the clock. The web is not controlled by one entity which decides what digital resource is important and needs to be preserved. It rather follows popular demand while also allowing uncountable niches to exist. While this "democracy" also bears the risk of single entities going out of business or shifting their operational focus, the web, in its unity with aggregated efforts of its users, will ensure preservation.

Emulating resources is primarily done on desktop computers using client software even though it seems to be a matter of time until the gap to networked and distributed emulation systems will be closed. However, examples for refreshing and migration can already be found in the web. Figure **??** shows the refreshing and migration of a web document. The original document was published as a NASA technical memorandum in 1993 as a compressed PostScript (.ps.Z) file on a `nasa.gov` website. Google Scholar has more than ten versions of the paper indexed, CiteSeer offers three remote and four local copies of the document and the Internet Archive (IA) provides five cached versions of the pdf document - all instances of refreshing. Although NASA eventually migrated the report into a PDF document, CiteSeer performed that migration independently and also migrated the document into PNG format. Yahoo! and Google provide dynamic conversion to HTML. It probably would be much harder now to completely eradicate the content of this resource from the web than it was to publish it in 1993.

Given this example we argue that even if an original resource in the web is being removed we are often able to find a copy at a different location which makes it a "just-in-time" preservation approach. However, it is often a problem to retrieve the resource of interest. It, for example, is entirely possible that there are many more copies available in the web than shown in Figure **??**, but we just did not find them. We therefore propose to apply information retrieval techniques to rediscover web resources.



Google Scholar BETA Web Images Video News Maps more »

ml nelson ja kaplan Search

Advanced Scholar Search
Scholar Preferences
Scholar Help

Scholar All articles - Recent articles Results 1 - 20 of about 31,300 for ml nelson ja kaplan. (0.28 seconds)

[PDF] ▶A Comparison of Queueing, Cluster and Distributed Computing Systems
JA Kaplan, ML Nelson, Langley Research Center - 1993 - ntrs.nasa.gov
... Joseph A. Kaplan and Michael L. Nelson National Aeronautics and Space Administration ...
Joseph A. Kaplan (j .a. kaplan@larc, nasa. gov) Michael L. Nelson ...
Cited by 113 - Related articles - View as HTML - Web Search - Library Search - All 14 versions

A Comparison of Queueing, Cluster and Distributed Computing Systems (1994) (Make Corrections) (8 citations)
Joseph A. Kaplan, Michael L. Nelson

CiteSeer Home/Search Context Related

Links: ACM

View or download:
fsu.edu/pub/drago/dep/kaplan.ps.Z
kari.re.kr/NASA/larc/94...tm109025.ps.Z
nasa.gov/pub/techreport...tm109025.ps.Z
Cached: PS.gz PS PDF Image Update Help

From: fsu.edu (more)
From: kari.re.kr/ltrs/1994cit
(Enter author homepages)

(Enter summary)

Rate this article: 1 2 3 4 5 (best)
Comment on this article

Abstract: Using workstations clusters for distributed computing has become popular with

INTERNET ARCHIVE WaybackMachine

Enter Web Address: http:// All Take Me Back Adv. Search Compare Archive Pages

Searched for http://techreports.larc.nasa.gov/ltrs/PDF/tm109025.pdf 5 Results

* denotes when site was updated.
Material typically becomes available here 6 months after collection. See FAQ.

Search Results for Jan 01, 1996 - Feb 29, 2008

1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
0 pages	0 pages	1 pages	0 pages	1 pages	0 pages	0 pages	0 pages	0 pages	3 pages	0 pages	0 pages	0 pages
		May 03, 1998 *		Sep 25, 2000 *					Jan 23, 2005 * May 12, 2005 Oct 01, 2005 *			

Fig. 1 Refreshing and Migration Occurring in the Living Web

1 THESIS MOTIVATION

Related research has shown that web pages become inaccessible over time (shown for example in [147] and [166]) and consequently HTTP 404 "Page Not Found" errors are part of everyone's web browsing experience. However, as mentioned above, our intuition is that web pages do not disappear but often just move to a new location which then needs to be discovered. We propose the use of information retrieval techniques to address the problem of web page preservation.

Regarding the frequent change of URIs mapping to some content in the web, four general scenarios can be observed:

1. the same URI maps to the same or very similar content at a later time
2. the same URI maps to different content at a later time
3. a different URI maps to the same or very similar content at the same or at a later time
4. the content can not be found at any URI.

Table 1 Conferences and their Original URIs

Conference	Original URI
JCDL 2005	`http://www.jcdl2005.org/`
Hypertext 2006	`http://www.ht06.org/`
PSP 2003	`http://www.pspcentral.org/events/annual_meeting_2003.html`
ECDL 1999	`http://www-rocq.inria.fr/EuroDL99/`
Greynet 1999	`http://www.konbib.nl/infolev/greynet/2.5.htm`

Table 1 provides five examples of research conference URIs for each of these scenarios. The column next to the conference name displays the original URI as it was published at the time the conference was held. The different cases of the conferences, their URIs, and the validity of the URIs are as follows:

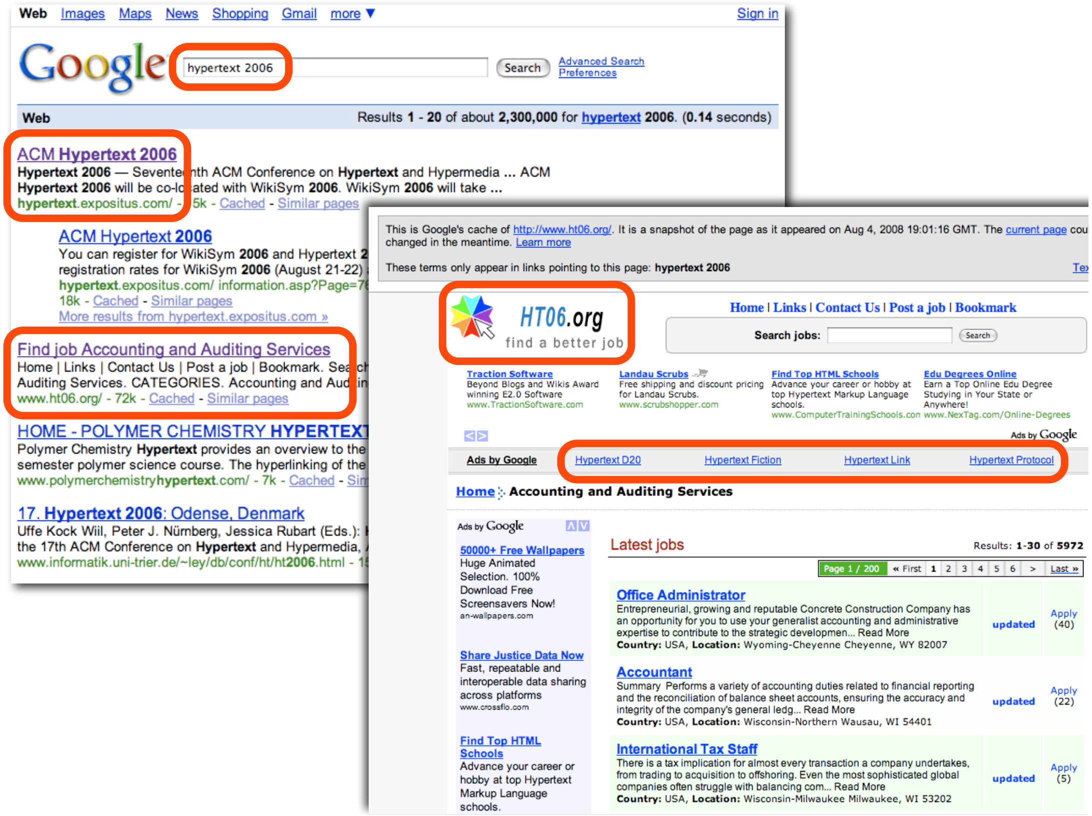
1. Joint Conference on Digital Libraries (JCDL) 2005, today this URI is still valid and accessible and also holds the same original content.
2. Hypertext 2006, the URI is still valid today but does not reference to the original content anymore. The snapshot in Figure 2 shows on the right the unrelated content that is available through the original URI today.
3. Professional Scholarly Publishing conference (PSP) 2003 and European conference on digital libraries (ECDL) 1999, both URIs return a 404 error today but suitable replacement pages are available.
   (a) PSP, the replacement page is hosted on the same server
   (b) ECDL, the page is hosted by a different domain but still summarizes the event
4. Greynet 1999, the URI returns a 404 error and a replacement page can not be found.

2 DESCRIBING THE MAPPING BETWEEN URIS AND CONTENT

The dynamic mapping between URIs and content can be formalized into a framework consisting of two functions. Let us consider a set of URIs to be $(u_1), (u_2), ...(u_n) \in U$ each with a timestamp t assigned and where U is the set of all URIs in the web. Let us further assume a set of content to be $(c_1), (c_2), ...(c_m) \in C$ also with a timestamp t attached and where C is the representations served when dereferencing the URI at time t, hence indexed and searchable content. The first function ($u2c$) shown in Equation 1 takes a URI and a timestamp t_x (for example to access and distinguish multiple cached copies) and maps this pair into content (c_a, t_x):

$$u2c(u_i, t_x) \rightarrow (c_a, t_x) \tag{1}$$

Note that content (c_a, t_x) has a timestamp assigned too.



Results 1 - 20 of about 2,300,000 for hypertext 2006. (0.14 seconds)

ACM Hypertext 2006
Hypertext 2006 — Sevente nth ACM Conference on Hypertext and Hypermedia ... ACM
Hypertext 2006 will be co-lo ated with WikiSym 2006. WikiSym 2006 will take ...
hypertext.expositus.com/ - 5k - Cached - Similar pages

ACM Hypertext 2006
You can register for WikiSym 2006 and Hypertext 2
registration rates for WikiSym 2006 (August 21-22)
hypertext.expositus.com/ information.asp?Page=76
18k - Cached - Similar pages
More results from hypertext.expositus.com »

Find job Accounting and Auditing Services
Home | Links | Contact Us | Post a job | Bookmark. Sea ch
Auditing Services. CATEGORIES. Accounting and Aud in
www.ht06.org/ - 72k - Cached - Similar pages

HOME - POLYMER CHEMISTRY HYPERTEXT
Polymer Chemistry Hypertext provides an overview to the
semester polymer science course. The hyperlinking of the
www.polymerchemistryhypertext.com/ - 7k - Cached - Sim

17. Hypertext 2006: Odense, Denmark
Uffe Kock Wiil, Peter J. Nürnberg, Jessica Rubart (Eds.): H
the 17th ACM Conference on Hypertext and Hypermedia, A
www.informatik.uni-trier.de/~ley/db/conf/ht/ht2006.html - 15

This is Google's cache of http://www.ht06.org/. It is a snapshot of the page as it appeared on Aug 4, 2008 19:01:16 GMT. The current page cou
changed in the meantime. Learn more

These terms only appear in links pointing to this page: hypertext 2006

Traction Software
Beyond Blogs and Wikis Award winning E2.0 Software
www.TractionSoftware.com

Landau Scrubs
Free shipping and discount pricing for Landau Scrubs.
www.scrubshopper.com

Find Top HTML Schools
Advance your career or hobby at top Hypertext Markup Language schools.
www.ComputerTrainingSchools.con

Edu Degrees Online
Earn a Top Online Edu Degree Studying in Your State or Anywhere!
www.NexTag.com/Online-Degrees

50000+ Free Wallpapers
Huge Animated Selection. 100% Download Free Screensavers Now!
an-wallpapers.com

Share Justice Data Now
Fast, repeatable and interoperable data sharing across platforms
www.crossflo.com

Find Top HTML Schools
Advance your career or hobby at top Hypertext Markup Language schools.

Latest jobs

Results: 1-30 of 5972

Office Administrator
Entrepreneurial, growing and reputable Concrete Construction Company has an opportunity for you to use your generalist accounting and administrative expertise to contribute to the strategic developmen... Read More
Country: USA, Location: Wyoming-Cheyenne Cheyenne, WY 82007
updated
Apply (40)

Accountant
Summary Performs a variety of accounting duties related to financial reporting and the reconciliation of balance sheet accounts, ensuring the accuracy and integrity of the company's general ledg... Read More
Country: USA, Location: Wisconsin-Northern Wausau, WI 54401
updated
Apply (22)

International Tax Staff
There is a tax implication for almost every transaction a company undertakes, from trading to acquisition to offshoring. Even the most sophisticated global companies often struggle with balancing com... Read More
Country: USA, Location: Wisconsin-Milwaukee Milwaukee, WI 53202
updated
Apply (5)

Fig. 2 The Official URI of the Hypertext 2006 Conference Provides Different Content Over Time

The second function ($c2u$) shown in Equation 2 maps content into URIs. Since the content can be held by multiple URIs at different times, the equation is displayed as a matrix:

$$c2u(c_a, t_x) \rightarrow \left\{ \begin{array}{llll} (u_i, t_x) & (u_i, t_{x+1}) & \ldots & (u_i, t_{x+k}) \\ (u_{i+1}, t_{x+a}) & (u_{i+1}, t_{x+b}) & \ldots & (u_{i+1}, t_{x+c}) \\ \ldots & & & \\ (u_j, t_{x-k}) & (u_j, t_{x+n}) & \ldots & (u_j, t_{x+m}) \end{array} \right\} \tag{2}$$

Turning our attention to the content, we define (c_a, t_x) as full content and (c'_a, t_x) as a subset of the content so that $(c'_a, t_x) \subseteq (c_a, t_x)$. (c'_a, t_x) is obtained by applying what we call a reduced representation function rr to (c_a, t_x) as shown in Equation 3.

$$(c'_a, t_x) = rr(c_a, t_x) \tag{3}$$

If we, for example, apply rr to two contents (c_a, t_x) and (c_b, t_x) and find that $rr(c_a, t_x) \approx rr(c_b, t_x)$ it is possible that $(c_a, t_x) \approx (c_b, t_x)$ or $(c_a, t_x) \not\approx (c_b, t_x)$. In order to further explain the notion of equality and similarity of content we state the following possible cases:

1. if $(c_a, t_x) = (c_b, t_x)$ then $rr(c_a, t_x) = rr(c_b, t_x)$

2. if $(c_a, t_x) \approx (c_b, t_x)$ then $rr(c_a, t_x) \approx rr(c_b, t_x)$, maybe even $rr(c_a, t_x) = rr(c_b, t_x)$

3. if $(c_a, t_x) \not\approx (c_b, t_x)$

   (a) $rr(c_a, t_x) \not\approx rr(c_b, t_x)$ is the expected scenario

   (b) $rr(c_a, t_x) \approx rr(c_b, t_x)$ then we need to further explore this problem. This scenario is comparable to hash collisions where dissimilar input yields similar output. Our problem in this case is that the $c2u$ function would return inaccurate values.

The following scenarios can occur for the function $u2c \rightarrow (c_a, t_x)$ where $\forall\ k > 0$ and $\forall\ l \geq 0$:

1. $u2c(u_i, t_x) = u2c(u_i, t_{x+k})$ - the same URI maps to the same or similar $(u2c(u_i, t_x) \approx u2c(u_i, t_{x+k}))$ content at a later time

2. $u2c(u_i, t_x) \not\approx u2c(u_i, t_{x+k})$ - the same URI maps to different content at a later time

3. $u2c(u_i, t_x) = u2c(u_j, t_{x+l})$ - a different URI maps to the same or similar $(u2c(u_i, t_x) \approx u2c(u_j, t_{x+l}))$ content at the same $(l = 0)$ or at a later time $(l \neq 0)$

4. $u2c(u_i, t_x)$ returns $(c_a, t_x) \notin C$ - content not found at any URI

Note that "similar" content, in contrast to "the same" content, is more forgiving of minor changes that do not effect the overall context and intention of the content of the web resource's representation. Such minor changes could be corrected typos in less significant terms, updated datestamps or frequently changing advertisements that occur on the page. There are numerous ways to measure similarity of textual documents such as simple term overlap, cosine similarity as well as the Jaccard and Dice coefficient which are explained in detail in Chapter II.

An example for scenario 1 is the JCDL 2005 URI, which at the time of the conference as well as today maps to the same content. The Hypertext 2006 URI is an example of scenario 2. It does not provide content about the conference anymore. In this case we are interested in finding a different URI that maps to the original content. Thus we need to solve the equation of scenario 3 in order to discover candidates for u_j. The second mapping function $c2u(c_a, t_x)$ is designed to provide such candidates. If its result set is not empty, meaning URIs have been returned, we can start the evaluation of u_j candidates. If however the result set is empty, we need to assume that $(c_a, t_x) \notin C$, which means the content is not discoverable. This process is also required for scenario 4 in order to discover possible values of u_j. The original URIs of the PSP, ECDL and Greynet conferences are examples for this scenario. All three scenarios imply that we have certain knowledge about the content (c_a, t_x). For example, if we do not know what "Hypertext 2006" is, how do we know which URI maps to the "right" content, or in other words, which (c_a, t_x) to use for $c2u$?

2.1 Possible Implementations

A possible and very likely implementation of the $u2c$ function is URI dereferencing. All items of Table 1 represent examples of URI dereferencing. Probably the most intuitive implementation of the $c2u$ function is a search engine query where a user "generates" the content in the form of query terms and retrieves URIs in return. Other possible implementations where the URI is used as the input to *c2u* are a search engine query for the in- and outlinks of a web page in order to analyze the link neighborhood. Queries to search engine caches as well as to web archives for copies of a page are also feasible. These implementations become especially useful when the URI dereferencing fails.

Further options are based on web browsing experience and domain knowledge. It is for example known that web sites of colleges in the US belong to the *.edu* domain and the computer science department usually is abbreviated with *cs*. An experienced user could for example anticipate the URI of the computer science department at Cornell University `http://www.cs.cornell.edu/`.

The earlier introduced reduced representation function *rr* can be implemented in various ways. If it exists, a web page's title is an intuitive candidate for the *rr* function since it is understood to summarize the document's content. A lexical signature, described in detail in Chapter II, Section 6, consisting of a few terms only is another option to transform textual content into a reduced representation. Tags given by Internet users can also refer to a document's content. The web service *Delicious* [7] for example offers the capability to share bookmarks and their annotations with the Internet community. In a hyperlinked environment such as the World Wide Web, the *rr* function can also be implemented with the help of the link neighborhood of a centroid URI. The underlying assumption is that content-wise related pages share links.

Hash functions can provide another option for a special reduced representation of the content. However, we prefer the previous four implementations of the transformation function because they allow us to implement the function *c2u* as search engine queries which would not be possible with hash values since search engines do not handle such input. It is important to notice that the framework presented here is independent of the implementation of the *u2c*, *c2u* and *rr* functions.

2.2 Sample Results of the Mapping Functions

Table 2 shows alternative URIs for the five conferences discovered using the *c2u* function. For the Hypertext conference we were able to discover that the site has moved to `http://hypertext.expositus.com/`. *c2u* (implemented as a search engine query) revealed that the original URI interestingly is still indexed by Google as shown in the left part of Figure 2. *c2u* also returned the alternative URI for the PSP and ECDL conferences. The alternative URI for PSP holds the same original content and the URI is hosted by the same domain, just the path to the HTML file has changed. The returned URI for ECDL is a conference overview page provided by the DBLP computer science bibliography service. This page provides meta information such as the conference location and references to all accepted papers of the conference. Note the two differences here:

1. the site is hosted by a different entity, even in a different top-level domain, and
2. the content of the discovered page is different compared to the original but probably still sufficient for the user seeking information about the conference.

In this case we are able to recover parts of the original content of the page but other information such as the conference schedule and the look and feel of the page may be lost.

The last example has a less positive outcome. No alternative URI for Greynet (a conference on the preservation of so called grey literature) was returned by *c2u* and thus, at this point, we need to consider the content of the Greynet conference as lost.

Table 2 Conferences and Their Alternative URIs

Conference	**Alternative URI**
JCDL 2005	`http://www.jcdl2005.org/`
Hypertext 2006	`http://www.hypertext.expositus.com/`
PSP 2003	`http://www.pspcentral.org/events/archive/annual_meeting_2003.html`
ECDL 1999	`http://www.informatik.uni-trier.de/~ley/db/conf/ercimdl/ercimdl99.html`
Greynet 1999	???

3 THESIS STATEMENT

The goal of this dissertation is to address the detriment to the web browsing experience caused by 404 responses and provide alternatives to missing pages in *real time.* To achieve this goal we propose the use of the following four implementations of the rr function:

1. lexical signatures
2. web page's titles
3. tags, used to annotate the page
4. link neighborhood based lexical signatures.

The idea of using lexical signatures for this purpose is not entirely new. Phelps and Wilensky [218] first introduced the term *lexical signature* and proposed their use to discover web pages that had been moved. Their preliminary tests confirmed that 5-term lexical signatures are suitable for discovering a page when used as search engine queries. Although Phelps and Wilensky's early results were promising, there were two significant limitations that prevented lexical signatures from being widely deployed.

1. Their scenario required the browser's source code to be modified to exploit lexical signatures and
2. they required lexical signatures to be computed a priori.

Park et al. [214] expanded on the idea of Phelps and Wilensky and studied the performance of nine different lexical signature generation algorithms (retaining the 5-term precedent). They proved that slight modifications in the generation process can improve the retrieval performance of relevant web pages.

Web pages' titles (Hyusein and Patel [130]) as well as tags (Bischoff et al. [71]) have been shown to be suitable for web search and the link neighborhood has been shown to refine the lexical signatures of centroid pages (Sugiyama et al. [247]). However, none of these three methods has been studied in context of (re-)discovering missing web pages. By building upon the idea of Phelps and Wilensky and leveraging the feasibility study of Park et al. we propose using all four implementations of the rr function to (re-)discover missing web pages based on the Web Infrastructure (WI) which is described in detail in Chapter II, Section 2.

The open source community around the Mozilla project [34] is growing and we see more and more browser extensions being implemented. Maybe even more importantly, we are able to implement the *rr* function in *real time* and for example do not need to compute lexical signatures a priori. We therefore overcome the limitations that prevented Phelps and Wilensky's system from being widely deployed.

We propose *Synchronicity*, a Mozilla Firefox add-on that catches 404 errors when they occur, meaning while the user is browsing the Internet. The system utilizes the *Memento* framework (see Chapter II, Section 4) to obtain cached and archived versions of the page. Besides offering these older versions to the user, the add-on further obtains the title and generates a lexical signature based on the previous versions of the page. The add-on also obtains tags as well as generates link neighborhood based lexical signatures of the missing page. Any of these methods represent implementations of the *rr* function and are meant to offer alternative pages to the user (with respect to the *c*2*u* mapping) while she is browsing the web.

Synchronicity implements the scenario of several components (which may be causally unrelated) working together and returning meaningful results. The components are web pages gone missing, the natural preservation of digital data in the WI and the application of diverse Information Retrieval (IR) methods. The meaningful results, of course, are (re-)discovered relevant web pages, again implemented with the *c*2*u* function.

In related research, McCown [189] has shown that the WI can be used to reconstruct missing websites. He developed a system called *Warrick* which crawls web repositories such as search engine caches and the IA for copies of the website. Warrick merges the findings (if necessary) and therefore reconstructs the entire missing website. Synchronicity is different compared to Warrick in a number of aspects. It locates the missing page or sufficient replacement pages in real time, it uses information retrieval techniques (like lexical signatures) to (re-)discover the pages and, unlike Warrick, which reconstructs entire sites using the URI as the seed for the WI, Synchronicity recovers single web pages.

The last point also determines that the target group is different. Where Warrick is aimed primarily towards website administrators and content generators who accidentally lost their websites, Synchronicity is geared towards end users, browsing the web and experiencing HTTP 404 errors.

The work done by Harrison for his master thesis [119] and published in Harrison and Nelson [120] can be considered as an early version of Synchronicity. His software system called *Opal* also feeds lexical signatures into the WI to find missing web pages. The main difference is that Opal is a server side system and therefore requires system administrators to install and maintain the software. We take a different path since our system is purely client sided and, due to the information retrieval components, more sophisticated.

Lastly, we create the "Book of the Dead", a corpus of missing web pages. To the best of our knowledge no such corpus is available thus far. Most related preservation and information retrieval research has been done on corpora of resources that are in fact not missing. Often experiments are conducted on synthetic web sites or pages or based on bounded datasets usually created for a particular research goal. Such corpora are described in detail in Chapter III, Section 8. We apply our methods to our new corpus and evaluate their outcome. We also offer the corpus to fellow researchers to enable and encourage continuative research in this area.

3.1 Organization

The dissertation is organized into the following chapters, separated by topic and contribution.

Chapter II: Background – We introduce basic terminology as well as components and protocols of the Internet used in the dissertation. We explain in detail information retrieval concepts and measures to evaluate the performance of retrieval systems and provide examples of their use.

Chapter III: Research Review: Digital Preservation for the Web – We analyze related research that motivates, supports and compliments our work. We show current circumstances with respect to missing web pages and give examples of common server responses. We review related work on the use of search engines and all of our methods mentioned above.

Chapter IV: TF-IDF Values for the Web – This chapter covers our work on finding a well performing estimation method for IDF values for textual web page content. It further contains the correlation study between term count and document frequency scores.

Chapter V: Lexical Signatures for the Web – Here we analyze the evolution of lexical signatures over time and their performance in dependence of their age. We develop a framework to automatically generate well performing lexical signatures.

Chapter VI: Titles – Our next retrieval method is evaluated in this part of the dissertation. We analyze its evolution over time and compare to the change of document content. We identify "stop titles", titles with no semantic value.

Chapter VII: Tags – In this chapter, we look at the retrieval performance of tags and introduce the notion of "ghost tags", tags that no longer describe a URI's content.

Chapter VIII: Link Neighborhood Lexical Signatures – We evaluate all parameters of link neighborhood based lexical signatures and provide a set of parameters that warrant the generation of well performing search engine queries.

Chapter IX: Book of the Dead – We introduce the corpus of missing web pages and provide the contextual "aboutness" of every single URI in the corpus. We apply all retrieval methods and report the results.

Chapter X: Synchronicity – In this chapter, we present our software implementation. We cover its functionality and show screen shots of its operation.

Chapter XI: Future Work and Conclusions – Here we give pointers to aspects for future work, summarize our findings and list all contributions of this work.

CHAPTER II

BACKGROUND

1 INTRODUCTION

In this chapter, we introduce components and protocols of the Internet that are used in the dissertation. For example, we explain in detail the web infrastructure, since it motivates the entire concept of this work. We further introduce the reader to the Memento framework, as it is another foundation of this work.

We demonstrate the underlying algorithms of the term frequency-inverse document frequency weighting scheme as well as the mathematical background for lexical signatures. This is meant to review these concepts and their success in IR research.

Since we are utilizing retrieval systems in this work, we explain some methods to evaluate their performance. String similarity methods are covered as well as rank correlation measures. We use these methods in the following chapters and hence a basic understanding is beneficial. With the help of several examples we hope to vividly introduce the concepts to the reader.

2 WEB INFRASTRUCTURE

The WI consists of Internet search engines such as Google, Yahoo! and Microsoft Bing and their caches. It also includes non-profit archives such as the Internet Archive or the European Archive as well as large-scale academic digital data preservation projects e.g., CiteSeer and NSDL. The WI is explored in detail in Frank McCown's dissertation thesis [189] and other related work [139].

McCown has done extensive research on the usability of the WI for preservation. He coined the phrase "lazy preservation" [196] as the utilization of the WI to reconstruct missing websites. He developed *Warrick*, a system that crawls web repositories such as search engine caches (characterized in [193]) and web archives to reconstruct websites. His system is targeted to individuals and small scale communities that are not involved in large scale preservation projects and suffer the loss of websites. This approach requires almost no effort from web administrators, hence its label "lazy preservation".

McCown et al. [191] also investigated what factors contribute to the reconstruction of websites based on the WI. Their findings indicate that the PageRank value (computed by Google) as well as the age of the missing page affect the success of the reconstruction greatly. The distance in number of hops of a missing page from the root page was a factor as well, since most web crawlers have a depth limit when crawling a website.

McCown and Nelson [195] also formally described web repositories in the WI. They distinguish between *flat* and *deep* repositories. A repository is considered flat if it only keeps one, the most recent, version of a resource as search engine caches do, for example. Deep repositories (such as the Internet Archive), on the other hand, store multiple versions of a resources distinguished by timestamp. In addition they refer to repositories that do not allow public access as *dark* repositories and to others that allow access as *light* repositories. As a logical consequence, they also define *gray* repositories as those that are partially accessible.

Another approach to preservation of web pages based on the web infrastructure is called "shared infrastructure preservation" [241]. The idea is to push resources (including metadata) via NNTP newsgroups and SMTP email attachments to other sites that may archive (refresh, migrate) the data. Smith further introduces an Apache module called *mod_oai* which provides OAI-PMH access to MPEG-21 DIDL representations of web resources [240]. She calls this approach "web server enhanced preservation". Nelson et al. [209] summarize these example implementations for alternative approaches to web scale preservation. They argue that conventional approaches to digital preservation, such as storing digital data in archives and applying methods of refreshing and migration, are, due to the implied costs, unsuitable for web scale preservation.

3 SEARCH ENGINES AND THEIR APIS

All three major search engines (Google, Yahoo! and Bing) offer APIs for the public to access their services. These APIs allow users to write scripts that retrieve search results from the search engine's index. Data such as the ranking of the results, their URIs, their snippets, as well as their titles as displayed through the web interface are available. Additionally the (estimated) total number of results are available. The user can further specify the range of the desired results by indicating in the query string the start value for the ranking and the number of results to obtain. Regular search operators such as *AND*, *LINK* or *SITE* are also possible, but the exact syntax may vary between search engines. In general, the user can chose to use the SOAP [40], JSON [21] or XML [41] format to retrieve data from the APIs.

Early on in this dissertation work, Google had a general web search API in service which allowed $1,000$ queries per day. This API was used for the experiments presented in Chapters IV and V. As of November 1st 2010, Google has officially deprecated its general web search API and now refers to using the Custom Search API [14] instead. The new API restricts users to a 100 query per day quota, but the user can pay 5 USD per $1,000$ queries for up to $10,000$ queries per day.

Yahoo! introduced their BOSS Search API [45] in 2008 and promised unlimited queries to their index. All of our other experiments are therefore based on this API. However, Yahoo! commercialized this service as well and now offers $1,000$ queries against the entire index for 0.80 USD or against a "limited" index with a slower refresh rate for 0.40 USD [22]. The initially introduced free service will discontinue soon after the new model is launched.

Bing established a new version of their API [3] recently. Since the previous one did not perform as well as the Yahoo! BOSS API and came with the same quota limits as the Google API, we did not use Bing extensively in this work.

An example query for the term *wimbledon* against the Yahoo! BOSS API could look like this:

```
http://boss.yahooapis.com/ysearch/web/v1/wimbledon/?appid=my_api_key&format=xml
```

Of course *my_api_key* would have to be replaced with a valid key. In this example, we are requesting an XML formatted response which could look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<ysearchresponse xmlns="http://www.inktomi.com/" responsecode="200">
  <nextpage><![CDATA[/ysearch/web/v1/wimbledon/?format=xml&count=10&appid=-my_api_key&start=10]]></nextpage>
  <resultset_web count="10" start="0" totalhits="6020511" deephits="40400000">
    <result>
      <abstract><![CDATA[The official site of the All England Lawn Tennis Club, home to The Championships
<b>Wimbledon</b>, featuring news, features, spectator information, museum guide and online shop]]></abstract>
      <clickurl>http://lrd.yahooapis.com/ ... http%3A//www.wimbledon.com/</clickurl>
      <date>2011/05/30</date>
      <dispurl><![CDATA[www.<b>wimbledon.com</b>]]></dispurl>
      <size>18049</size>
      <title><![CDATA[<b>Wimbledon</b> - The Home of Tennis]]></title>
      <url>http://www.wimbledon.com/</url>
    </result>
    <result>
      <abstract><![CDATA[The 2010 <b>Wimbledon</b> Championships took place on the outdoor grass courts
at the All England Lawn Tennis and Croquet Club in <b>Wimbledon</b>, London, United Kingdom. <b>...</b>]]>
</abstract>
      <clickurl>http://lrd.yahooapis.com/ ... http%3A//en.wikipedia.org/wiki/2010_Wimbledon_Championships</clickurl>
      <date>2011/05/10</date>
      <dispurl><![CDATA[<b>en.wikipedia.org</b>/wiki/<wbr>2010_<b>Wimbledon</b>_Championships]]></dispurl>
      <size>596495</size>
      <title><![CDATA[2010 <b>Wimbledon</b> Championships - Wikipedia, the free encyclopedia]]></title>
      <url>http://en.wikipedia.org/wiki/2010_Wimbledon_Championships</url>
    </result>
```

We replaced some cryptic, potentially query specific data between the *¡clickurl¿* tags in order to improve the readability of the otherwise obvious response. We retrieve the estimated total number of results as totalhits (no duplicates) and deephits (with duplicates). Each result is embedded between *¡result¿* tags which further include the abstract (snippet) of the result, its date (presumably last visited date), the URI for a web interface user to click on, the displayed URI, size and title of the result page and again the URI.

Since search engines and their APIs are essential to web related research, they have been subject to performance evaluations in the past. McCown and Nelson [192], for example, compared search results from Google, Yahoo! and (at that time) MSN obtained from their APIs and web user interfaces and found significant discrepancies. Their results suggest that the index the Google and Yahoo! APIs are drawing from are not older but seem to be smaller than the index for the web interface. MSN appeared to have the most synchronized interfaces overall.

McCown and Nelson [194] further investigated the persistence of search results. They found that it can take up to one year for half of the top 10 results to be replaced in Google and Yahoo! for popular queries. For MSN, however, it may only take two to three months.

Search engines mostly also provide cached versions of indexed pages. The caching strategies behind this service were investigated by McCown and Nelson [193] in order to see whether one could build a reliable preservation service on top of the caches. They analyzed the Google, Yahoo!, MSN and Ask caches and found that Ask had the fewest resources cached. The rates for the other three were much better, at or above 80%. MSN showed the least stale cached resources (12%) and Google the most, with 20%.

With the aggregated work of McCown and Nelson, we know more about the otherwise entirely non-transparent “black box” search engines and their APIs.

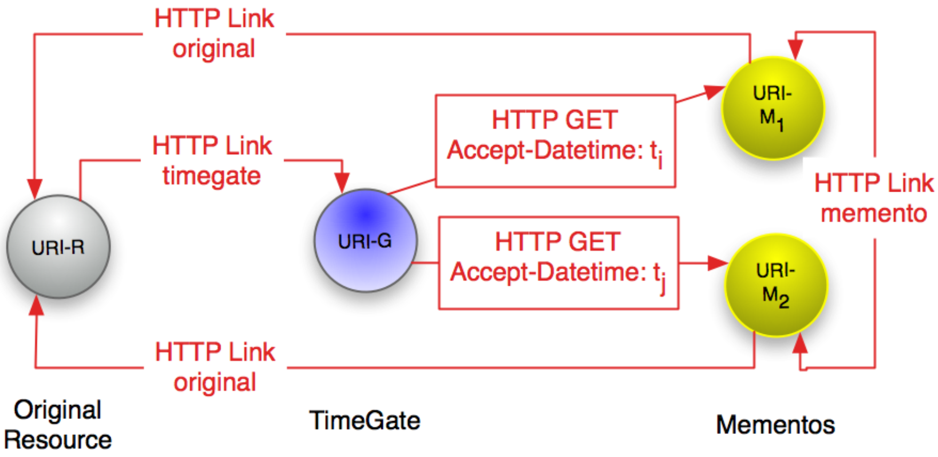


Fig. 3 The Memento Architecture

4 MEMENTO AND WEB ARCHIVES

The web architecture [2] does not define the dimension of time. It acknowledges that creating a new URI for every single change of state for any given resource would lead to a "significant number of broken references". It therefore promotes independence between the state of a resource and its identifier (URI). This means that only the current representation of a resource is available and older representations are lost.

This is where web archives come into the picture. The Internet Archive (IA) [20], for example, holds copies of websites beginning in 1996 to the present [147]. It implements, unlike commercial search engines, a best effort approach for web page preservation. That means its crawlers generally traverse the web and archive individual web pages at irregular intervals with no specific policy as to what pages to crawl at what times. The IA also participates in focused crawling based on policies and topics such as the Olympic Games and other international events. After crawling a site, the IA will not publish it in their index for another six to twelve months because of an applied "quarantine period". As of 2010, the IA holds more than 1.5 billion unique URIs [207], which makes it the largest web archive to date. We also see several international libraries and institutions focused on archiving particular portions of the web, mainly distinguished by country code of the domain [46, 87, 173, 223].

4.1 Memento Basics

Memento is a framework for time-based access of web resources [259, 260]. It utilizes the WI to provide previous representations of web resources. Memento uses search engine caches and various web archives, with the Internet Archive probably being the most reputable contributor. Each previous resource is called a *Memento* [24]. The framework further includes *TimeGates*, web resources that redirect to the proper Memento depending on the client-specified datetime. Figure **??**[1] displays the general Memento architecture. The web resource that we want to retrieve Mementos of is identified

[1] This figure is taken from `http://www.mementoweb.org`

as URI-R, the TimeGate as URI-G and the single Mementos (in this example M_1 and M_2) are identified as URI-M_1 and URI-M_2.

The Memento framework is based on transparent content negotiation [128] in HTTP. In order for a web client to retrieve a Memento of URI-R it sends an HTTP request to the original resource including an *Accept-Datetime* HTTP header. The resource responds with an HTTP *Link* header pointing to the pre-defined TimeGate. The TimeGate performs the content negotiation over the time dimension with the client-given datetime and responds, providing an HTTP 302 status code meaning "Found" and the *Location* header pointing to the matching Memento, for example URI-M_1. The client will then send another GET request to retrieve the Memento from the specified URI. The TimeGate and each Memento can also respond with a Link header pointing to the *first* and *last* known Memento of URI-R. Note that a URI's first Memento can at the same time be its last Memento. The individual Mementos can further point to the *next* and *previous* Memento again using the HTTP Link header. If a client specifies a datetime for a Memento that is out of its range, meaning before the first or after the last Memento, the TimeGate also responds with the HTTP 302 status code and includes the Location header pointing to the first or most recent Memento, respectively. If an archive has more than one Memento with the same datetime, the TimeGate selects one of the matching Mementos and sends a 302 response code plus the proper Location header.

It is possible that Mementos of any given resource are distributed over numerous archives. Rather than the client querying every single archive separately, the Memento framework offers an aggregator to address this problem. A resource that provides an overview of all available Mementos is called a *TimeBundle*. The TimeBundle is a conceptual resource, which means it does not have a representation. It is a resource that aggregates other resources such as the original resource, its TimeGate and its Mementos. The resource called *TimeMap* describes the TimeBundle, as it is a machine readable document from which the URIs of all available Mementos can be obtained. A TimeBundle redirects to the TimeMap with a HTTP 303 response. Figure 4[2] depicts the underlying concept of TimeBundles and TimeMaps in the Memento framework. The Memento framework currently has the state of an IETF Internet draft [258].

4.2 Uses of Memento

Despite its early stage Memento has already stimulated research in global web archiving. For example, Ainsworth et al. [55] approach the question of how much of the web is archived. They utilize Memento to check for archived versions of a page and find that the IA and search engines have a similar URI coverage with respect to their sample sets. They built four sets of randomly sampled URIs from the Open Directory Project (DMOZ) [27], Delicious, the URI shortening service Bitly [4] and search engine indexes. Their results show that 35-90% of the URIs have at least one Memento, 17-49% have two to five Mementos, only 1-8% have six to ten Mementos and 8-63% of URIs have more than ten Mementos.

Sanderson and Van de Sompel [236] utilize the Memento framework and introduce a technique

[2] This figure is taken from `http://www.mementoweb.org`

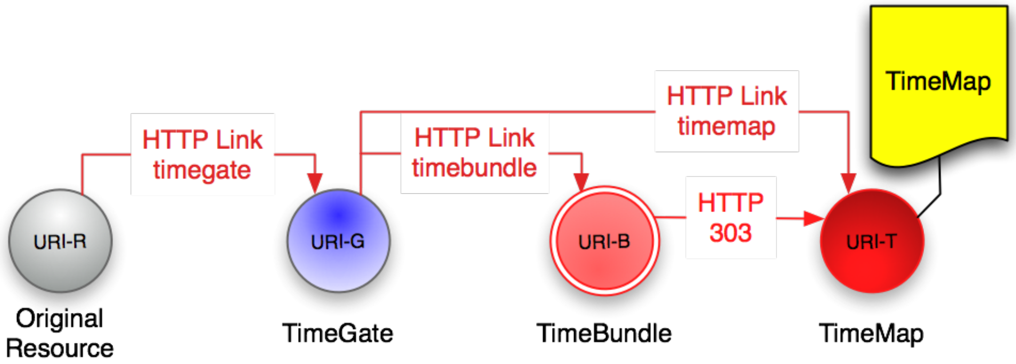


Fig. 4 Memento Timebundle Concept

that bridges the gap between an annotation about a web resource and its ever changing representation. The problem is that an annotation about a resource made today may be invalid tomorrow since the resource has changed. They build upon the Open Annotation Collaboration (OAC) [235] techniques and the availability of archived versions of resources through Memento. This combination ensures persistence of web annotations over time.

In later work, Sanderson et al. [233] also utilize OAC techniques to implement a system facilitating the interoperability of repositories. In this case they address the need for collaboration while researching digital copies of medieval manuscripts. This, for example, includes the sharing and modifying of annotations made about the resources. A manuscript (or a part of it) is visualized as a canvas which means that there is a need to aggregate multiple canvases to display the entire manuscript. Sanderson et al. utilize OAI-ORE aggregations (Lagoze et al. [171, 172]) for this requirement. ORE aggregations are sets of resources that can also include metadata about the resources. It therefore is a good ontology for a sequence of canvases.

5 TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY

Term frequency (TF) is the number of occurrences of a term within a particular document. It therefore represents the importance of a term for that particular page. Inverse document frequency (IDF), on the other hand, is defined as the number of documents a term occurs within the entire corpus of documents. It hence provides an indicator of the rareness of that term in the document's corpus. Karen Spärck Jones [144] introduced in 1972 the notion of term specificity, which later became known as IDF. The intuition was that for document retrieval a term occurring in many documents in the corpus is not a good discriminator and hence should get a lower score compared to a term only occurring in a few documents. Coupled as TF-IDF, the score provides a very accurate measure of terms' local (within the document) and global (within the entire corpus) importance.

The computation of TF values is not difficult, simply counting the occurrences of a term within a document usually suffices. Document length normalization can be applied to avoid a bias towards longer documents. A longer document possibly uses the same term more often than a shorter document and is therefore bound to show an increased relevancy towards a query containing this term. Also longer documents tend to contain more unique terms, which increases the number of matches between that document and a query. The most common approach for TF normalization is to normalize each TF value individually with the maximum TF value of the document [62, 110]. Equation 4 shows how to normalize using maximum TF values.

$$TF_{norm} = a + (1-a)\frac{TF}{TF_{max}} \tag{4}$$

TF_{norm} is the normalized TF value, TF is actual TF value of a term and TF_{max} is the maximum TF value in that document. This equation includes a smoothing factor a which is generally set to 0.4 as stated in [182]. Cosine normalization [231] is a common alternate normalization scheme. It is computed with Equation 5

$$TF_{norm} = \frac{TF}{\sqrt{{w_1}^2 + {w_2}^2 + ... + {w_t}^2}} \tag{5}$$

where w_i denotes the basic TF-IDF score of term i. Cosine normalization tends to favor short documents in retrieval. This intuitively makes sense since a higher normalization factor results in lower retrieval chances for a document. In fact, the retrieval probability of a document is inversely related to the normalization factor used for term weighting in that document. To balance that tendency we can apply a pivoted normalization scheme introduced by Singhal et al. [239] to, for example, cosine normalization. The pivoted scheme is based on predicted relevancy in dependence of document length. It lowers the value of the normalization factor for shorter documents and increases the value for larger documents.

In practice we can also find other approaches to normalization. Simply using the logarithm of the raw TF value (Equation 6) or the document's byte length as applied in the Okapi system [226] has been shown to be feasible.

$$TF_{norm} = 1 + log(TF) \tag{6}$$

Okapi TF [225] normalizes for document length normalization and also accounts for term frequency saturation. It is, for example, used in the BM25 ranking function [226]. One possible equation for Okapi TF is shown in Equation 7

$$TF_{d_i,D} = \frac{(k_1+1) \times tf(d_i,D)}{k_1 \times (1 - b + b \times \frac{|D|}{|D_{avg}|}) + tf(d_i,D)} \tag{7}$$

where $tf(d_i, D)$ is the number of times that term d_i occurs in D, $|D|$ is the total number of terms and $|D_{avg}|$ is the average document length in the entire corpus. It further includes two parameters k_1 and b that can be freely tuned.

The computation of IDF values is done with Equation 8, where D denotes the total number of documents in the entire corpus and d_i is the number of documents in D that contain term i.

$$IDF_i = log\frac{|D|}{|d_i|} \tag{8}$$

Since it is possible that a term does not occur in any documents, which would lead to the division by zero, the denominator is frequently computed as $|d_i|+1$. Obviously the computation of IDF depends on global knowledge about the corpus, namely $|D|$ and $|d_i|$. If, as given in our case, the entire web is the corpus, these values cannot be computed accurately. No single entity (search engines, archives, etc.) possesses a copy of the entire web. Hence, the mandatory values to compute IDF scores have to be estimated. To compute TF-IDF values for any given term, its TF and IDF values are multiplied.

The significance of IDF has been studied in the past, for example by Karen Spärck Jones [145] and Papineni [212]. They found that IDF is optimal for document self-retrieval which means, given an information retrieval environment, IDF is the perfect weight associated with a term to have the document (that contains the term) retrieve itself. In other words, this means that IDF is believed to have the ability to successfully discriminate against other documents. Intuitively, one would consider low-DF terms to be most desirable for discriminating between documents especially in combination with a high TF value. While that holds true for TF-IDF, it has been shown that for text classification problems a high dimensional feature space is undesirable. This means, isolated from TF, using DF as a feature space applying a threshold for cutting off low-DF terms is beneficial for the performance in text categorization [271].

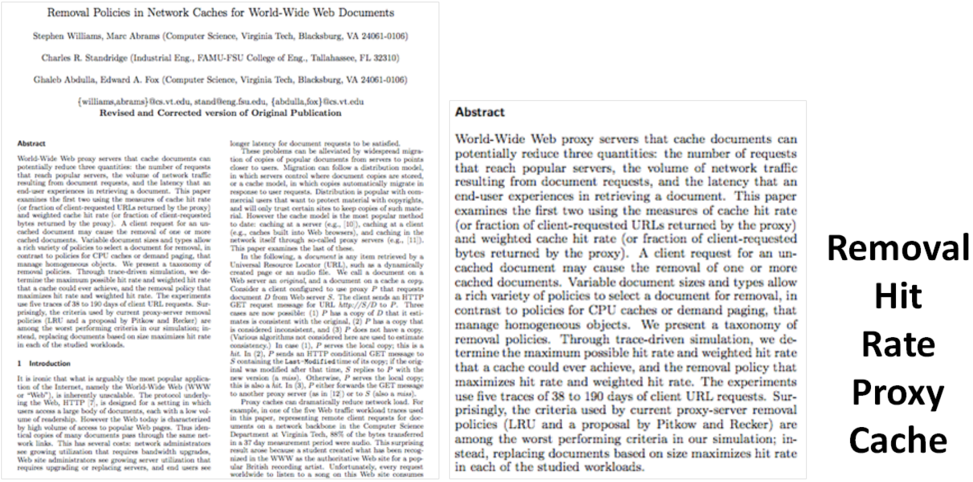
One of the main disadvantages of TF-IDF is that it considers unigrams only, meaning that, for example, it treats "New York" as two terms and results in two TF-IDF values, even though (depending on the context) it should probably be considered as one phrase resulting in one score. This can lead to a lower performance, and therefore an approach called *soft TF-IDF* has been introduced [88]. The basic idea is to apply string similarity measures (some of which are introduced in Section 7.3) to do a more "fuzzy" matching on the terms when obtaining DF values. This should result in the same (higher) score for similar terms. Soft-TFIDF has, for example, been applied for gene name normalization [105].

6 LEXICAL SIGNATURES

A lexical signature is a small set of terms derived from a document that captures the "aboutness" of that document. It can be thought of as an extremely lightweight metadata description of a document, as it ideally represents the most significant terms of its textual content.

Lexical signatures are usually generated following the TF-IDF weighting scheme, which means all terms of the document are ranked in decreasing order of their TF-IDF score. The top n terms from that list form the lexical signature. Usually so called stop words, words frequently occurring in a language but not contributing to the lexical character of a document [109, 230] are dismissed prior to the TF-IDF calculation. Stop words usually have a large DF- and hence a low IDF value which means their TF-IDF value will be low too. They will consequently not be highly ranked in a list of terms ordered by their decreasing TF-IDF values and therefore not make it into the lexical signature. However, removing stop words in advance reduces the complexity of the overall lexical signature computation.

A set of stop words is also called a negative dictionary since the terms cannot be chosen as index terms for retrieval systems. Wilbur and Sirotkin [266] introduced a method to automatically identify stop words. Their claim is that stop words have the same probability to occur in both documents not relevant to a given query and documents relevant to the query. It is further not uncommon to shorten terms to their stem in order to avoid quasi duplicates due to trivial word variations. Probably the most famous and commonly applied stemming algorithm is the Porter stemmer, first introduced by Porter [220].



Removal Policies in Network Caches for World-Wide Web Documents

Stephen Williams, Marc Abrams (Computer Science, Virginia Tech, Blacksburg, VA 24061-0106)

Charles R. Standridge (Industrial Eng., FAMU-FSU College of Eng., Tallahassee, FL 32310)

Ghaleb Abdulla, Edward A. Fox (Computer Science, Virginia Tech, Blacksburg, VA 24061-0106)

{williams,abrams}@cs.vt.edu, stand@eng.fsu.edu, {abdulla,fox}@cs.vt.edu
Revised and Corrected version of Original Publication

Abstract

World-Wide Web proxy servers that cache documents can potentially reduce three quantities: the number of requests that reach popular servers, the volume of network traffic resulting from document requests, and the latency that an end-user experiences in retrieving a document. This paper examines the first two using the measures of cache hit rate (or fraction of client-requested URLs returned by the proxy) and weighted cache hit rate (or fraction of client-requested bytes returned by the proxy). A client request for an uncached document may cause the removal of one or more cached documents. Variable document sizes and types allow a rich variety of policies to select a document for removal, in contrast to policies for CPU caches or demand paging, that manage homogeneous objects. We present a taxonomy of removal policies. Through trace-driven simulation, we determine the maximum possible hit rate and weighted hit rate that a cache could ever achieve, and the removal policy that maximizes hit rate and weighted hit rate. The experiments use five traces of 38 to 190 days of client URL requests. Surprisingly, the criteria used by current proxy-server removal policies (LRU and a proposal by Pitkow and Recker) are among the worst performing criteria in our simulation; instead, replacing documents based on size maximizes hit rate in each of the studied workloads.

1 Introduction

It is ironic that what is arguably the most popular application of the Internet, namely the World-Wide Web (WWW or "Web"), is inherently unscalable. The protocol underlying the Web, HTTP [7], is designed for a setting in which users access a large body of documents, each with a low volume of readership. However the Web today is characterized by high volume of access to popular Web pages. Thus identical copies of many documents pass through the same network links. This has several costs: network administrators see growing utilization that requires bandwidth upgrades, Web site administrators see growing server utilization that requires upgrading or replacing servers, and end users see longer latency for document requests to be satisfied.

These problems can be alleviated by widespread migration of copies of popular documents from servers to points closer to users. Migration can follow a distribution model, in which servers control where document copies are stored, or a cache model, in which copies automatically migrate in response to user requests. Distribution is popular with commercial users that want to protect material with copyrights, and will only trust certain sites to keep copies of such material. However the cache model is the most popular method to date: caching at a server (e.g., [10]), caching at a client (e.g., caches built into Web browsers), and caching in the network itself through so-called proxy servers (e.g., [11]). This paper examines the last of these.

In the following, a *document* is any item retrieved by a Universal Resource Locator (URL), such as a dynamically created page or an audio file. We call a document on a Web server an *original*, and a document on a cache a *copy*. Consider a client configured to use proxy P that requests document D from Web server S. The client sends an HTTP GET request message for URL *http://S/D* to P. Three cases are now possible: (1) P has a copy of D that it estimates is consistent with the original, (2) P has a copy that is considered inconsistent, and (3) P does not have a copy. (Various algorithms not considered here are used to estimate consistency.) In case (1), P serves the local copy; this is a *hit*. In (2), P sends an HTTP conditional GET message to S containing the `Last-Modified` time of its copy; if the original was modified after that time, S replies to P with the new version (a *miss*). Otherwise, P serves the local copy; this is also a *hit*. In (3), P either forwards the GET message to another proxy server (as in [12]) or to S (also a *miss*).

Proxy caches can dramatically reduce network load. For example, in one of the five Web traffic workload traces used in this paper, representing remote client requests for documents on a network backbone in the Computer Science Department at Virginia Tech, 88% of the bytes transferred in a 37 day measurement period were audio. This surprising result arose because a student created what has been recognized in the WWW as the authoritative Web site for a popular British recording artist. Unfortunately, every request worldwide to listen to a song on this Web site consumes

(a) Paper's First Page

Abstract

World-Wide Web proxy servers that cache documents can potentially reduce three quantities: the number of requests that reach popular servers, the volume of network traffic resulting from document requests, and the latency that an end-user experiences in retrieving a document. This paper examines the first two using the measures of cache hit rate (or fraction of client-requested URLs returned by the proxy) and weighted cache hit rate (or fraction of client-requested bytes returned by the proxy). A client request for an uncached document may cause the removal of one or more cached documents. Variable document sizes and types allow a rich variety of policies to select a document for removal, in contrast to policies for CPU caches or demand paging, that manage homogeneous objects. We present a taxonomy of removal policies. Through trace-driven simulation, we determine the maximum possible hit rate and weighted hit rate that a cache could ever achieve, and the removal policy that maximizes hit rate and weighted hit rate. The experiments use five traces of 38 to 190 days of client URL requests. Surprisingly, the criteria used by current proxy-server removal policies (LRU and a proposal by Pitkow and Recker) are among the worst performing criteria in our simulation; instead, replacing documents based on size maximizes hit rate in each of the studied workloads.

(b) Paper's Abstract

Removal
Hit
Rate
Proxy
Cache

(c) Paper's Lexical Signature

Fig. 5 Lexical Signature Example

Figure 5 displays an applied example of a lexical signature. Consider the scholarly paper titled "Removal Policies in Network Caches for World-Wide Web Documents" by Abrams et al. [47] published in 1996. The layout of the paper is typical for a scholarly publication as its first page displayed in Figure 5(a) shows. Besides the title, list of authors and their affiliations, the paper starts with an abstract. The abstract (shown in Figure 5(b)) briefly (often in not more than 250 words) summarizes the essence of the paper. Search engine APIs advertise a maximum query length of 2048 bytes [15] which means a query containing 250 terms is in most cases not supported. However, assuming the publication's abstract is possible as a search query, it would be very specific to this particular paper. That means, given the paper is indexed by the search engine, chances are very good that it will be returned, very likely even with a high rank. The high degree of specificity on the other hand also implies that in case the paper is not returned, the search engine would likely not return other related results but rather an empty set. With such a query we would always be in danger of over-specifying potential results. Figure 5(c) shows the 5-term lexical signature derived from the paper. The terms intuitively make sense given the anticipated topic of the paper. They seem promising to have a search engine return the paper high in rank but at the same time, they are general enough to expect other related documents to be returned. Indeed this lexical signature works very well and returns the paper in the top ranks in all three major search engines (Google, Yahoo! and Bing) as well as a other potentially relevant results.

7 INFORMATION RETRIEVAL MEASURES

7.1 Search Engine Performance

Precision and Recall

Precision and recall are well-known methods to evaluate a retrieval system's performance [90, 182]. Intuitively, precision is a measure of specificity, meaning how good the system is at rejecting non-relevant documents, and recall tells us how well a system performs in returning all relevant documents. Consider a collection of documents D that the retrieval system to be evaluated is based on, and let $|D|$ be the number of documents in this set. The set of documents relevant to a query is denoted as R, with $|R|$ being the number of relevant documents. A query against the system returns an answer set A, and $|A|$ is the number of returned documents. Equations 9 and 10 show that precision is the fraction of retrieved documents that are relevant, and recall is the fraction of relevant documents that are retrieved.

$$P = \frac{|R \cap A|}{|A|} \tag{9}$$

$$R = \frac{|R \cap A|}{|R|} \tag{10}$$

These measures imply that there is a set of documents retrieved and also a set of documents not retrieved. It is further assumed that relevance is binary. Having these two measures for IR systems makes sense since, depending on the circumstances, one may be more important that the other. A typical web surfer may always prefer all results on the first page of the result set to be relevant - high precision - whereas intelligence analysts, for example, may be more interested in retrieving as many documents as possible - high recall. So, it becomes obvious that the two measures are a trade-off. It is always possible to obtain a high recall by simply returning all documents on any query but that very likely implies low precision. Systems with high precision, on the other hand, often suffer from low recall. F-measure [182] is the harmonic mean between precision and recall and can be used to give weights to either of them (see Equation 11).

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} \;=\; \frac{(\beta^2+1)PR}{\beta^2 P + R} \;\; where \;\; \beta^2 = \frac{1-\alpha}{\alpha} \tag{11}$$

In its simplest form, where $\beta = 1$ it is written as F_1, which is short for $F_{\beta=1}$. F_1 can be computed following Equation 12.

$$F_1 = \frac{2PR}{P+R} \tag{12}$$

To further explain the concept of precision and recall, consider the following example. An index D contains 10 relevant documents to a query q. The query, however, results in a set of 20 returned documents, 5 of which are relevant. The recall value is computed as $R = \frac{5}{(5+5)}$, resulting in $R = 0.5$. That means the query retrieved half of the relevant documents but left the other half not retrieved. Precision is computed as $P = \frac{5}{(5+15)} = 0.25$, which means that only one out of four retrieved documents is relevant to the query.

Mean Average Precision

Precision and recall, as previously introduced do not take into account the ranking of the results. They, rather, treat all results as an unordered set of documents. To evaluate the performance of modern retrieval systems with ranked results, these measures need to be extended. Precision is often measured at certain low levels of retrieved results. It is referred to as *precision at n* or *P@N*, where n often is five, ten or twenty. The obvious advantage of this method is that only the relevant documents within n need to be considered and the overall number of relevant documents (denoted as $|R|$ in Equation 9) do not have to be estimated. However, that also implies a potentially low level of stability especially with a small n if we, for example, see a dramatic change in relevance at ranks just beyond n. Mean average precision (MAP) [182] measures the quality of results across all observed ranks (or recall levels). MAP is defined as the mean of average precision values for individual information needs. An information need is not necessarily a search engine query, even though it often is translated to one query or a set of queries. An information need, for example, can be:

```
Who won the men's singles competition at Wimbledon 1992?
```

which can be transformed into a query:

```
tennis AND men AND Wimbledon AND 1992 AND NOT doubles
```

MAP is computed with Equation 13.

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \tag{13}$$

where the set of relevant documents for an information need $q_j \in Q$ is $d_1 ... d_{m_j}$ and R_{jk} is the set of retrieved results ranked higher than document d_k. To further illustrate MAP, let us consider an example with two rankings R_1 and R_2, which are the result of two information needs. Each ranking contains ten documents $d_1 ... d_{10}$. In R_1 we find to be relevant d_1, d_2, d_5, d_7 and d_{10}. This translates to the following precision scores of the relevant documents at their corresponding rank: $P(d_1) = \frac{1}{1}$, $P(d_2) = \frac{2}{2}$, $P(d_5) = \frac{3}{5}$, $P(d_7) = \frac{4}{7}$ and $P(d_{10}) = \frac{5}{10}$. The average precision of R_1 AP_{R_1} therefore is

$$AP_{R_1} = \frac{(\frac{1}{1} + \frac{2}{2} + \frac{3}{5} + \frac{4}{7} + \frac{5}{10})}{5} \;=\; 0.73 \tag{14}$$

In R_2 we only find the following four results to be relevant: d_2, d_3 and d_6. This translates to the precision scores $P(d_2) = \frac{1}{2}$, $P(d_3) = \frac{2}{3}$ and $P(d_6) = \frac{3}{6}$. AP_{R_2} is computed as

$$AP_{R_2} = \frac{(\frac{1}{2} + \frac{2}{3} + \frac{3}{6})}{3} \;=\; 0.56 \tag{15}$$

MAP of R_1 and R_2 is now simply computed as

$$MAP_{(R_1,R_2)} = \frac{0.73 + 0.56}{2} \;=\; 0.65 \tag{16}$$

Discounted Cumulative Gain

Discounted Cumulative Gain (DCG) [136] has become a popular measure for web search applications. It is based on the assumptions that highly relevant documents are more useful than less relevant documents and that the lower the rank of relevant documents the less useful it is for the user since it is less likely to be examined. Therefore, relevance is seen as a *gain* from examining a document, and the gain is *discounted* at lower ranks. DCG is computed by using the relevance score for each item i in an ordered result set of size p, as shown in Equation 17.

$$DCG = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(1+i)} \tag{17}$$

To normalize DCG, we take the DCG value and divide it by the Ideal Discounted Cumulative Gain (IDCG) produced by a perfect ordering of the result set. Equation 18 shows the formula for nDCG [137].

$$nDCG = \frac{DCG}{IDCG} \tag{18}$$

Let us again use the ranking R_1 from the above example where five documents were relevant. Let us assume relevance scores between 0..3, where 0 stands for irrelevant documents, 1 for mildly relevant, 2 for relevant and the score of 3 is given to highly relevant documents. For the ten documents we observe the following relevance scores: $R(d_1) = 3$, $R(d_2) = 1$, $R(d_3) = 0$, $R(d_4) = 0$, $R(d_5) = 2$, $R(d_6) = 0$, $R(d_7) = 2$, $R(d_8) = 0$, $R(d_9) = 0$ and $R(d_{10}) = 1$. With the relevance given, we can compute DCG as

$$DCG_{R_1} = 7 + 0.63 + 0 + 0 + 1.16 + 0 + 1 + 0 + 0 + 0.29 \;=\; 10.08 \tag{19}$$

The ideal ranking with respect to the relevance would be: d_1, d_5, d_7, d_2 and d_{10} which results in a IDCG of

$$IDCG_{R_1} = 7 + 1.89 + 1.5 + 0.43 + 0.39 \;=\; 11.21 \tag{20}$$

With DCG and IDCG computed, we can finally determine nDCG as

$$nDCG_{R_1} = \frac{10.08}{11.21} \;=\; 0.9 \tag{21}$$

7.2 Rank Correlation

Spearman's ρ [203] and Kendall τ [152, 153] are frequently used rank correlation coefficients. Their values always fall in the range $[-1, 1]$. The value of 1 means a perfect agreement between the two rankings. and a value of -1 indicates a perfect disagreement. The value of 0 means the rankings are independent of each other. One difference between these two correlation measures is that Kendall τ penalizes two independent swaps in a ranking with the same weight as two sequential swaps, whereas Spearman's ρ penalizes the sequential swaps more. For example, let us consider the optimal ranking $R_o = (1, 2, 3, 4)$ and compute its correlation to the rankings $R_1 = (2, 1, 4, 3)$ and $R_2 = (1, 3, 4, 2)$. We compute $\tau(R_o, R_1) = 0.33$ and $\tau(R_o, R_2) = 0.33$, but $\rho(R_o, R_1) = 0.6$ and $\rho(R_o, R_2) = 0.4$. Both R_1 and R_2 have two swaps, which explains the same τ value. All values in R_1 need to be swapped once, but one value in R_2 needs to be swapped twice, which results in a lower ρ.

The *M-Measure* is another correlation measure. It was introduced by Bar-Ilan et al. [64] and is based on an eye-tracking study conducted by Enquiro Research [9] showing that users are much more likely to pay attention to the search results on top of a page than to the bottom set of results. A low score means the compared rankings show discordance in the high ranks, and a high score stands for concordance in the top ranks.

7.3 Textual Similarity

Edit Distance

The Levenshtein Edit Distance [177] is a common distance measure between strings or entire documents. It is defined as the minimum number of insertions, deletions and substitutions needed to transform one string into the other. For example, the edit distance between *canceled* and *cancelled* is one, and between *survey* and *surgery* is two (as seen in [62]). Possible extensions to the edit distance include assigning different weights to each operation.

Dice Coefficient

The Dice Coefficient [97] is a simple similarity measure. It is defined for two sets (A and B) as their intersection scaled by their size (see Equation 22). It returns a value between 0 and 1.

$$Dice(A, B) = \frac{2 \times |A \cap B|}{|A| + |B|} \tag{22}$$

Similar to Dice is the Jaccard coefficient [250], which is computed with Equation 23.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{23}$$

Jaro and Jaro-Winkler Distance

The Jaro distance [134, 135] and the Jaro-Winkler distance [267] as a variant of the Jaro distance measure the similarity between two strings. They are frequently used in duplicate detection especially for duplicates of names. Both scores are normalized, which means that a distance of 0 means no similarity and a distance of 1 means the compared strings are identical. For two strings s_1 and s_2, the Jaro distance J_{s_1,s_2} is computed as shown in Equation 24, where $|s_1|$ is the number of characters in s_1 and $|s_2|$ the number of characters in s_2.

$$J_{(s_1,s_2)} = \frac{1}{3} \times \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{(m-t)}{m} \right) \tag{24}$$

The number of matching characters between the two strings is indicated by m and defined as a character that is no further away than $\lfloor \frac{max(|s_1|,|s_2|)}{2} \rfloor - 1$ in the other string. The letter t stands for the number of so called transpositions which is defined as the number of matching terms in a different order divided by 2. For example, for the strings *Michael* and *Micheal*, we see $|s_1| = |s_2| = 7$ and also $m = 7$. We have two characters *ae* and *ea* that are a match but in different sequence which means $t = \frac{2}{2} = 1$. Therefore

$$J_{(s_1,s_2)} = \frac{1}{3} \times \left(\frac{7}{7} + \frac{7}{7} + \frac{(7-1)}{7} \right) \quad = \quad 0.95 \tag{25}$$

The Jaro-Winkler distance JW is based on the Jaro distance but puts more weight on similarity at the beginning of the string. It uses a constant scaling factor p for this weight, which is commonly set to $p = 0.1$. Equation 26 shows how to compute JW.

$$JW_{(s_1,s_2)} = J(s_1, s_2) + (l * p(1 - J(s_1, s_2))) \tag{26}$$

The length of the prefix that the two strings have in common (in number of characters) is indicated by l. That results for our previous two strings in the following Jaro-Winkler distance:

$$JW_{(s_1,s_2)} = 0.95 + (4 * 0.1(1 - 0.95)) \; = \; 0.97 \tag{27}$$

Cosine Similarity

One of the most successful similarity measures is based on cosine correlation. It is based on the vector space model [232], which means the two strings or documents to be compared are represented as an n-dimensional vector. The vector contains one component for each term in the document. Note that it is immaterial here that the order of terms is lost. The vector representations $\vec{V}(d_1)$ and $\vec{V}(d_2)$ of two documents d_1 and d_2 are used to compute the cosine similarity as shown in Equation 28.

$$cosine\_sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|} \tag{28}$$

The numerator is the dot product of the vectors, and the denominator is the product of their Euclidean lengths, resulting in a document length normalization of the vectors. With that Equation 28, can be rewritten as shown in Equation 29, where the unit vector $\vec{v}(d_1) = \vec{V}(d_1)/|\vec{V}(d_1)|$ and $\vec{v}(d_2) = \vec{V}(d_2)/|\vec{V}(d_2)|$.

$$cosine\_sim(d_1, d_2) = \vec{v}(d_1) \cdot \vec{v}(d_2) \tag{29}$$

Shingles

Andrei Broder introduced a technique for near-duplicate web page detection known as *shingling* in [77] and Broder et al. [78]. Given an integer k and a sequence of terms in a document D, the concept is to define the k-shingles of D to be the set of all consecutive sequences of k terms in D. An often used example is the following text:

```
a rose is a rose is a rose
```

The unique 4-shingles ($k = 4$) for this text are:

```
a rose is a
rose is a rose
is a rose is
```

with the first two shingles occurring twice in the text. Two documents are near-duplicates if their sets of shingles are nearly the same. The Jaccard coefficient for example can be used to compute the similarity of the obtained sets of shingles. Broder et al. use a fingerprinting technique to identify duplicate documents in the web. That means for each document they compute a hash value and if two hash values match they found a duplicate. One of the duplicates will be dismissed since duplicate documents by definition have identical shingles. To decrease the complexity of the method and more directly determine the similarity of documents, they further use “super shingles”. Super shingles are shingles of shingles which means if two documents have one super shingle in common they share a sequence of shingles indicating a high level of similarity.

The disadvantage of shingles is that the error in estimating similarity increases for shorter documents because short documents contain only a few shingles. Super shingles consequently make this problem even worse. Further the choice of a good k-value is essential. With a smaller k for example, the amount of shared shingles between documents can increase which can lead to a false sense of similarity. Common values for k found in the literature are $4 \leq k \leq 6$.

CHAPTER III

RESEARCH REVIEW: WEB PRESERVATION AND WEB INFORMATION RETRIEVAL

1 INTRODUCTION

This dissertation is a synthesis of the previously disparate areas of digital preservation and information retrieval. In this chapter, we will give an overview of related research in both areas and beyond.

First, we refer to research investigating the dynamic character of the web. The research shows a quantification of the amount of web pages disappearing and (re-)appearing as well as the amount of pages whose content has changed to various degrees. This part of the research review further includes work on longevity of URIs and HTTP approaches to overcome this problem.

One of the main aspects of this dissertation work is to provide a solution to the problem of missing web pages. A web page is considered missing once it returns an HTTP 404 response code. We therefore cover several examples of this phenomenon from the web and give an overview of what is currently being done by web server administrators to support the user when she encounters such a negative response.

This work is largely based on finding Mementos as well as on the performance of web search engines. The majority of our methods to rediscover missing web pages generate strings to submit as queries to conversational search engines. Hence research on web search is of interest to us in this chapter too. We refer to work investigating user's search behavior, patterns and trends. Most of this work is based on large scale log analysis of Internet search engines.

In the previous chapter we introduced the notion of lexical signatures and the ways they are generated. As we will show in Chapter V, lexical signatures are one of the methods we investigate to generate output that can be used as a search engine query to rediscover missing pages. This idea however is not entirely new, so we include related work on lexical signatures, lexical signature-based applications and TF-IDF related work in this chapter as well. Our work on TF-IDF computation will be covered in Chapter IV.

Tags used to annotate URIs are a method that we investigate in Chapter VII. The question whether tags can be used for search and how much we can gain by including tags in our web search has been the topic of various research work. We will give an overview of the significant findings in this chapter.

Relevant for Chapter VIII is work on alternative search query generation methods. In particular, the evaluation of the retrieval performance of anchor text is of interest to us. We give an insight into what other researchers have accomplished in this area in Section 6 of this chapter.

Since our experiments introduced in Chapter IV are based on web content corpora, we provide an overview of existing corpora. We compare the corpora we used with possible alternatives and reasons for our choices.

2 DYNAMIC CHARACTER OF THE WEB

2.1 Change in the Web

An early study on the rate of change of web pages was done by Douglis et al. [99]. Their research shows that HTML documents change more frequently than documents containing rather static content types such as images. They also show that the most frequently accessed resources have the shortest intervals between content modifications.

Brewington and Cybenko [73] find that the web is rather young, meaning 20% of the web pages in their sample set are not more than 12 days old and one out of four pages is younger than 20 days. Due to their analysis, the mean lifetime of a page is between 63 and 190 days with a median of 117 days. They further find that, trusting the last modified timestamp returned from web servers, most of the page modifications are made during normal US office hours between 5 a.m. and 5 p.m. Pacific Standard Time.

Cho and Garcia-Molina [85] studied changes on the web in a binary way (change/no change). They found, for example, that it took only 11 days for 50% of the *.com* domains in their sample set to change but it took almost 4 month for 50% of the *.gov* domains to change. They also found that even though web pages change rapidly overall, the actual rates of change vary dramatically from page to page. Cho and Garcia-Molina stated that the overall average change interval of a web page would be approximately 4 months.

Lim et al. [179] investigate the degree of change of web pages and the "clusteredness" of changes. A cluster of changes explains where in the document the changes happen. There is a difference, for example, whether two sentences are inserted at the top of a page as an entire new paragraph or all terms inserted individually and spread across the document. Their results show that a vast majority of pages (90%) have changes smaller than a distance of 20% to their previous version, which means a fairly low degree of change. Also, only one out of five pages shows changes that are less than 70% clustered, which indicates that the changes that are happening are being made mostly in the same "area" of the page. These findings can have an effect on how pages are stored or cached, for example, in case only the updated portions need to be replaced.

In a later work by Cho and Garcia-Molina [86], the authors introduce a model to estimate the change frequency in the web. For web pages, the actual change history is usually not known, since even crawlers with small revisitation intervals may miss some changes in between crawls. However, Cho and Garcia-Molina showed that their estimator predicts the change frequency of web pages better than a naive model based on dividing the number of observed changes by the monitoring period. Their findings are noteworthy, since web crawlers can improve their effectiveness by adjusting their revisitation frequency. Compared to the naive approach, their model detected 35% more changes.

Fetterly et al. [107] build on Cho's work and conduct a similar study on the change of web pages but expanded in terms of web page coverage and sensitivity to change. They found that, for example, larger pages (in terms of byte size) change more often and more severely than smaller pages.

Ntoulas et al. [210] investigated textual changes of web pages of a period of one year and found that the link structure of pages changes more than the actual content (they observed 25% new links each week). Their second main result was that the content of 50% of the pages (even after one year) was less than 5% different compared to their initial version.

Adar et al. [53] conducted a study exploring changes of web pages. They analyzed change on the content level, the term level and structural level of 55,000 web pages over a five week period. Roughly 65% of their pages showed some change, while the degree of change depended on the domain and structure of the page. They identify page specific ephemeral vocabulary as well as terms with high staying power, both potentially useful to determine the page's "aboutness". The authors find that various structural elements of web pages change with different rates. Adar et al. [50] introduce a tool called *Zoetrope* that enables users browse the web and adds the time dimension to it. The system lets users explore past versions of web pages or only parts of web pages. Zoetrope crawls and indexes web pages and is therefore able to offer temporal content to the user. In their study, the authors investigated the structural changes of web pages by analyzing the frequency of change of Document Object Model (DOM) elements within the pages. They found that the median survival rate of DOM elements after one day is 98%, 95% after one week, dropps to 63% after five weeks and is only 11% after one year. They found indicators that DOM elements containing more characters tend to survive longer.

2.2 Longevity of URIs

Despite well-known guidelines for creating durable URIs [69], missing pages (HTTP response code 404) remain a pervasive part of the web experience. The lack of link integrity on the web has been addressed by numerous researchers [58, 59, 93, 92]. Brewster Kahle [147] wrote an article in 1997 focusing on preservation of Internet resources. He claimed that the expected lifetime of a web page was 44 days. A different study of web page availability performed by Koehler [166] shows the random test collection of URIs eventually reached a "steady state" after approximately 67% of the URIs were lost over a 4-year period. Koehler estimated the half-life of a random web page is approximately two years. Lawrence et al. [174] found in 2000 that between 23 and 53% of all URIs occurring in computer science related papers authored between 1994 and 1999 were invalid. By conducting a multi-level and partially manual search on the Internet, they were able to reduce the number of inaccessible URIs to 3%. This confirms our intuition that information is rarely lost, rather it is just moved. Spinellis [243] conducted a similar study investigating the accessibility of URIs occurring in papers published in *Communications of the ACM* and *IEEE Computer Society*. He found that 28% of all URIs were unavailable after five years and 41% after seven years. He also found that in 60% of the cases where URIs where not accessible, a 404 error was returned. He estimated the half-life of a URI in such a paper to be four years from the publication date. The work done by McCown et al. [190] focused on articles published in the *D-Lib Magazine*. Their results show a 10-year half-life of these articles. Nelson and Allen [208] studied object availability in digital libraries and found that 3% of the URIs were unavailable after one year. Dellavalle et al. [96] examined Internet references in articles published in journals with a high impact factor (IF) given by the Institute for Scientific Information (ISI). They found that Internet references occur frequently (in 30% of all articles) and are often inaccessible within one month after publication in the highest impact (top 1%) scientific and medical journals. They discovered that the percentage of inactive references (references that

Table 3 Overview of URI Persistency Research Results

Reference **Year**	**Object of Interest**	**Amount** **Inaccessible**	**Misc** **Information**
Lawrence et al. [174] 2000	URIs in Computer Science related papers published between 1994 and 1999	23-53%	manual search revealed relevant content and thus number dropped to 3%
Koehler [166] 2002	random collection of web pages	67% after 4 years	estimated half-life 2 years
Markwell and Brooks [184] 2002	URIs in distance learning materials	16.5%	found *.gov* domain to be most stable
Nelson and Allen [208] 2002	objects in various digital libraries between 2000 and 2001	3% of objects	refers to after a manual search was done
Dellavalle et al. [96] 2003	URIs from high IF journals	3.8% after 3 months 13% after 27 months	recovered relevant information of 50% of inactive URIs
Spinellis [243] 2003	URIs published in CACM and IEEE articles	28% after 5 years 41% after 7 years	60% failure cases returned 404 error estimated half-life 4 years from publication date
McCown et al. [190] 2005	URIs in D-Lib Magazine articles published between 1995 and 2004	about 30%	majority of failures returned 404 error half-life 10 years
Sanderson et al. [234] 2011	URIs in scholarly articles from arXiv and UNT	about 25% (and not archived)	45% of URIs that exist in arXiv are not archived

return an error message) increased over time from 3.8% after 3 months to 10% after 15 months and up to 13% after 27 months. The majority of inactive references they found were in the *.com* domain (46%) and the fewest were in the *.org* domain (5%). By manually browsing the IA they were able to recover information for about 50% of all inactive references. Markwell and Brooks [184] conducted a similar study observing links from a Biochemistry course intended for distance learning for high school teachers. They also found that the number of accessible links steadily decreased, and after one year 16.5% of their links were non-viable. They observed that the *.gov* domain was the most stable one and links referring to the *.edu* domain were more transient. 17.5% of these links had disappeared within a year.

Sanderson et al. [234] study the persistence and availability of web resources that are referenced in scholarly articles. They collected such articles from the *arXiv.org* repository and the University of North Texas digital libraries. Their collection contains more than 160,000 URIs, the largest dataset analyzed thus far. The Memento framework enables them to test for archived versions of a given URI. Sanderson et al., for example, found that roughly three out of four URIs from both datasets are still available, either at their original location or in some archive. The number of still existing but not archived URIs was much greater in the arXiv dataset though. These results also indicate that approximately 25% of all URIs do not exist anymore and are not archived. Table 3 gives a brief overview of all mentioned research results on URI longevity.

2.3 Approaches to Broken URIs

In the following three sections we give a brief overview of different approaches to address the problem of broken links in the web. The one characteristic all these methods have in common is that they assume the web administrator will take action such as changeing server configurations or installing and maintaining additional software.

Hypertext Transfer Protocol

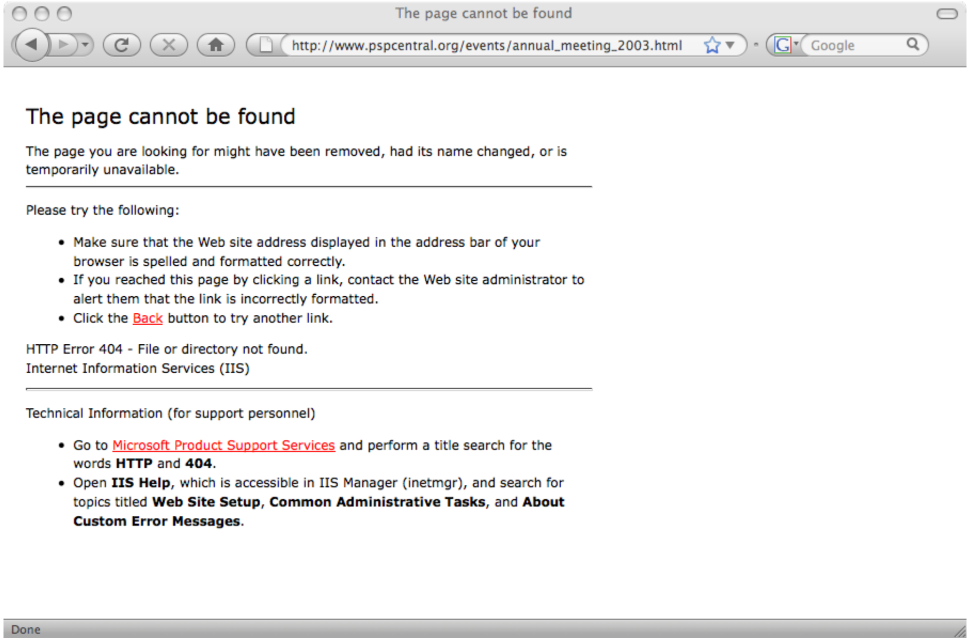
The Hypertext Transfer Protocol (HTTP) [108] by default responds with the code 404 to a requested URI that can not be found on the server. This response gives no indication of whether the erroneous condition is of a temporary or permanent nature. We cover the 404 response code seperately in Section 2.4. HTTP further provides the functionality to redirect a request to a page that has moved to a different location. The response code 301, for example, stands for a resource that has permanently been assigned a new URI. This response can result in an automatic redirect to the resource's new location. HTTP code 302, on the other hand, indicates a temporary move of a resource to a different URI. These procedures are helpful to avoid broken links if the web administrator is aware of the actual new location of the page and modifies the configuration of the web server accordingly. It is less useful for common web users since they do not have this kind of administrative access to the server.

Persistent Identifiers

Several approaches for persistent identifiers have been proposed to address the problem of broken links (also called linkrot). A Digital Object Identifier (DOI) [215] is a permanent identifier of an digital object that can be resolved to an instance of the required data. The DOI is resolved through the Handle system [249]. PURL (Persistent Uniform Resource Locator) [238] does not refer to the location of the resource itself but to a (supposedly) more persistent, intermediate location. The current technological background of PURIs is HTTP 302 redirects.

Link Authorities

Nakamizo et al. [204] developed a tool that discovers the new URI of a web page in case it has been moved. The system is based on what they call "link authorities" – reliable web links that are updated as soon as a page moves. They give an example of a company that changed names and therefore the URI to their website changed also. However, they were able to discover the new URI through the link authority of the old page, a consortium site to which the company belonged. That site was updated (with the new URI) as the company name changed. Morishima et al. [200] extended this work by enhancing the tool with heuristics based on assumptions about the location of the page that has been moved. They embrace the probability that despite the fact that the page has moved, it is still hosted by the same domain (but the URI changed). In related experiments [201, 202] they furthermore use HTTP redirect information if available and perform a keyword search with web search engines to locate the new page.



The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

Please try the following:

- Make sure that the Web site address displayed in the address bar of your browser is spelled and formatted correctly.
- If you reached this page by clicking a link, contact the Web site administrator to alert them that the link is incorrectly formatted.
- Click the Back button to try another link.

HTTP Error 404 - File or directory not found.
Internet Information Services (IIS)

Technical Information (for support personnel)

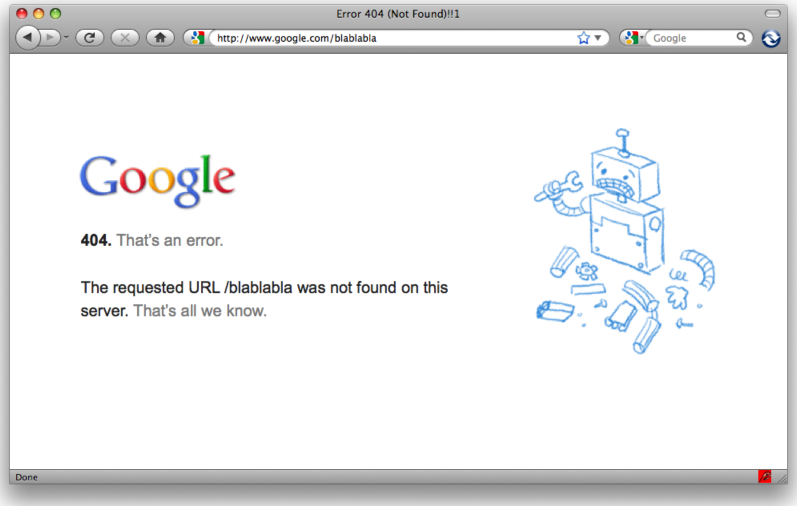
- Go to Microsoft Product Support Services and perform a title search for the words **HTTP** and **404**.
- Open **IIS Help**, which is accessible in IIS Manager (inetmgr), and search for topics titled **Web Site Setup**, **Common Administrative Tasks**, and **About Custom Error Messages**.

Fig. 6 Default 404 Page – `http://www.pspcentral.org/events/annual_meeting_2003.html`

Popitsch and Haslhofer [121, 219] introduced a tool they call *DSNotify* to handle broken links in linked data environments. DSNotify is designed as a change detection framework which means it can monitor subsets of a linked environment and detect and correct broken links. It can also notify subscribed applications about (undesired) changes in the dataset as well as forward requests to new resource locations in case they are known to the system. The functionality of DSNotify is based on the periodic access of the supervised dataset. It creates an item for each resource it encounters and can therefore extract a feature vector from its representation, which is stored along with the item. While this approach maybe suitable for a bounded dataset, it seems problematic to apply it to a web scale environment.

2.4 404 – Page Not Found

Figures 6 and 7 show two common appearances of 404 "Page Not Found" errors. Figure 6 shows a default error page returned from the web server with the message that the original URI of the PSP conference (Table 1) could not be found on the server. Figure 7, in contrast, shows the customized 404 response page from Google. It contains corporate design elements and is probably meant to appear a bit more pleasant than a standard error page. An even more sophisticated error page can be seen in Figure 8. The requested URI was `http://www.rice.edu/bla/bla/bla`, and the user is given the option to correct spelling and other obvious mistakes, use a shortened URI trying to access a page on a higher level within the same domain and search for copies of the requested page in the Google cache and the IA. Unfortunately, Rice University has decided to discontinue their "service", and the website is returning a default 404 error page now. The example shown in Figure 9 requires



Error 404 (Not Found)!!1

http://www.google.com/blablabla

404. That's an error.

The requested URL /blablabla was not found on this server. That's all we know.

Done

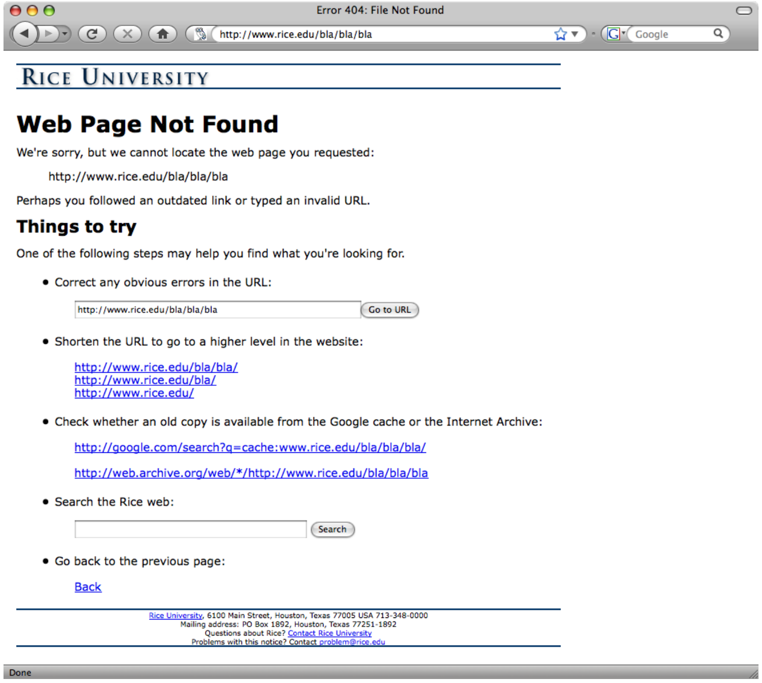
Fig. 7 Customized 404 Page – `http://www.google.com/blablabla`

a browser plugin. The add-on is called *ErrorZilla* [10] and it is designed for the Mozilla web browser. It catches 404 errors and provides several options for network layer operations (ping, trace, etc.) and also offers a search in the Google cache and the index of the IA. ErrorZilla also provides the option to "coralize" the URI, which means to use Coral [33], a free peer-to-peer content distribution network, to access the requested URI. To use Coral, `.nyud.net` is simply appended to the URI.

The work of Francisco-Revilla et al. [111] is related in the sense of missing and changing web pages. They have developed the Walden's Paths Path Manager, which is a tool that allows users (the target group is school teachers) to construct trails or paths using web pages which are usually authored by others. The path can be seen as a meta-document that organizes and adds contextual information to those pages. Thus, part of their research is about discovering relevant changes to websites. Simply comparing the candidate page with a cached copy may not be sufficient for them because some changes are actually desirable and should not be automatically dismissed. It is possible that pages change on a constant rate (such as weather or news sites) and therefore a simple comparison is not sufficient. Their focus, however, is on discovering *significant* changes to pages. Their evaluation of change is based on document signatures of paragraphs, headings, links and keywords. They also keep a history of these values so that a user can actually determine long-term as well as short-term changes. Just recently they redesigned the software and launched version four of the system [72].

Soft 404 Error

Bar-Yossef et al. [65] introduced a formalized decay measure for the web. They realized that not only single web pages but collections and even entire neighborhoods of the web show significant decay. They also focus on the problem of 404 errors. Inarguably, detecting hard 404s (the case



RICE UNIVERSITY

Web Page Not Found

We're sorry, but we cannot locate the web page you requested:

http://www.rice.edu/bla/bla/bla

Perhaps you followed an outdated link or typed an invalid URL.

Things to try

One of the following steps may help you find what you're looking for.

- Correct any obvious errors in the URL:

  http://www.rice.edu/bla/bla/bla

- Shorten the URL to go to a higher level in the website:

  http://www.rice.edu/bla/bla/
  http://www.rice.edu/bla/
  http://www.rice.edu/

- Check whether an old copy is available from the Google cache or the Internet Archive:

  http://google.com/search?q=cache:www.rice.edu/bla/bla/bla/

  http://web.archive.org/web/*/http://www.rice.edu/bla/bla/bla

- Search the Rice web:

- Go back to the previous page:

  Back

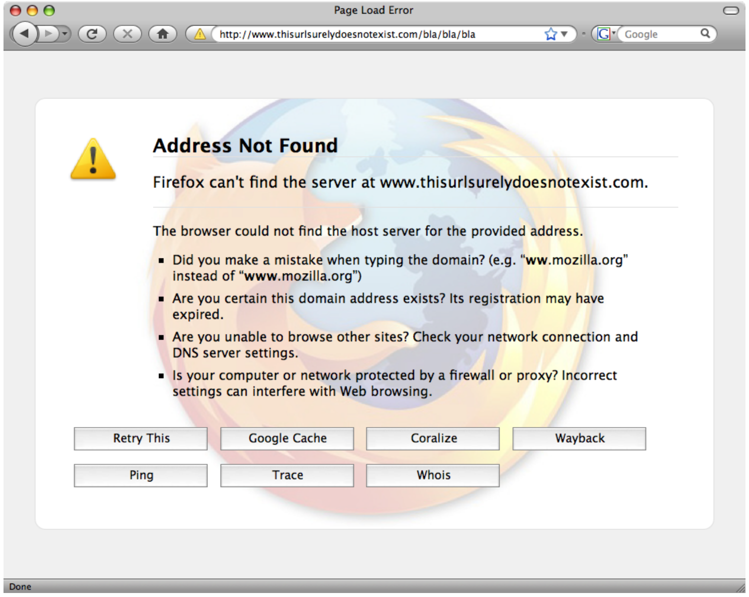
Rice University, 6100 Main Street, Houston, Texas 77005 USA 713-348-0000
Mailing address: PO Box 1892, Houston, Texas 77251-1892
Questions about Rice? Contact Rice University
Problems with this notice? Contact problem@rice.edu

Fig. 8 More Sophisticated 404 Page – `http://www.rice.edu/bla/bla/bla`

where the 404 is indeed returned) and detecting syntactically incorrect URIs is trivial, but they describe an algorithm to detect so called "soft 404s" – actual "Page Not Found" errors that do not return the HTTP response code 404 but instead return code 200 (meaning "OK"). The user's request causes a 404 internally on the server side but due to its configuration, a customized page is returned along with the response code 200. Similar to customized "Page Not Found" response pages (that come with the response code 404), the main benefit of soft 404s is that the layout of the returned page is held in the corporate design of the company often including the logo and an opportunity for the user to search the website for other pages. Figure 10 shows an example of a soft 404 response. The URI `http://www.baer.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=3378&category_id=1310&option=com_virtuemart&Itemid=10` is the result of a product search with the web site's search tool. This is a common error where the script exists (thus a 200 response) but the arguments supplied to the script no longer identify a resource in the database the script accesses. A search in Google for *ford master cylinder* restricted to this site with

`http://www.google.com/search?q=ford+master+cylinder+site%3Abaer.com`

will return several results with the same soft 404 error. The HTTP headers of the response to the requested URI shown in Figure 10 are displayed in Figure 11(b). We can see that the server responds with a HTTP 200 response code meaning "OK". Figure 7 and the corresponding headers shown in Figure 11(a) display the response to another resource that does not exist `http://google.`



Address Not Found

Firefox can't find the server at www.thisurlsurelydoesnotexist.com.

The browser could not find the host server for the provided address.

- Did you make a mistake when typing the domain? (e.g. "**ww**.mozilla.org" instead of "**www**.mozilla.org")
- Are you certain this domain address exists? Its registration may have expired.
- Are you unable to browse other sites? Check your network connection and DNS server settings.
- Is your computer or network protected by a firewall or proxy? Incorrect settings can interfere with Web browsing.

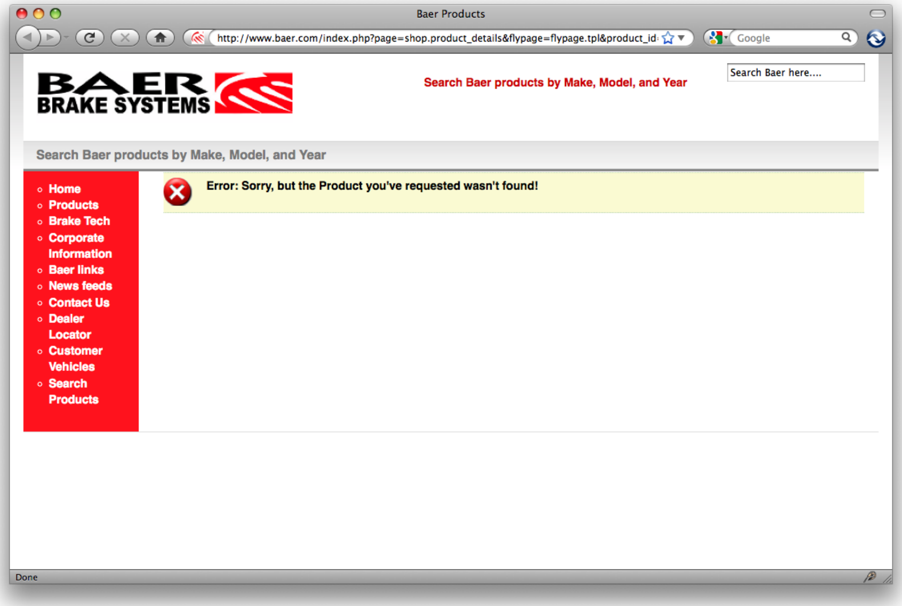
Fig. 9 404 Page with Add-On ErrorZilla – `http://www.thisurlsurelydoesnotexist.com/bla/bla/bla`

`com/blablabla`, but this server sends the proper 404 response code.

The URI `http://jcdl2007.org` is another example of a soft 404. It used to be the canonical URI for the JCDL conference in 2007, but no longer returns conference-related content. Our assumption is, similar to the example of the Hypertext 2006 conference website introduced in Chapter I, that the website administrators did not renew the domain registration and someone else took over, maybe hoping to take advantage of the popularity the site has gained thus far. Such a phenomenon is also known as "parked web sites". The report by Edelman [102] provides a good overview and gives a specific case study on the issue.

If we send a request to a resource that very likely does not exist on this server, such as `http://jcdl2007.org/blablabla` we would expect a 404 error in return. As seen in Figure 12, the web page tells us that the requested resource no longer exists on this server. The figure also shows the HTTP headers returned from this request. It confirms that the server responds with a 200 code meaning it is indeed a soft 404. In this case however, we are led to believe that the intentions of the website administrator are somewhat suspicious since the page still contains some random content about digital libraries, probably to make sure it will keep its ranking in search engines.

Bar-Yossef et al. [65] present an approach to learn whether or not a web server produces soft 404s. This is achieved by sending two requests to a "suspicious" server. The first request is asking for the page of interest and the second for a page that with very high probability does not exist. It then compares the server behavior for the two returns such as number of redirects. The content of the returned pages are also compared using *shingles* (Chapter II, Section 7.3) so that, in case the two behaviors and the content of the returned pages are very similar, the algorithm gives a clear indication

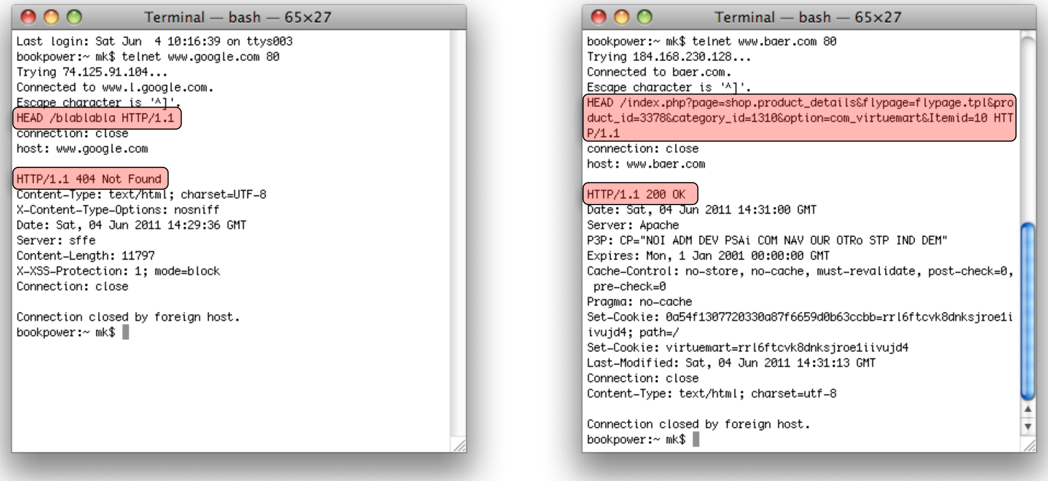


Error: Sorry, but the Product you've requested wasn't found!

Fig. 10 Soft 404 Page at `http://www.baer.com`

of having detected a soft 404. The authors are aware of the problem with this approach: a case where a soft 404 URI and a legitimate URI both redirect to the same page. An example they give is the URI: `http://www.cnn.de` (the German site of CNN) used to redirect to `http://www.n-tv.de` which is a German 24 hour news channel (similar to Bloomberg). The problem is a resource that surely does not exist such as `http://www.cnn.de/blablabla` also redirects to `http://www.n-tv.de`. The solution the authors present is to declare a URI that is the root of a web site (`http://www.cnn.de` in this example) can not be a soft 404. Bar-Yossef et al. found almost 30% of their URIs to be cases of soft 404.

Since soft 404s usually return pages that the user did not expect, the difference between the expected page (the one the user has experienced before) and the actually returned page can be significant. Search engines have therefore identified soft 404s as undesirable and consider them as not useful to the user. Google calls for a farewell to soft 404s [12] and more recently announced that its crawling mechanism can detect soft 404s and interactively work with a website administrator to overcome this detriment [11]. While this procedure in general appears desirable, it has been critiqued as inaccurate [16] and shown to also include server side errors displayed as 500-level response codes [17]. Yahoo! and its web crawler *Slurp* have been seen to apply the approach introduced by Bar-Yossef et al. to detect soft 404s [44]. Conceptually, this method is similar to the method introduced by Francisco-Revilla et al. [111], since they are both comparing the expected with the actual returned page.



```
Last login: Sat Jun  4 10:16:39 on ttys003
bookpower:~ mk$ telnet www.google.com 80
Trying 74.125.91.104...
Connected to www.l.google.com.
Escape character is '^]'.
HEAD /blablabla HTTP/1.1
connection: close
host: www.google.com

HTTP/1.1 404 Not Found
Content-Type: text/html; charset=UTF-8
X-Content-Type-Options: nosniff
Date: Sat, 04 Jun 2011 14:29:36 GMT
Server: sffe
Content-Length: 11797
X-XSS-Protection: 1; mode=block
Connection: close

Connection closed by foreign host.
bookpower:~ mk$
```

(a) Hard 404

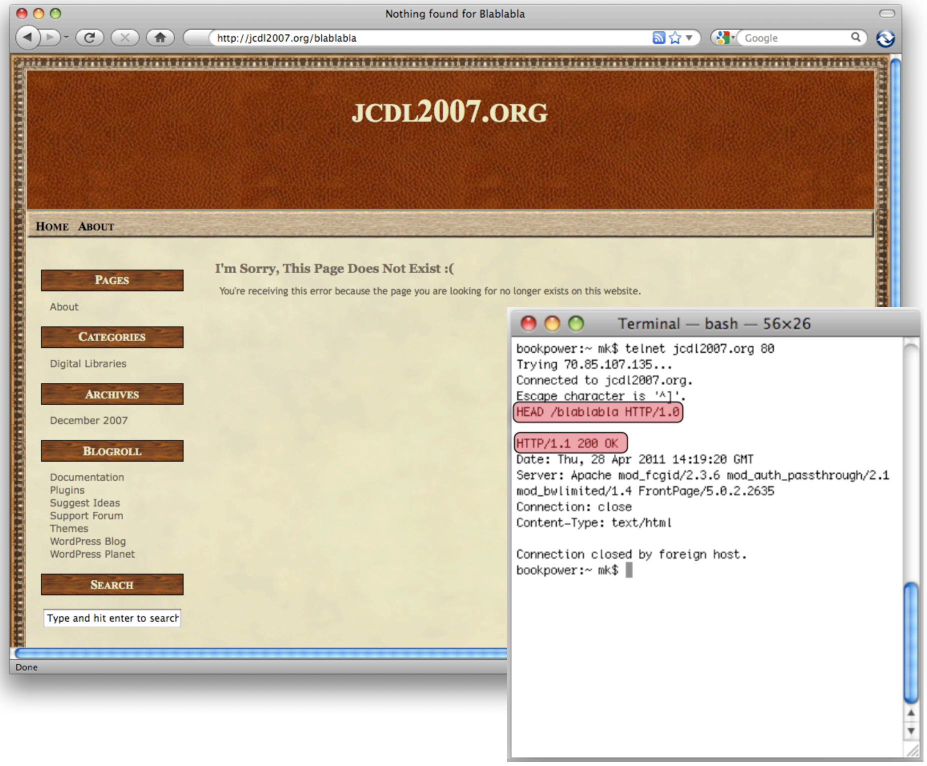
```
bookpower:~ mk$ telnet www.baer.com 80
Trying 184.168.230.128...
Connected to baer.com.
Escape character is '^]'.
HEAD /index.php?page=shop.product_details&flypage=flypage.tpl&pro
duct_id=3378&category_id=1310&option=com_virtuemart&Itemid=10 HTT
P/1.1
connection: close
host: www.baer.com

HTTP/1.1 200 OK
Date: Sat, 04 Jun 2011 14:31:00 GMT
Server: Apache
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
Expires: Mon, 1 Jan 2001 00:00:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0,
 pre-check=0
Pragma: no-cache
Set-Cookie: 0a54f1307720330a87f6659d0b63ccbb=rrl6ftcvk8dnksjroe1i
ivujd4; path=/
Set-Cookie: virtuemart=rrl6ftcvk8dnksjroe1iivujd4
Last-Modified: Sat, 04 Jun 2011 14:31:13 GMT
Connection: close
Content-Type: text/html; charset=utf-8

Connection closed by foreign host.
bookpower:~ mk$
```

(b) Soft 404

Fig. 11 HTTP Headers of Hard and Soft 404 Responses

2.5 Near-Duplicate Web Pages

One part of the dynamic character of the web is the fact that pages disappear. Another aspect however is the observation that numerous duplicates or near-duplicates for web pages exist. Even though research in the field of (near-)duplicate web page detection has introduced techniques intended for optimizing web crawling (by identifying and hence omitting duplicates) such duplicate web pages can still be of use for web page preservation. The shingle technique introduced by Broder et al. (see Chapter II, Section 7.3) can be used to cluster web pages based on their shingle values, which, for example, can be applied for what the authors propose as a "Lost and Found" service for web pages. Charikar [83] introduced another technique that became very popular. It is a hashing function that changes relative to the changes of the input set. That means entire web pages or subset of pages can be compared by their hash values. Henzinger [124] compared both techniques and found that both perform well on identifying (near-)duplicates on different sites but do not perform well for duplicates within the same site. Hence she proposes a combination of both methods to overcome that weakness. Fetterly et al. [106] created clusters of near-duplicate web pages while their similarity was measured using shingles and other means. They found that about 28% of their pages were duplicates and 22% were virtually identical. Their results also support the intuition that a lost page often can be restored by finding other pages on the web. The work done by Brin et al. [75] is also related since they introduced further methods to detect copied documents by comparing "chunks" of the documents. In [74] Brin transforms text into a metric space and computes the document similarity based on the distances in the metric space. Baeza-Yates et al. [61] also show that a significant portion of the web is created based on already existing content. They explored



I'm Sorry, This Page Does Not Exist :(

You're receiving this error because the page you are looking for no longer exists on this website.

```
bookpower:~ mk$ telnet jcdl2007.org 80
Trying 70.85.107.135...
Connected to jcdl2007.org.
Escape character is '^]'.
HEAD /blablabla HTTP/1.0

HTTP/1.1 200 OK
Date: Thu, 28 Apr 2011 14:19:20 GMT
Server: Apache mod_fcgid/2.3.6 mod_auth_passthrough/2.1
mod_bwlimited/1.4 FrontPage/5.0.2.2635
Connection: close
Content-Type: text/html

Connection closed by foreign host.
bookpower:~ mk$
```

Fig. 12 Soft 404 Page at `http://jcdl2007.org/blablabla`

the notion of genealogical trees for web pages where children would share a lot of content with their parents and are likely to become parents themselves.

3 SEARCHING THE WEB

A lot of research has been done analyzing the user's search behavior and intent. Jansen et al. [133] conducted an extensive study on user queries on the web in 2000. They analyzed the log files of the search engine *Excite* and found that 35% of the queries were unique (first query on one distinct user), 22% modified (second, third, etc. query of one user which is modified due to added/deleted words) and 43% were identical (same query as the one preceded by the same user). The large amount of identical queries has two reasons: 1) the user tries to re-find some result and 2) the search engine automatically issues the same query if the user views the second, third, etc. result page. They also found that about two thirds of all users issued just one query and most of the users that issue more than one do not add/delete a lot of their initial query, the most frequent change was a modification of the terms (where adding and deleting occurs with a similar frequency). The mean number of pages examined per user was 2.35, while more than half of the users did not access the second result page. The mean number of query terms was 2.21, and boolean operators were not frequently used (most frequently: AND in 8% of all queries). They also analyzed the distribution of the query terms, finding a few terms used very often and a long tail of terms only used once (with a broad middle in between). The term distribution they found is similar to the Zipf distribution [49].

Jansen and Spink [132] compared nine published studies providing data about searching through web search engines. Since these datasets were published between 1997 and 2002, they are able to draw conclusions about trends in search. They found, for example, that the number of queries per user in one session remained stable, which can be interpreted as such that the complexity of interaction between the systems and the user is not increasing. Considering that the authors also found the number of users viewing only one result page to be increasing (also supported in Spink et al. [244]), one can even argue for a decrease of that complexity. As for the length of the queries, there seemed to be a slight decrease in one-term queries from 30% to 20%. They further found a slight shift in topics that were searched for. Nearly half of all queries (in US search engines) were for people, places, and things. This number appeared to be steadily increasing while the number of queries for sex and pornography, as well as for entertainment and recreation, were decreasing.

Weber and Jaimes [265] conduct a complex analysis of the Yahoo! query logs in order to gain insight into *who* searches for *what* and *how*. They examine demographics, session statistics and query types (navigational, informational and transactional as introduced by Jansen et al. [131]). With this method they find "stereotypical" results such as white conservative men searching for business and gardening, baby boomers searching for finance-related topics and liberal females seeking shopping-related content and maintaining long sessions which hints at a browsing and comparison behavior. Baeza-Yates et al. [60] applied supervised and unsupervised learning techniques to identify the users' intent in a search query. Unlike Jansen et al. [131], they defined informational, non-informational and ambiguous queries. Zaragoza et al. [274] asked the questions whether the problem of web search was solved and whether all rankings across major search engines were the same. They point out, similar to Saraiva et al. [237], that query frequencies follow a power law distribution [48]. They found that the three major search engines, namely Google, Yahoo! and Bing, all perform very well for the very frequently issued queries and for navigational queries. They saw differences in performance at the small end of the power law tail and for non-navigational queries. According to their results each of the three search engines seem to work on different portions of the tail.

Pass et al. [216] provide a "Picture of Search" from the perspective of the AOL search service. Their analysis shows, for example, a mean query length of 3.5 terms. They also find that 28% of all queries are reformulations of a previous query, in which cases the average query is reformulated 2.6 times. That indicates that users frequently do not find what they are looking for and change their input in case the desired result is not shown immediately, i.e. on the first result page. Pass et al. also find indicators for a geographical distribution of queries. They see that between 12% and 28% of queries include a local aspect. The query *los angeles dodgers*, for example, is most frequently issued in the greater Los Angeles area. Lazonder et al. [175] investigate the relationship between the level of experience of users for web search and their search performance. They found that more experienced users were faster and more efficient to successfully complete the "locate site" tasks. However, they did not find a significant performance difference between experienced and novice users for the "locate information" task. That means when it comes to discovering information on a given web site, the level of experience is insignificant. It is well-known that domain expertise enhances search performance in terms of efficiency and effectiveness [197, 217, 127, 183]. It has also been shown that users with domain expertise do better in judging the complexity of search tasks [154] and also show a different strategy (for example, prioritizing) to approach search tasks [261]. Teevan et al. [254] investigate search for socially-generated content on Twitter and compare it to web search. They explore that users search on Twitter for temporally relevant content and information about people. While this is not much different from web search, they find Twitter queries to be shorter and less likely to change within a user session. They claim Twitter users repeat their queries in order to monitor the associated search results. Web search queries, in contrast, often evolve with the user learning about a certain topic of interest. Sun et al. [248] confirms these findings by observing that blog and news queries often refer to people and temporally-relevant content. Mishne and de Rijke [199] also see blog searches mainly for people and products but also find that the users' overall search patterns are similar to web search patterns.

Ji et al. [140] introduce a method to support the user in generating well-performing search queries. This especially becomes helpful for searches where the user has only limited knowledge about the desired information. Modern search engines already offer some level of support by autocompleting the query while the user is typing. If she, for example, starts typing “andre ag”, Google will automatically offer queries for the Andre Agassi Foundation, his book, his school and quotes of him. The approach by Ji et al. is based on fuzzy keywords, meaning a query term can occur in multiple attributes in the target document and the order of terms is not relevant. It further includes documents that approximately match the query as potentially relevant.

3.1 Re-finding Web Pages

The work done by Teevan et al. [251, 252, 253] addresses repeat searches − instances of the same query string issued by the same user within a 30-minute period. She analyzed the Yahoo! log files over a period of one year and found that repeat searches are indeed prevalent. 40% of all queries led to clicks on the same result by the same user but only 7% were clicks on URIs clicked by multiple users which indicates that people much more likely click on results that they themselves have seen before. Teevan proposed a system to predict the likelihood of users clicking on the same result, which has obvious value to search engines. She also implemented a browser plugin that interfaces with existing search engines. It catches a user query that is similar to an older query. The plugin fetches the response (current result set) from the search engine and also the previously viewed results (from a local cache). It merges both result sets and marks the previously viewed results. Therefore, users are quickly able to re-find web resources that they have discovered before.

Adar et al. [51] investigate the reasons behind the users' revisiting behavior of web pages. They analyze web interaction logs of more than $600,000$ users and observe four revisitation patterns. First, they identify a fast group, meaning pages being revisited with intervals of less than one hour, for example, due to reloads of the page. Examples for these pages are shopping sites (or hubs) offering a broad range of products a user may click on but shortly after return to the overview page in order to navigate to the next product. These pages see almost no long-term revisitations. The second category is a medium group where pages get revisited hourly or daily. Examples are portals such as bank and news pages as well as web mail. Classic characteristics for such pages are shorter URIs and a lower mean directory depth compared to the fast group. The third group is called the slow group and contains pages that were revisited in intervals longer than one day. Special search pages for travel or jobs, for example, belong to this category, as well as event and movie pages most frequently visited on weekends. Lastly, they identified a hybrid group with web pages showing characteristics from several other groups. Local Craig's List homepages are considered an example for this group.

Adar et al. [52] explore the relationship between content changes of web pages and users' revisitation behavior of these pages. Their findings support the intuition that pages that change more frequently are being revisited more often. However, the amount of change varies, which makes them conclude revisitation is not purely about the amount of change but more about what is changing. Regarding the frequency of change, they found that pages changing more often within a shorter period of time were revisited more frequently also.

The authors argue that understanding the user's revisitation pattern can help improve the interaction between content providers and Internet users. For example, search engines can benefit from better organizing their results and browsers could improve the display of the pages content.

4 LEXICAL SIGNATURES OF WEB PAGES

So far, little research has been done in the field of lexical signatures for web resources. Phelps and Wilensky [218] first proposed the use of lexical signatures for finding content that had moved from one URI to another. Their claim was "robust hyperlinks cost just 5 words each", and their preliminary tests confirmed this. The lexical signature length of 5 terms however was chosen somewhat arbitrarily. Phelps and Wilensky proposed "robust hyperlinks", a URI with a lexical signature appended as an argument such as:

```
http://www.cs.berkeley.edu/~wilensky/NLP.html?lexical-signature=
texttiling+wilensky+disambiguation+subtopic+iago
```

where the lexical signature is everything after the "?" in the URI. This example was taken from Robert Wilensky's website. They conjectured that if the above URI would return a 404 error, the browser would take the appended lexical signature from the URI and submit it to a search engine to find a similar page or a relocated copy of the page.

Park et al. [214, 213] expanded on the work of Phelps and Wilensky, studying the performance of 9 different lexical signature generation algorithms (and retaining the 5-term precedent). The performance of the algorithms depended on the intention of the search. Algorithms weighted for TF were better at finding related pages, but the exact page would not always be in the top n results. Algorithms weighted for IDF were better at finding the exact page but were susceptible to small changes in the document (e.g., when a misspelling is fixed). They also measured the performance of lexical signatures depending on the results returned from querying search engines and the position of the URI of interest in the result set. They do not compute a performance score but distinguish between four performance classes:

1. the URI of interest is the only result returned
2. the URI is not the only one returned but it is top ranked
3. the URI is not top ranked but within the top 10

4. the URI is not returned in the top 10.

Park et al. thoroughly evaluated algorithms for generating lexical signatures, but they only re-checked their results once after ten months. Like Phelps and Wilensky, they did not perform an extensive evaluation of how lexical signatures decay over time and did not explore lexical signatures of more or less than five terms.

4.1 Lexical Signatures based Applications

Harrison and Nelson [120] developed a system called *Opal* which uses lexical signatures to find missing web pages using the WI. They adopted Phelps and Wilensky's function to generate lexical signatures but also did not explore the reasoning behind the 5-term lexical signature. Part of their framework is the Opal server which catches 404 errors and redirects the user to the same page at its new URI or to a different page with related content. Opal servers learn from experience and are able to exchange data with other Opal instances by utilizing the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) introduced in [170, 257]. The drawback of this approach is the effort required by the web administrator since the web server configuration for each Opal-enabled server needs to be modified and maintained.

Wan and Yang [263] explore the "WordRank"-based lexical signatures. This lexical signature generation method takes the semantic relatedness between terms in a lexical signature into account and chooses "the most representative and salient" terms for a lexical signature. The authors also examined 5-term lexical signatures only and found (similar to Park et al. [214, 213]) that *DF*-based lexical signatures are good for uniquely identifying web pages and hybrid lexical signatures (variations of TF-IDF) perform well for retrieving the desired web pages. They claim however that WordRank-based lexical signatures perform best for discovering highly relevant web pages in case the desired page can not be located.

Staddon et al. [245] introduce a lexical signature-based method for web-based inference control. Following the TF-IDF method, they extract salient keywords (which can be considered a lexical signature) from private data that is intended for publication on the Internet and issue search queries for related documents. From these results they extract keywords not present in the original set of keywords which enables them to predict the likelihood of inferences. These inferences can be used to flag anonymous documents whose author may be re-identified or documents that are at risk to be (unintentionally) linked to sensitive topics.

4.2 TF-IDF Values in Hyperlinked Environments

Sugiyama et al. [246, 247] uses the TF-IDF scheme to represent the content of web pages without particularly focusing on lexical signatures. They claim that for documents in a hyperlinked structure like the web the content of neighboring pages need to be exploited too, in order to obtain more accurate descriptions of a page. Their research is based on the idea that the content of a centroid web page is often related to the content of its neighboring pages. This assumption has been proven by Davison in [94] and Dean and Henzinger in [95]. The authors define neighboring pages as pages that refer to the centroid page (inlinks for the centroid) and pages the centroid links to (outlinks). Sugiyama et al. show that by refining the original TF-IDF with input from the neighborhood, the performance of the lexical signature in terms of precision and recall while querying search engines for related pages can be improved. However, the problem of off-topic links − for example links that refer to the download page of the Adobe Acrobat Reader or Macromedia Flash Player, pages that may support the user in viewing content of a page but are unrelated to the actual content − is not particularly addressed in this work.

Zhu and Rosenfeld [275] used Internet search engines to obtain estimates for DF values of unigrams, bigrams and trigrams. They plotted the obtained phrase count (comparable to what we call term count TC in Chapter IV) and web page count (our DF in Chapter IV) and were able to apply a log-linear regression function to all three n-gram cases which implies a strong correlation between the obtained TC and DF values. Zhu and Rosenfeld also found that the choice of one particular search engine did not matter much for their results. Keller and Lapata [151] also used Internet search engines to obtain DF values for bigrams. They compare these values to corpus frequencies (comparable to our TC) obtained from the British National Corpus (BNC) and the North American News Text Corpus (NANTC). Despite significant differences between the two corpora, Keller and Lapata found a strong correlation between the web based values (DF) and the values obtained from the two text corpora (TC). The main application Keller and Lapata see for their results is estimating frequencies for bigrams that are missing in a given corpus. Nakov and Hearst [206] give a justification for using the search engine result count (DF) as estimates for n-gram frequencies (which can be TC). They chose the noun compound bracketing problem [205] to demonstrate their results. The compound bracketing problem has traditionally been addressed by using n-gram frequencies. They found that the n-gram count from several Internet search engines differs, and these differences are measurable but not statistically significant. They come to the conclusion that the variability over time and across different search engines represented by the obtained n-gram frequencies does not affect the results of a specific natural language processing task.

5 TAGS FOR WEB SEARCH

Web sites such as Delicious, which enable users to annotate and share their bookmarks with the entire web community, play an important role for web users. With the emerging phenomenon of social annotations, a lot of research has been done to investigate the value of tags for search in the web. Bao et al. [63], for example, observe that tags from Delicious are usually good summaries of the corresponding web pages and the count of the tags indicates the popularity of the pages. They introduce two algorithms which incorporate these observations into the ranking of web pages. The first, called *SocialSimRank (SSR)*, returns a similarity estimate between tags and web queries, and the second, *SocialPageRank (SPR)*, captures the popularity of web pages. SSR is shown to find semantic associations between tags and a query which can improve web search and SPR ranks pages from the users' perspective. Yanbe et al. [270] also exploit social annotations from Delicious to enhance web search. They propose to combine the current link-based ranking methods with characteristics derived from social annotations. They introduce *SBRank* which captures the popularity of a page. It is computed by counting the number of times a page has been bookmarked (voted for by users) and can therefore be seen as a simplistic version of SPR described above. The authors implemented a prototype search portal which enables searching by “common query terms” as well as by tags. The user can also give certain weight to the source e.g., have tags twice as important for the query as the common terms. The ranking of the results is determined by combining link-based methods and the output of SBRank. Morrison [138] also investigated the usefulness of tags for search and found in an extensive study that search in folksonomies can be as precise as search in major modern web search engines. By comparing Delicious data with search engine log data, Krause et al. [169] found that tags and search terms both follow a power law distribution. That implies a low overall overlap and an increased overlap for the more frequent terms. They further found sparse overlap in Delicious and search engine rankings but if there was overlap it occurred at the top end of the rankings. Heymann et al. [126] conducted probably the most extensive study on tags with a dataset of about 40 million bookmarks from Delicious. They observed that roughly 9% of the top 100 results for search queries (from the AOL logs) are annotated in Delicious. This coverage increases to 19% considering only the top 10 results. That means despite the relatively small coverage of web pages Delicious URIs are disproportionately common in search results. They also found that tags significantly overlap with popular search terms which indicates that tags can indeed help locating relevant pages. Interestingly, despite the overlap, tags and search terms were not correlated. Their results further show that about half of the tags occur in the content of the page they annotate and 16% even occur in the page's title. They additionally found that in one out of five cases the tags neither occur in the page nor in the page's inlinks or outlinks. They conjecture that tags therefore can provide data for search that is otherwise not available. However, they state that annotated URIs are rather sparse compared to the size of a modern search engine's index.

Bischoff et al. [71] analyze tags from three different systems: `delicious.com`, `last.fm` (a music portal) and `flickr.com` (a photo portal). Due to the variety of the sources they classify the tags into eight categories. They found that different categories are important for different domains e.g., the category *Topic* was dominant for tags from Delicious and Flickr since it describes the domain and anything that can be seen on a picture but the category *Type* was prominent for Last.fm tags since it describes the file format as well as the music genre. Therefore the predicted usefulness of tags for web search (assessed by a user study) depends on the category of the tags. This observation is intuitively confirmed since tags that belong to the *Location* category are more useful to discover an image on Flickr than piece of music from Last.fm or a bookmark from Delicious. Bischoff et al. confirm the findings of Heymann et al. [126] with 44.85% of tags from Delicious occurring in the text of the annotated page and hence claim that more than 50% of the tags provide new information about the URI they describe which in fact could be useful for web search.

Yanbe et al. [270] propose a social bookmarking-based ranking and use it to enhance existing link-based ranking methods. They also find that tag proportions stabilize over time which means users eventually come to an agreement over tags and even copy each other. The work done by Bao et al. [63] incorporates the frequency of tags used to annotate URIs as an indicator for its popularity and quality.

Hayes et al. [123] found that tags performed poorly in clustering textual documents (blogs) compared to even a simple clustering approach. Tags were imprecise and showed a low recall. However, Hayes and Avesani [122] found that within the cluster the tags were very useful to detect cluster topics. These results confirm the general notion that tags in general are often too vague (for search) but in special cases contain valuable information.

6 FURTHER SEARCH QUERY GENERATION METHODS

Henzinger et al. [125] provide related web pages to TV news broadcasts using a search engine query containing two terms. This query summarizes the content of the broadcast by extracting its closed captions. Various algorithms are used to compute the scores determining the most relevant terms. These terms can be thought of as a 2-term lexical signature. The terms are used to query a news search engine where the results must contain all of the query terms. The authors found that one-term queries return results that are too vague and three-term queries too often return zero results, thus they focus on creating two-term queries. Bharat and Broder [70] investigated the size and overlap of search engine indexes. They tried to discover the same page in several different search engine indexes. Their technique was generally based in randomly sampling URIs from one index and checking whether they exist in another. They introduce the notion of "strong queries", a set of salient keywords representing the randomly sampled URI and used as the query against the other indexes, hoping it would return the URI. To generate the strong queries they simply used the n terms from the document that least frequently occurred in their entire data set and formed a conjunctive query with the "AND" operator. With this approach they, similar to lexical signatures, leveraged the power of IDF. Martin and Holte [185] introduced the notion of a content-based address. They argue that URIs are due to the dynamics in the web unreliable and a content-based addressing approach that results in a set of words and phrases, when used as a search engine query, is able to overcome these changes. Such queries were supposed to retrieve the document in the top 10 results and also have the potential to return other related documents. They introduce a system called *QuerySearch* which initially generates a query for a document based on its terms frequency and position in the document. It then tries to find several simplified versions of the query which all can form the content-based address of the document. Jiang et al. [141] later enhanced this approach and focused on query suggestion methods.

Fuxman et al. [115] introduce an approach for the problem of keyword generation. Their problem domain is identifying appropriate keywords for a concept or domain that advertisers can purchase from search engines. This problem is relevant for both search engines and advertisers. All major search engines generate revenue with advertisement and offering a broad set of relevant keywords enables them to even reach small businesses. On the other hand businesses have the opportunity to advertise their services through a powerful medium. The authors analyze the click log of a major search engine and define a relationship between a concept (e.g., shoes), a URI and a query. If a user clicks on a certain URI as a result of her query one can argue for an association between the query and the URI. The authors further define an association between URIs and concepts (e.g., *www.shoes.com* and *shoes*) which then means that a query can be associated to a concept. Fuxman et al. employ Markov Random Fields [178] to model the graph defined by these pairwise relationships and apply a random walk algorithm [100] to efficiently generate keywords.

Unlike Fuxman et al., the approach taken by Turney [255] for keyword extraction is based on documents and their content. He proposed GenEx which is a rule-based system to extract keyphrases from text. The Keyword Extraction Algorithm (KEA) introduced by Witten et al. [268] and Gutwin et al. [112] works similarly by training the system in advance based on a number of textual features such as TF-IDF and the location of the term in the document. KEA was enhanced with web specific features such as the estimated number of results from web search engines in [256] and hyperlink data in [150].

The Keyphrase Identification Program (KIP) introduced by Wu et al. [269] is also based on a prior knowledge of positive samples for keyphrases to train the system. KIP is specialized on extracting significant nouns. Hulth [129] explored the use of natural language techniques for keyword extraction. She showed that better results can be obtained when splitting text into np-chunks and considering part-of-speech (POS) tags for keyword extraction compared to statistical measures such as term frequencies. Yih et al. [273] proposed a method to extract keywords from web pages using observed frequencies in query logs. Their approach was motivated for contextual advertisement. All of the above methods result in keywords or keyphrases that describe the "aboutness" of a textual document. Therefore they are all relevant alternatives for generating web search engine queries.

6.1 Anchor Text

It is well known that the link structure in the web holds valuable information for search. The great success of link analysis algorithms such as PageRank [76] and HITS [164] provide the evidence. However, they only consider the link structure and not the anchor text itself for identifying authoritative pages. Craswell et al. [89], for example, found that link anchor information can be more useful than the content itself for site finding. This point is supported by the work of Fujii et al. [114] on the NTCIR-5 WEB task [26]. Dou et al. [98] provided indicators that anchor text is similar to user queries for search engines but also showed that anchors within the same site are less useful than external anchors. They build a dependence model for anchor text depending on the origin of the text and give weights accordingly. This finding is supported by Eiron and McCurley [103] who investigated the properties of anchor text in a large intranet. Metzler et al. [198] proposed a method to overcome the so called anchor text sparsity problem. The goal is to enrich a document's representation even though the document might only have very few inlinks. The idea is to propagate anchor text across the web graph and aggregate anchor text of external pages linking to any given page and its internal inlinks. Yi and Allan [272] address the same anchor text sparsity problem by applying a language modeling based technique. Their approach is based on the assumption that similar pages have distinct inlinks with similar anchor texts. In case one page suffers from anchor sparsity its could benefit from a richer body of anchor text from a contextually similar page. Kraft and Zien [168] propose the use of anchor text to refine search queries. Query refinement is an often interactive process to modify insufficiently performing queries with the goal of improving returned results in terms of quality and quantity. This refinement is usually achieved by applying query expansion techniques [79, 222] or relevance feedback mechanisms [80, 117]. Kraft and Zien show that anchor text can provide terms for query refinement that perform better that terms obtained from a document's content itself. Hyusein and Patel [130] showed that indexing web documents by their title, anchor text and some emphasized text only results in poor precision and recall for search. Their findings imply that despite the usefulness of anchor text and titles for search indexing additional terms derived from the document's content is essential. Dai and Davison [91] investigate the temporal aspect of anchor text. They find that propagating historical weights of anchors over time can produce significant improvement in ranking quality. Anchor text has also been shown to be useful to enrich textual representations of other kinds of resources. Harmadas et al. [118] have shown, for example, that propagated anchor text can be useful for building textual representations of images.

Martinez-Romo and Araujo [186, 187, 188] have recently addressed the problem of broken web links. They introduce a method to recover from linkrot based mainly on querying the anchor text used to point to the missing page against a search engine. However, they also find that expanding the query with contextual data from the missing page (obtained from the Internet Archive) can improve the retrieval performance.

7 MESSAGE DIGEST ALGORITHMS

Hash functions can be used to transform the content of a resource into a non-reversible representation of that resource. They have traditionally been used for data lookup and comparison tasks. However, if we consider a web page as the input to the hash function, we can compare its output to a lexical signature since both capture the content of the page (ignoring the fact that the hash value is the transformation of the entire content and the lexical signature consists of a limited number of significant terms only). Changes in the input set (web page) are reflected in the hash value and as long as they are significant also in the lexical signature.

However, there are two problems with this approach. First, with common hash functions such as MD5 [224] and SHA-1 [101] the output changes dramatically, even with the smallest changes in the input set. Secondly, even if we would be able to implement a hash function that changes the output symmetrically to the degree of change of the input, we would not have the infrastructure to use these output sets as queries for search engines. To the best of our knowledge, none of the major general topic search engines support hash values as input. We are mostly restricted to textual content and URIs because this is the data input supported by the WI.

The *Simhash* function introduced by Charikar [83] is different. For any given input set the hash function changes relative to the modification of the input. That means if the input only changes slightly, the change in the hash functions is minor and if the input set changes significantly the change in the hash function is major. We still can not use Simhash as input for search engines but it is useful as a similarity measure of web pages. Manku et al. [181] demonstrates how Simhash can be applied to find similar web pages in order to improve the quality of a web crawler.

To demonstrate the differences between the hashing functions we show in Table 4 the base64 encoded MD5 and SHA-1 hash values of a passage from the US Declaration of Independence in two variations. The first is the original phrase as it appears in the Declaration and the second is a slightly modified version of the original. Instead of *"**We** hold these truths to be self-evident..."* the modified phrase is *"**I** hold these truths to be self-evident..."*. As we would expect, the hash values are dramatically different despite the only minor change in the input. Table 5 shows the base64 encoded Simhash value of the original and the modified phrase of the declaration In contrast to the MD5 and SHA-1 values we can see some similarity between the Simhash values. The eighth, ninth and tenth line are completely identical and the third and seventh line are similar. Simhash reports a normalized similarity score between the two hashes of 0.97 on a scale of 0 to 1 indicating a very strong similarity. Interestingly, if we make an even smaller and this time semantically insignificant

Table 4 MD5 and SHA-1 Hash Values for the Original and Slightly Modified Passage from the Declaration of Independence

	*“**We** hold these truths to be self-evident...”*	*“**I** hold these truths to be self evident...”*
MD5	85e1187eab57b877a7f1584724dca1f8	5843aaf98e8ded90ce8fafc41d9206b0
SHA-1	faea9c76bb42581f5e234f78ef4d28a756395270	deb3f16c13b8a10ffdd8b47162f229f88c749674

Table 5 Base64 Encoded SimHash Values for the Original and Slightly Modified Passage from the Declaration of Independence

Line	*“**We** hold these truths to be self-evident...”*
1	ywEACK8oVwiuNc2Kq8EvdKtZqy6qpA9FqqI8/ajtb8GmiTCTpD6P6qQFdwmh2TfwodNBbJ+RPYif
2	WInVneZiHp2mAbScWWIwm20SmZdnNjSWz0yKlMX7SpSrFXiTgmIVkirAkY15Q8uNIdOvjGLe6om7
3	AnCIjI2sg3FKUIKKCKt/xTH+f7HDmH+pbpJ+oiSdfcqpnXwMQtB7tyzRewHHuXnK1BR4c5+IeB8n
4	knflkDt3fvJodsiMBnbDqXx0093FdKA633OoiE5wk0wZb+FGfG6YFaRs4sRuaLlVPmU7ONZkkrB/
5	Y0iu7mL34rJhE8a6YJk1E13OFyVX7pjQVv9m9VV5iLdVJab+UVd1MFEpfgdOSqQVTTGyQkyUTY1L
6	Qx2USpn770hlv/ZH/8f5R/KnN0W5yBRETFgLQ4eyW0EMjSBAUtNqPmEIiz0JRIo6XrCSOGrSpzMS
7	2XozBuDXMsid/TJBiMQxF56dLuJmbSySGCcsSV4cKqFsSSl4sislTtu4JUde3yMXhX0iDPwyIgiB
8	zyCs9tYgolDLHiIM4h1VNxcceFD7GvjwoBruGcAaOHTQGfH6QBkxMccYLUT1F/h+sBTNHtUUDC3e
9	E5AC7hIeg6IQAtqNDbfBqA2XM4ANYTBCDUMt7Az7d5MK
10	myRqB3u4kQdas2wBz7yQAM8UjABv030AMaei

Line	*“**I** hold these truths to be self-evident...”*
1	ywEACLBEeCmvKFcIrjXNiqvBL3SrWasuqqQPRaqiPP2o7W/Bpokwk6Q+j+qkBXcJodk38KHTQWyf
2	kT2In1iJ1Z3mYh6dpgG0nFliMJttEpmXZzY0ls9MipTF+0qUqxV4k4JiFZIqwJGNeUPLjSHTr4xi
3	3uqJuwJwiIyNrINxSlCCigirf8Ux/n+xw5h/qW6SfqIknX3KqZ17tyzRewHHuXnK1BR4c5+IeB8n
4	knflkDt3fvJodsiMBnbDqXx0oDrfc6iITnCTTBlv4UZ8bpgVpGzixG5ouVU+ZTs41mSSsH9jSK7u
5	YvfismETxrpgmTUTXc4XJVfumNBW/2b1VXmIt1Ulpv5RV3UwUSl+B05KpBVNMbJCTJRNjUtDHZRK
6	mfvvSGW/9kf/x/lH8qc3RbnIFERMWAtDh7JbQQyNIEBS02o+YQiLPQlEijpesJI4atKnMxLZejMG
7	4NcyyJ39MkGIxDEXnp0u4mZtLSrKNiySGCcsSV4cKqFsSSl4sislTtu4JUde3yMXhX0iDPwyIgiB
8	zyCs9tYgolDLHiIM4h1VNxcceFD7GvjwoBruGcAaOHTQGfH6QBkxMccYLUT1F/h+sBTNHtUUDC3e
9	E5AC7hIeg6IQAtqNDbfBqA2XM4ANYTBCDUMt7Az7d5MK
10	myRqB3u4kQdas2wBz7yQAM8UjABv030AMaei

change to the original meaning we modify *“We hold these truths to be **self-evident**...”* to be *“We hold these truths to be **self evident**...”* we see the Simhash value drop slightly to 0.91. It is easy to imagine that our small modification of the original text neither changes the title of the document nor affects the lexical signature of the document. As mentioned in Chapter I, Section 2.1, hash functions are one way of implementing the reduced representation function rr. However, due to several reasons mentioned earlier we chose alternate implementations such as lexical signatures and titles.

8 WEB CONTENT CORPORA

Numerous text corpora are available for information retrieval research. A small sample of corpora and their characteristics can be found in Table 6. These corpora are generally considered a representative sample for the Internet [242] but have also been found to be somewhat dated [84]. The *TREC Web Track* is probably the most common and well known corpus in information retrieval research. It is based on a crawl of English language web pages and has, for example, been used in [247] for DF estimation. The *British National Corpus (BNC)* [176] is not based on web page content but on miscellaneous written documents and transcripts of verbal communication such as phone calls and interviews of British origin. It has been used for [245].

Table 6 Available Text Corpora Characteristics

Corpus	**Google N-Gram**	**TREC WT10g**	**BNC**	**WaC**
Source	Google indexed English language Web Pages	English language Web Pages	British English Texts (newspapers, journals, books) Transcripts of Verbal Language (meetings, radio shows)	uk.Domain Web Pages
Date	2006	1997	1994	2006
Unique Terms	$> 13M$	$5.8M$[167]	N/A $> 100M$ Total Terms	$> 10M$
Number of Documents	$> 1B$ (Not Available)	$1.6M$ (Available)	$4,124$ N/A	$> 2.6M$ (Available)
TC	Available	Not Available	Available from 3^{rd} Party	Available
Freely Available	No[a]	No	No	Yes

[a] A limited number of free copies of the corpus are available from the Linguistic Data Consortium, University of Pennsylvania

The *Google N-Grams* [113] were published in 2006. This corpus is based on the Google index and therefore provides a powerful alternative to the corpora mentioned above. Not only does it contain the most amount of unique terms it also is based on the most amount of documents.

The *Web as Corpus kool ynitiative (WaCky)* [42] provides another corpus based on web pages. Their *Web as a Corpus (WaC)* is freely available to everyone and, similar to the Google N-Grams, contains a huge number of unique terms. It was also created fairly recently and it is based on a crawl of the *.uk* domain.

CHAPTER IV

TF-IDF VALUES FOR THE WEB

1 BACKGROUND

As shown in Chapter III, Section 8, various corpora containing textual content of web pages are available to researchers. Corpora such as the *TREC Web Track* and the *BNC* provide all documents they are based on. Researchers can therefore precisely determine TF and DF values for all terms they contain. The DF value is represented as $|d_i|$ in Equation 8 of Chapter II, Section 5. Since the sum of documents in the corpus ($|D|$ in the same Equation 8) can easily be computed both mandatory values for the IDF computation are given. Since both corpora are not freely available we turn our focus to the *Google N-Grams.* This corpus provides all unique terms (or n-term tokens) of the corpus along with the count of their overall occurrences in the entire corpus. We call this value *term count (TC).* The N-Grams do not provide the documents they are based on which means we can not determine DF values for the terms. To illustrate the difference between TC and DF, let us consider a small sample corpus of 5 documents $D = d_1...d_5$ where each document contains the title of a song by The Beatles:

$d_1 = Please\ Please\ Me$
$d_2 = Can't\ Buy\ Me\ Love$
$d_3 = All\ You\ Need\ Is\ Love$
$d_4 = All\ My\ Loving$
$d_5 = Long,\ Long,\ Long$

Table 7 summarizes the TC and DF values for all terms occurring in our corpus. We can see that the values are identical for the majority of the terms (8 out of 10). The example also shows that term processing such as stemming would have an impact on these numbers. Without stemming *Love* and *Loving* are treated as different terms with stemming they both would transform to *Lov.* The *WaC* also provides TC values of all their unique terms. Unlike the Google N-Grams all documents are provided as well and so we can determine DF values as well. Due to the size of the corpus and the documents it is based on (the Google index) we would prefer to use the N-Grams for computing our DF values. However, from personal conversations with a Google representative we know that Google does not intend to publish the DF values of their corpus. But given these two corpora we can investigate the correlation between TC and DF values. With a positive outcome we could say

Table 7 *TC-DF* Comparison Example

Term	All	Buy	Can't	Is	Love	Me	Need	Please	You	My	Loving	Long
TC	2	1	1	1	2	2	1	**2**	1	1	1	**3**
DF	2	1	1	1	2	2	1	**1**	1	1	1	**1**

with confidence that the TC values provided by the N-Grams can be used interchangeably with DF values from the WaC and therefore are suitable to estimate DF values for our lexical signatures. We achieve this by investigating the relationships between TC and DF values within the WaC and also between WaC based TC and Google N-Gram based TC values as shown in Klein and Nelson [159].

2 CORRELATION OF TERM COUNT AND DOCUMENT FREQUENCY VALUES

2.1 Correlation Within the WaC

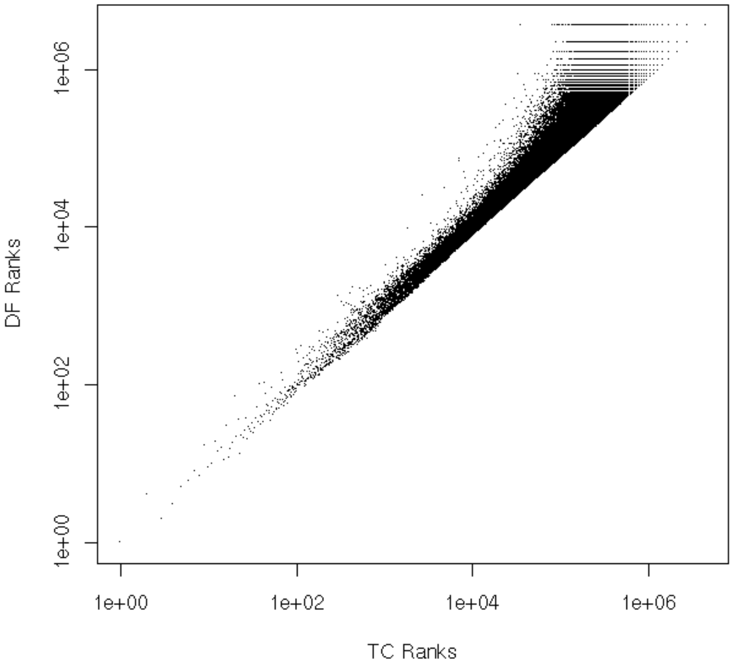


Fig. 13 Term Count and Document Frequency Ranks in the WaC Dataset

We take the WaC dataset and rank all its terms according to their TC and DF values in descending order. The scatterplot of the ranks from these two lists is displayed (in log-log scale) in Figure 13. The x-axis represents the TC ranks and the y-axis the corresponding DF ranks. We see the majority of the points fall in a diagonal corridor. That indicates a high similarity between the rankings since two identical lists would be displayed as a perfect diagonal line. Figure 14 shows the measured and estimated correlation between TC and DF values in the WaC dataset. The increasing size of the dataset, meaning the increasing list of terms, is shown on the x-axis. The solid black line displays the Spearman's ρ values. The value for ρ at any size of the dataset is above 0.8 which again indicates a very strong correlation between the rankings. The results are statistically

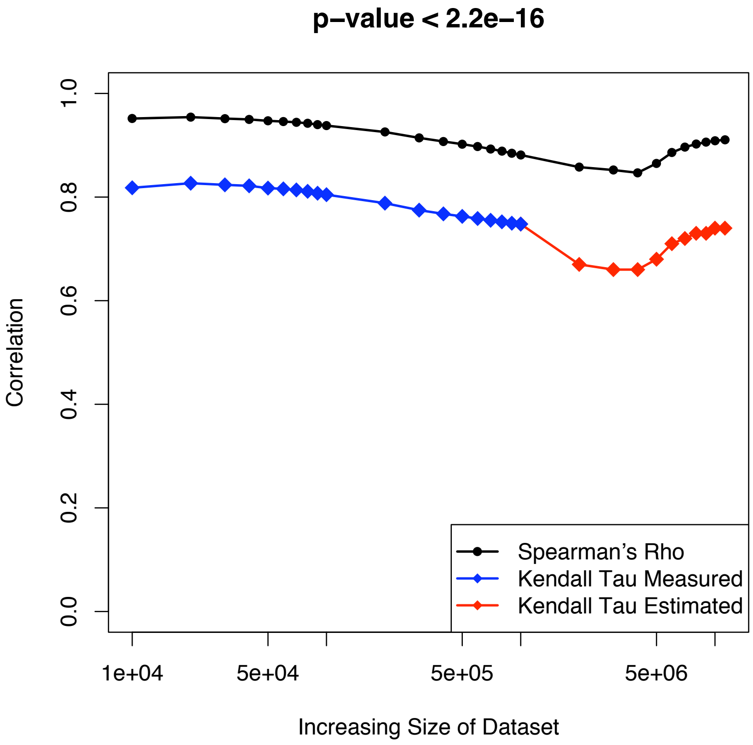
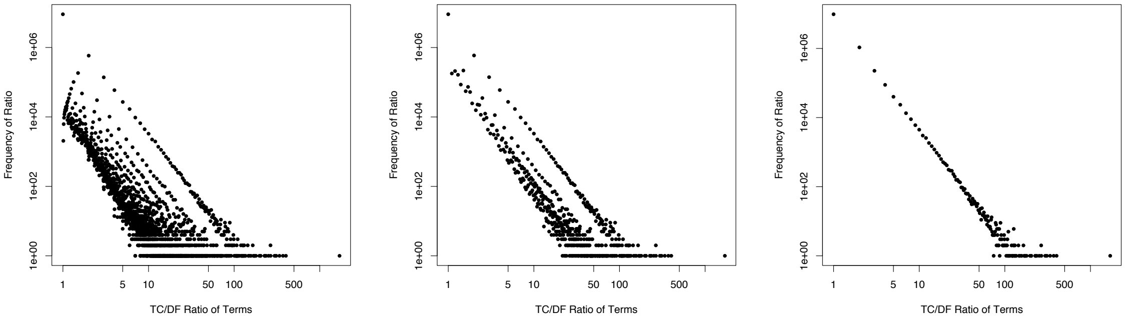


Fig. 14 Measured and Estimated Correlation Between Term Count and Document Frequency Ranks

significant with a p-value of $p \leq 0.05$. The blue line shows the *computed* Kendall τ values for the top $1,000,000$ ranks and the red line represents the *estimated* τ values for the remaining ranks. We again find a strong correlation with computed τ values between 0.82 and 0.74 and estimated τ values of at least 0.66. We did not compute τ for greater ranks since it is a very time consuming operation. For the entire WaC dataset (over 11 million unique terms) we estimated a computation time for Kendall τ of almost 11 million seconds or more than 126 days which is clearly beyond a reasonable computation time for a correlation value. Kendall τ was computed using an off-the-shelf correlation function as part of the R-Project [36], an open source software environment for statistical computing. The software (version 2.6) was run on a Dell Server with a Pentium P4 2.8Ghz CPU and 1 GB of memory.

Gilpin [116] provides a table for converting τ into ρ values. We use this data to estimate our τ values. Even though the data in [116] is based on τ values computed from a dataset with bivariate normal population (which we do not believe to have in the WaC dataset), it supports our measured values.



(a) Two Decimals

(b) One Decimal

(c) Integer Values

Fig. 15 Frequency of TC/DF Ratios in the WaC – Rounded

2.2 Term Count – Document Frequency Ratio in the WaC

We further emphasize the correlation between the two rankings by plotting the TC/DF ratios of all terms. For two ranked lists which are perfectly correlated the ratio for all list items is equal to 1. Figure 15 shows (in log-log scale) the frequency of all ratios. The point with by far the highest frequency is at $x = 1$ which confirms the dominance of the "perfect ratios". Figure 15(a) shows the distribution of TC/DF ratios with values rounded after the second decimal and Figure 15(b) shows ratios rounded after the first decimal. It is clearly visible that the vast majority of ratio values are close to 1. The visual impression is supported by the computed mean value of 1.23 with a standard deviation of $\sigma = 1.21$ for both Figure 15(a) and 15(b). The median of ratios is 1.00 and 1.0 respectively. Figure 15(c) shows the distribution of TC/DF ratios rounded as integer values. It is consistent with the pattern of the previous figures. The mean value is equally low at 1.23 ($\sigma = 1.22$) and the median is also 1. Figure 15 together with the computed mean and median values accounts for another solid indicator for the strong correlation between TC and DF values within the corpus.

2.3 Similarity Between WaC and N-Gram TC Values

After showing the correlation between TC and DF values within the WaC, we investigate the similarity between the TC values available from both corpora, WaC and the Google N-Grams. Since both corpora are based on different sources and the N-Gram dataset was generated from a much greater set of documents, a direct comparison of intersecting terms could be misleading. However, a comparison of the frequency of all TC values of the two corpora will give an indication of the similarity of the two datasets. Figure 16 displays in log-log scale these frequencies of unique TC values from both corpora. Visual observation of the figure confirms the intuition that the distribution of TC values in both corpora is very similar. The assumption is that only the greater size of the Google N-Gram corpus compared to the WaC is responsible for the offset between the points in the graphs. Figure 16 further shows the TC threshold of 200 that Google applied while creating the N-Gram dataset meaning that unigrams occurring less than 200 times in their document set were dismissed. Now, knowing that the TC values are very similar between the two corpora and

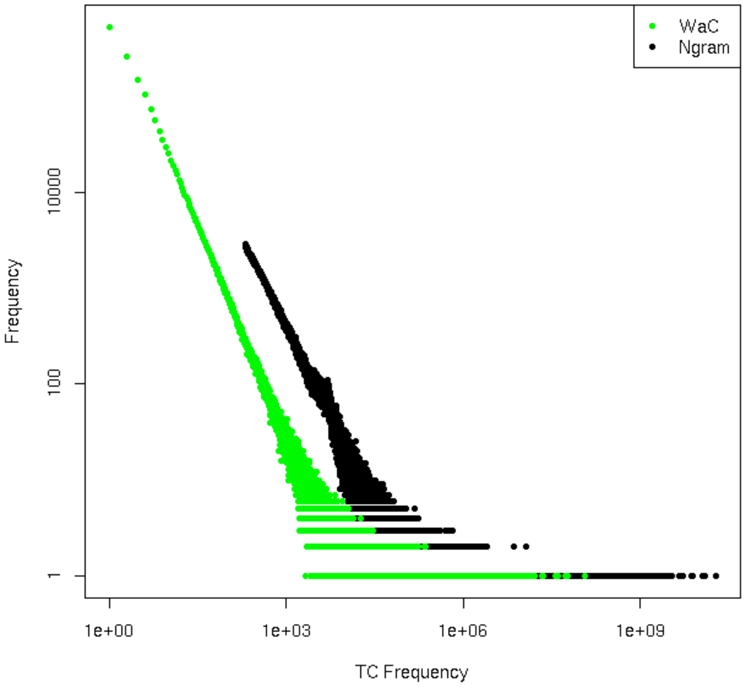


Fig. 16 Term Count Frequencies in the WaC and N-Gram Corpus

seeing the high correlation between TC and DF values, we are led to the conclusion that the Google N-Gram (TC) values are usable for accurate IDF computation. These results are also supported in related research conducted by others [151, 206, 275].

The TREC Web Track has been used to compute IDF values in [247] and the BNC in [245] but it is unknown whether the authors actually computed DF values from the corpora or used TC values, which in case of the BNC are available through a 3^{rd} party.

3 ESTIMATING IDF VALUES TO GENERATE LEXICAL SIGNATURES FOR THE WEB

The Google N-Grams provide an attractive corpus for the following reasons:

1. it is created by Google, a well established source
2. the size of the collection seems suitable
3. the collection was created recently (2006)
4. and it is based on web pages from the Google index and thus representative of what we generate lexical signatures from.

However, Google's terms of use prohibit us from utilizing the corpus in any kind of service or public software. But since we have seen that the N-Grams are suitable for generating accurate IDF values we can use the corpus as a baseline and compare other potential IDF value generation methods against it.

We validate two different approaches against this baseline:

1. a local collection of web pages and

2. "screen scraping" document frequency from search engine result pages.

This comparison determines whether the two alternative methods return results that are sufficiently similar to our baseline and thus warrant use for (re-)discovering web pages. We evaluate the results by analyzing the top n terms ranked by their TF-IDF value generated by the three different means. The analysis consists of measuring the term overlap and the correlation of the list of ranked terms as shown in Klein and Nelson [157].

3.1 Choice of Baseline Corpus

The Google dataset provides TC values for n-grams where $n = 1$ to $n = 5$. In case of $n = 1$ the tokens are called unigrams and bigrams in the case where $n = 2$. The number of documents Google used in the compilation of the N-Grams was not published and thus not available to us at the time we conducted our experiment. Since the N-Gram dataset is based on the Google index we use the last officially publicized Google index size of 8 Billion (in 2005) for the value of $|D|$ for the IDF computation. We are aware that this is a rather conservative estimate but since all major search engines have discontinued publishing their index sizes we trust in the last officially announced numbers. Other (unofficial) sources like `http://www.worldwidewebsize.com/` report much larger numbers and a case can be made to use their experiments in future work (even though, mathematically speaking, it would not make a difference).

The dataset only reports n-grams that appear at least 200 times. For terms that do not appear in the N-Gram dataset, we optimistically assume they just missed the threshold for inclusion and give them a value of 199. Since we compute IDF values of single terms we use all 1-term tokens from the N-Gram set and refer to the resulting data from the baseline as *N-Gram (NG)* data.

3.2 Local Universe Data

Even though for the computation of IDF values the intuitive approach would be to download all existing web pages, parse their content and compute the document frequency of all terms, this is clearly not feasible. However, we can download a sample set of web pages and all their copies from the last N years to create a "local universe" with document frequencies available for all terms.

We randomly sampled 300 URIs from `dmoz.org` as our initial set of URIs. From this pool we selected only the more common domains .com, .org, .net and .edu similarly to the filters applied in Park et al. [214] we dismiss:

- all non-English language websites and

- all websites with less than 50 words of textual content (HTML code excluded).

Our set of URIs eventually contains 78 .com, 13 .org, 5 .net and 2 .edu URIs for a total of 98.

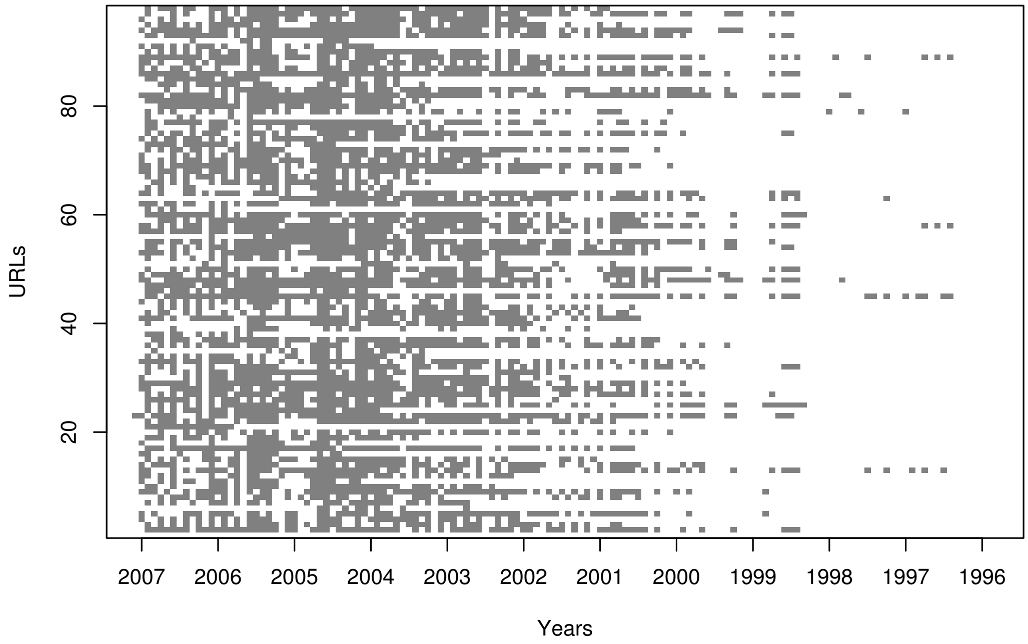


Fig. 17 Mementos from the Internet Archive from 1996 to 2007

We used the IA to download all available copies for each of our 98 URIs. Our local universe consists of a total of $10,493$ Mementos, each identified by a URI (U) and the timestamp (t) it was archived. Note that at the time we conducted this experiment the Memento framework was not yet deployed and we therefore used the IA as our only resource for previous versions of the URIs. Despite that, for reasons of consistency, we use the term Memento for the here obtained copies. The model of the corpus is shown in equation 30.

$$local\ universe = \left\{ \begin{array}{llll} U_{1,t_1} & U_{1,t_2} & ... & U_{1,t_n} \\ U_{2,t_1} & U_{2,t_2} & ... & U_{2,t_n} \\ ... & & & \\ U_{98,t_1} & U_{98,t_2} & ... & U_{98,t_n} \end{array} \right\} \tag{30}$$

Figure 17 shows all (in September 2007) downloaded Mementos of all 98 URIs in this 12 year span. The date of the Mementos is shown on the x-axis and the URIs, alphabetically ordered and numbered, along the y-axis. Within the 12 year time span we only see a few Mementos in the early years of 1996 and 1997. The graph becomes more dense however from the year 2000 on. The first Mementos date back to December 1996 and the latest in September 2007. Figure 17 shows an interesting fact: at any given point in time at least one of the URIs does not have a Memento or, in other words, at no point in time do we have Mementos for *all* our sample URIs. We extracted all terms from each Memento and stored them in a local database. We do not include the content of neighboring pages at this point. This term database consequently consists of all unique terms that appear in any of the 98 web pages within the 12 year time span and the number of documents they appear in. Thus the database represents the union of the textual content of all our Mementos and the document frequency of all terms. With the frequencies and the total number of documents in our universe we have both mandatory values available to compute IDF values for all terms. We call the data that results from this computation *locally computed (LC)* data.

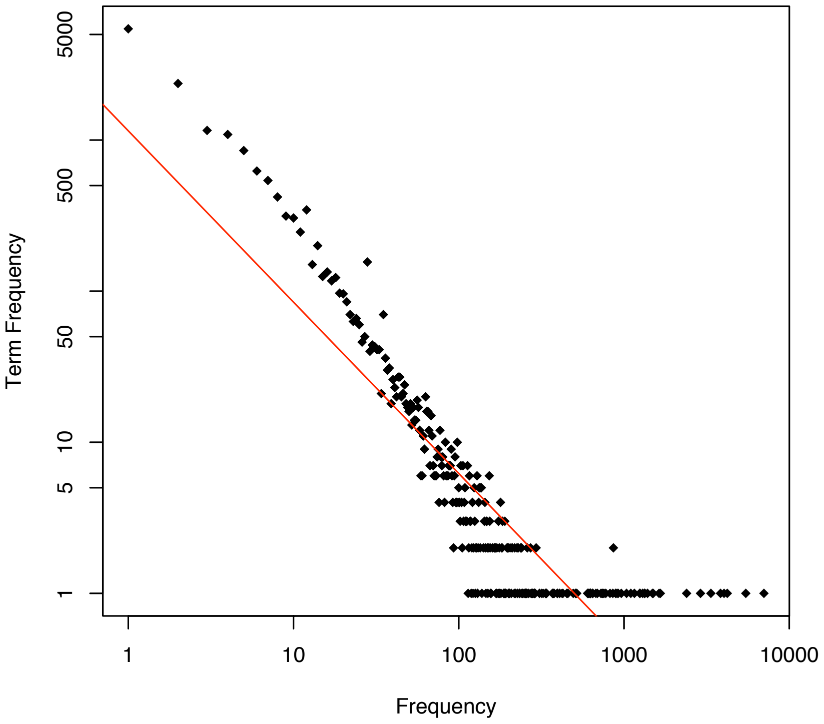


Fig. 18 Term Frequency Distribution in the Local Universe

Figure 18 confirms that the term distribution in our local universe (despite its limited size) follows a Zipf distribution [49] of $P_k = c \times (k^{-a})$ where $a = 1.134$, similar to the distribution of English language terms. There are a total of $254,384$ terms with $16,791$ unique terms. Figure 19 shows the development of these two numbers over time where the left y-axis represents the numbers for the total terms and the right y-axis shows the numbers for the unique terms. We also computed the number of new terms per year normalized by the total number of Mementos in the respective year. The values range from 87.7 to 177.5 with a mean of 131.7 and a standard deviation $\sigma = 24.3$.

It is worth mentioning that we also generated term databases for each and every single year separately and computed IDF values for each year in respect to the according database. We found however that the results for the per year generated IDF values were very similar to the values

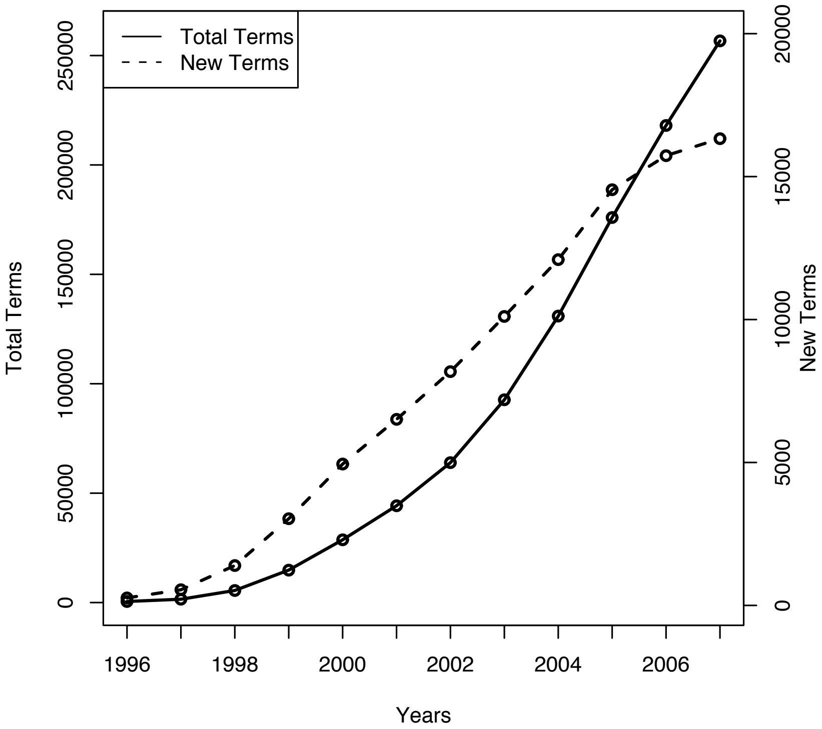
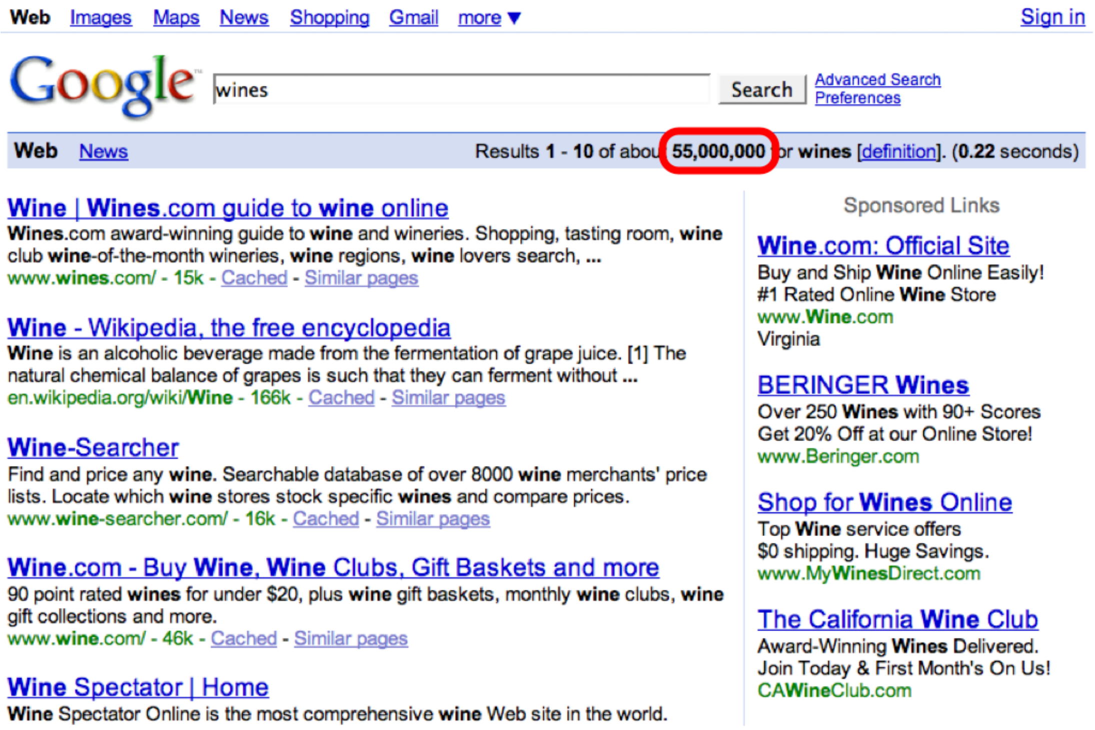


Fig. 19 New vs Total Number of Terms

generated based on the LC data and thus we do not report on them separately.

3.3 Screen Scraping the Google Web Interface

A very common approach ([120, 180, 214, 218]) to compute IDF values of web page content has been to glean document frequency information from search engine result pages. This approach is known as “screen scraping” and can be done in real time which makes it attractive for many scenarios. In this approach, every single term is queried against a search engine and the “Results”-value is used as the document frequency for that term. Figure 20 shows an example of the Google result set (from May 2008) for the query term *wines* and the estimated number of documents the term occurs in: $55,000,000$. Although the document frequency information is only estimated [18], this is the only information available to us. Our screen scraping data was generated in January of 2008 and we use the same value for $|D|$ as in Section 3.1. We refer to this data as *screen scraping data (SC).*



Results **1** - **10** of abou **55,000,000** or **wines** [definition]. (**0.22** seconds)

Wine | Wines.com guide to wine online
Wines.com award-winning guide to **wine** and wineries. Shopping, tasting room, **wine** club **wine**-of-the-month wineries, **wine** regions, **wine** lovers search, **...**
www.**wines**.com/ - 15k - Cached - Similar pages

Wine - Wikipedia, the free encyclopedia
Wine is an alcoholic beverage made from the fermentation of grape juice. [1] The natural chemical balance of grapes is such that they can ferment without **...**
en.wikipedia.org/wiki/**Wine** - 166k - Cached - Similar pages

Wine-Searcher
Find and price any **wine**. Searchable database of over 8000 **wine** merchants' price lists. Locate which **wine** stores stock specific **wines** and compare prices.
www.**wine**-searcher.com/ - 16k - Cached - Similar pages

Wine.com - Buy **Wine**, **Wine** Clubs, Gift Baskets and more
90 point rated **wines** for under $20, plus **wine** gift baskets, monthly **wine** clubs, **wine** gift collections and more.
www.**wine**.com/ - 46k - Cached - Similar pages

Wine Spectator | Home
Wine Spectator Online is the most comprehensive **wine** Web site in the world.

Sponsored Links

Wine.com: Official Site
Buy and Ship **Wine** Online Easily!
#1 Rated Online **Wine** Store
www.**Wine**.com
Virginia

BERINGER **Wines**
Over 250 **Wines** with 90+ Scores
Get 20% Off at our Online Store!
www.Beringer.com

Shop for **Wines** Online
Top **Wine** service offers
$0 shipping. Huge Savings.
www.My**Wines**Direct.com

The California **Wine** Club
Award-Winning **Wines** Delivered.
Join Today & First Month's On Us!
CA**Wine**Club.com

Fig. 20 Google Result Set for the Query Term *wines*

3.4 Comparison Methods for TF-IDF Ranked Term Lists

Since the TF computation is done the same way in all three approaches (including the baseline), we compare lists of terms ranked by their TF-IDF (and not only IDF) value in decreasing order. We transform the lists of terms ranked in descending order by their TF-IDF values into 5-, 10- and 15-term lexical signatures. Table 8 shows an example of terms in a lexical signature and the according TF-IDF values. We took the textual content from the URI `http://www.perfect10wines.com` in 2007 and list the top 10 terms in decreasing order of their TF-IDF values for each of our three approaches. All three lists of table 8 contain 10 terms but despite different TF-IDF values we only see 12 unique terms in the union of all lists. From these lists one could create a lexical signature for the given website. The 5-term lexical signature generated by the screen scraping method, for example, would be *wines, robles, perfect, paso, wine.* This example also shows that we did not apply stemming algorithms because otherwise *wines* and *wine* would be treated as one entity.

To measure the differences between these lexical signatures we use three different comparison methods which have been proven in related research [192] to be efficient in comparing top x search engine results. The first method we apply is normalized term overlap. Since the idea is to feed the lexical signatures back into search engines, term overlap is an important metric. For simplicity, we

Table 8 Top 10 TF-IDF values generated from `http://www.perfect10wines.com`

	Local Universe		Screen Scraping		N-Grams	
	Term	**TF-IDF**	**Term**	**TF-IDF**	**Term**	**TF-IDF**
1	perfect	7.77	wines	5.97	wines	7.56
2	wines	6.95	robles	5.3	perfect	7.25
3	10	6.57	perfect	4.35	robles	7.18
4	paso	6.29	paso	4.27	paso	6.93
5	wine	6.18	wine	3.26	wine	4.86
6	robles	5.4	sauvignon	3.16	10	4.52
7	sauvignon	3.54	chardonnay	3.15	chardonnay	3.99
8	cabernet	3.54	robles84	3.11	sauvignon	3.93
9	monterey	3.36	cabernet	3.09	cabernet	3.89
10	chardonnay	3.36	enthusiast85	2.91	monterey	3.49

assume query term commutativity, although in practice the order of query terms can slightly change the results based on proximity in the original document and other factors. We normalize the overlap of k-term lists by k. Let us, for example, assume list α contains three terms $\alpha = [a, b, c]$ and list β contains $\beta = [b, c, d]$. The normalized overlap would be 2/3 since α and β share two out of three terms.

In a scenario with m total terms (all candidates to make it into the lexical signature) and an k-term lexical signature, there is value in knowing whether a term is ranked $k+1$, meaning it has just missed the top k, or whether it is ranked m or $m-1$, meaning in the lower end of the ranked list of all terms. The overlap value will not reveal this information and always leave us wondering if the performance would be better if we chose a bigger or even a smaller k. For this reason we also use a modified version of Kendall τ correlation introduced by Fagin et al. [104]. This measure will answer the question since we compare lexical signatures of different length (5, 10 and 15 terms). The common Kendall distance measure can not be applied here since its general assumption of both lists having all items in common can not be guaranteed. The modified version of Kendall does not require both lists to share all items.

Since our lexical signatures start with the top k terms, we also think there is value in giving more credit to lists that are very similar in the high ranks and maybe less similar at the lower end than lists that show the opposite pattern. We use the M-measure as our third correlation method. The M-Score will provide information about the locality of the term ranks in the lexical signatures. A low score means the compared lexical signatures show discordance in the high ranks and a high score stands for concordance in the top ranks.

All three methods are normalized and return scores between 0 and 1 where 0 means complete disagreement and 1 complete agreement.

3.5 Results

With our three different models for IDF generation (*LC, SC, NG*) and the three different comparison and similarity measures between ranked lists of terms in decreasing TF-IDF order we can report on all together nine comparisons. Since we consider the *NG* based data our baseline, the comparison

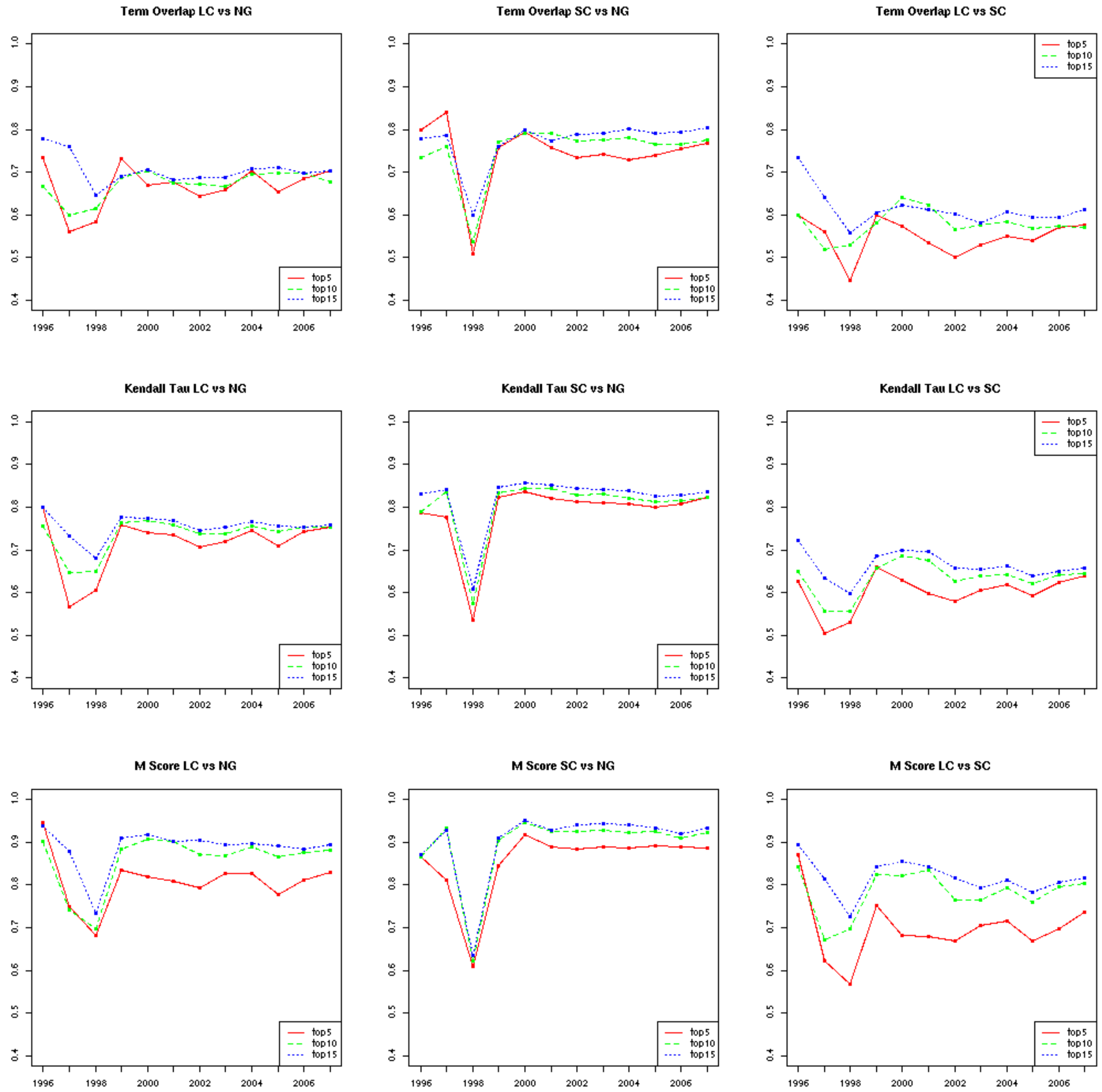


Fig. 21 Term Overlap, Kendall τ and M-Score of All Three Lexical Signature Generation Methods

between LC and SC is included just for completeness. It is more important for us to see how LC and SC compare against NG. Figure 21 shows all comparison and correlation scores of all 98 URIs over time. The progress in time (which is due to the temporal character of our local universe) is displayed on the x-axis and the y-axis shows the appropriate values as the mean of all 98 URIs for a particular year. The first horizontal line of graphs holds the normalized term overlap scores, the second the Kendall τ scores and the third the M-Scores. The first vertical column of the plots shows the comparison values between LC and NG based data, the middle one shows SC and NG based data comparison and the graphs in the rightmost column display comparison values between LC and SC based data. The three lines visible in each of the plots represent lexical signatures consisting of 5, 10 and 15 terms respectively.

We can observe that all nine plots look fairly similar with the lines being somewhat out of tune in the early years and from approximately year 2000 on the scores become more and more consistent with values of well above 0.5 for all three top k lexical signatures. The noise in the early years can be explained with our sparse dataset in these years. Over time, as the dataset grows, the scores level off. We can also see the highest correlation in terms of overlap, Kendall τ and M-Score between the SC and NG based data. This is not surprising since these two datasets are supposedly based on the same (Google) index which exceeds the size of our local index (LC based data) by several orders of magnitude. It furthermore becomes visible that the similarity between LC and NG is greater than between LC and SC. We have two explanations for this observation. First, we argue that data returned by screen scraping is not as accurate as what Google reported in their N-Grams since the Google web interface only returns estimated values for number of documents a term appears in. Second, we do have a frequency for every single term in the NG dataset (due to Google's threshold we assign a value of 199 to all terms that do not appear in the N-Grams) but there are cases where the Google web interface does not return a value for certain terms which we consequently can not include in the lexical signature computation. For example, one URI contains the term *companyproductsfaqorder* which queried against the Google web interface returns no results[1].

As mentioned above we consider the normalized term overlap displayed in the top row of Figure 21 as a very important similarity measure. After the initial noise in the early years we see an overlap of 80% in the best case (SC vs NG), about 70% in LC vs NG comparison and a worst case of 50% comparing LC and SC. The scores for Kendall τ (middle row of Figure 21) are even higher which means that not only the term overlap is good in the lexical signatures that are compared but also the order of the terms is fairly similar meaning the number of pairwise disagreements within the top k lexical signatures and thus the number of switches needed to transform one lexical signature into the other is rather low.

The M-Score (displayed in the bottom row of Figure 21) accounts for the highest score in this comparison. The best cases report scores of 0.9 and above. This graph confirms that the disagreements in the lexical signatures are rather at the low end of the ranked list of terms regardless of the method the IDF values were computed with. The M-Score is the only comparison where a slight difference between top k lexical signatures becomes visible. Top 5 lexical signatures seem to perform slightly worse than top 10 and top 15 especially in the comparisons LC vs NG and LC vs SC. This

[1] This is not quite accurate since the query term does return one result: the dissertation proposal published on this author's website.

means the TF-IDF methods become more correlated as k increases.

All methods shown in Figure 21 comparing LC and NG as well as SC and NG data show that both the local universe based data as well as the screen scraping based data is similar compared to our baseline, the N-Gram based data. The presented scores seem to imply that the agreement between the methods for IDF computation improves or at least stays the same as we chose a greater number of terms.

3.6 Inter-Search Engine Lexical Signature Performance

We have shown that screen scraping a search engine's web interface to estimate a term's DF value is a feasible approach. It provides accurate results and can be done in real time.

The purpose of this small scale experiment was to investigate the relationship between a search engine's DF value and the retrieval performance of the lexical signatures based on those DF values using the very same search engine. We are trying to answer the question whether we can improve retrieval values by using DF values from one search engine and querying the lexical signatures against a different search engine as shown in Klein and Nelson [160].

We use Google, Yahoo! and MSN (Bing had not yet been introduced at the time we conducted this experiment) for this approach. We used the 309 URIs from the dataset introduced below in Chapter V, Section 3.2 and generated three lexical signatures for all of those URIs. For each URI we eventually have a Google based lexical signature, a Yahoo! based lexical signature and a MSN based lexical signature. We then cross-query all lexical signatures against the two search engines they were not generated from using the corresponding APIs. Since all of the sample URIs are from the live web and indexed by all of the search engines we should be able to locate them in the result sets. We therefore can compare the retrieval performance of lexical signatures across search engines.

Figure 22 shows the 5-term Lexical Signature performance in all search engines. The labels on the axes indicate what search engine the lexical signatures were derived from (first letter) as well as what search engine they were queried against (second letter). G, M and Y stand for Google, MSN and Yahoo! respectively. The label GM, for example, represents lexical signatures based on Google and queried against MSN. The size of the dots is proportional to the number of URIs returned. We distinguish between one of four retrieval categories:

1. the URI is returned as the top ranked result or

2. the URI is returned in the top 10 but not as the top ranked result or

3. the URI is ranked somewhere between 11 and 100 or

4. the URI was not returned which means in our case is ranked somewhere beyond rank 100.

The categories are further argued for in Chapter V, Section 3.1. The absolute values are also plotted in the graph either inside or right next to the corresponding dot. MSN based lexical signatures perform better when queried against Yahoo! or Google than against MSN itself. They return more top ranked URIs (MY), more URIs in the top 10 and top 100 (MG) and leave fewer URIs undiscovered in both scenarios. Yahoo! and Google based lexical signatures perform best when queried against the search engine they were generated from. Even though YG returns almost twice as many URIs in the top 10 than YY its performance in the top ranks is much worse.

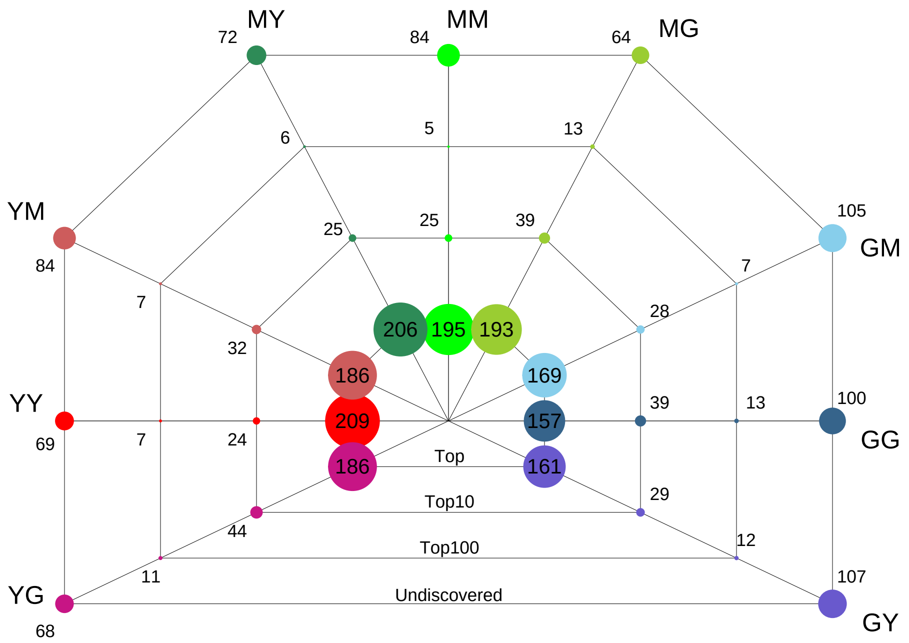


Fig. 22 Lexical Signature Performance Across All Three Search Engines

4 SUMMARY

In this chapter we have shown that term count (TC) values can be used to estimate the document frequency of a given term for the purpose of computing the term's TF-IDF value. This conclusion is derived from two of our findings. First we see a very strong correlation between the TC and DF ranks within the WaC. Second we find a great similarity between TC values of terms in the WaC and the Google N-gram corpus despite the difference in corpus sizes. These results do not provide a bulletproof case that all values correlated to TC (TF may be an example) can automatically be used as a replacement for DF but they do give strong indicators that TC values gained from the Google N-grams (a recently generated, on web pages based corpus) are usable for the generation of accurate IDF values. With this finding the computation of accurate IDF values becomes more convenient and the resulting lexical signatures of web pages still perform well for (re-)discovering the page when fed back into search engines.

However, since the Google N-grams can only be used for research purposes we can not utilize them for our software implementation. Therefore we investigated two additional methods for computing IDF values for web pages. We used the Google N-grams as a baseline to evaluate the performance of the two methods. The first method is based on a local universe meaning a set large of web pages that, considered as a corpus, can provide DF values. The second method is known as screen scraping the search result page from a search engine. Even though neither of these methods is new to the community, the comparison of the two against a baseline is novel. The results in this chapter show a great similarity between the two methods and the baseline. In particular, all three similarity measures, term overlap, Kendall τ and the M-Score support this finding, especially for the screen scraping based data. The scores for local universe based data are still good but slightly below the screen scraping values. Another interesting finding is that it appears with the increasing number of terms for a lexical signature the agreement between the methods also increases or at the very minimum remains stable. In conclusion, both approaches provide good correlation with data derived from the Google N-grams and are suitable for our purpose. However, screen scraping has the major advantage of being feasible in real time and it can be implemented in our software system. Including a local universe into our software does not seem realistic.

CHAPTER V

LEXICAL SIGNATURES FOR THE WEB

1 BACKGROUND

A lexical signature of a textual document is typically generated using the TF-IDF scheme. To generate lexical signatures of web pages the pages of interest are usually filtered by language, mime type and length. Pages containing less than 50 words have been dismissed in related research [214] to ensure a good sized body of text to extract a lexical signature from. Naturally such a signature is language dependent and therefore non-English content is also dismissed. Furthermore, non-HTML resources, for example, PDF or Microsoft Word documents, are discarded. Several techniques are feasible to enforce this filter. It is possible (but not mandated by RFC 2616 [108]) that the HTTP server provides an HTTP header specifying the content type of the resource. In case this information is not provided or there is a need to validate the provided data, third party software such as the UNIX *file* command line tool can be used to provide a best guess about the content type of the investigated document as shown in [264].

Once these hurdles are overcome all terms of the page are extracted into a "bag of words". We compute TF-IDF values for all terms and sort the terms in decreasing order of their TF-IDF value. The top n terms of that list form the lexical signature of a document and therefore represent our first implementation of the rr function introduced in Chapter I. The main purpose of a lexical signature is to be used as a query against a retrieval system such as an Internet search engine and retrieve the same document in the result set as well as other highly relevant documents. This concept consequently follows the $c2u$ function also formally introduced in Chapter I.

2 EVOLUTION OF LEXICAL SIGNATURES OVER TIME

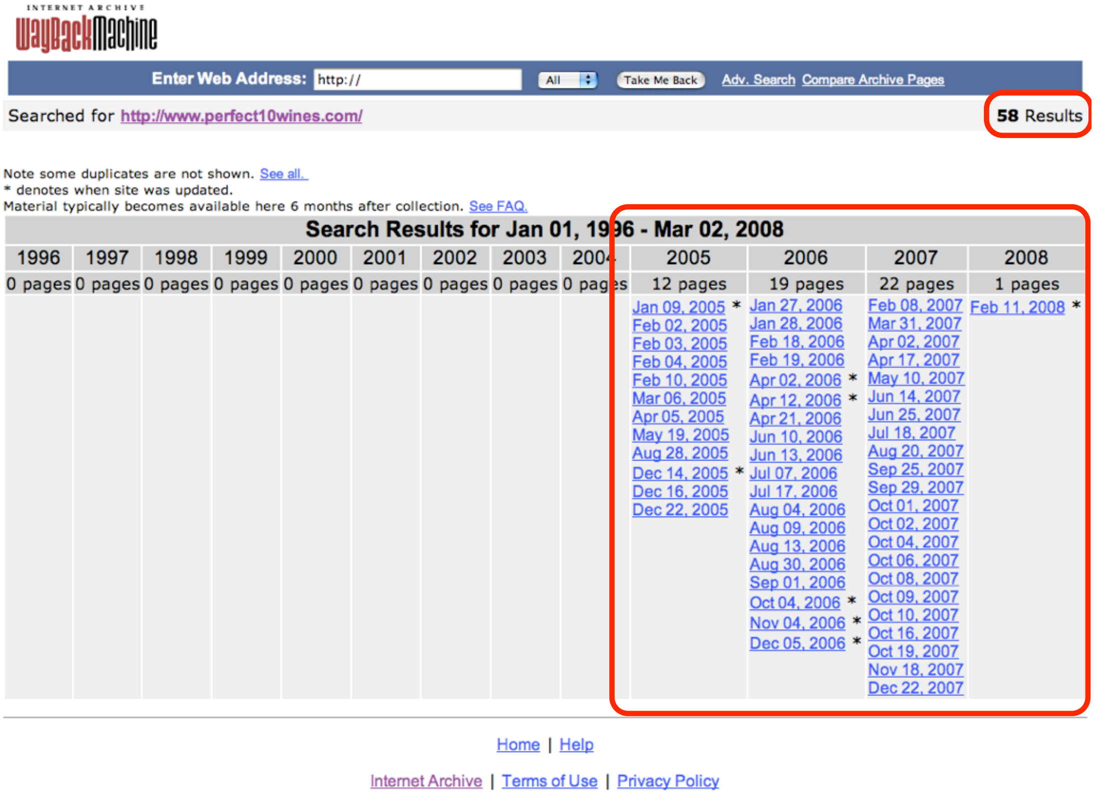
Intuitively one understands that the content of web pages changes over time. Given that, we are interested in investigating how much lexical signatures change and decay over time. Table 9, for example, shows various results of the rr function in dependence of time. The lexical signatures were created at some point in the past and again at the time of writing in early 2011. The first two lexical signatures were created by Phelps and Wilensky in the late 1990s. The lexical signatures for the Endeavour project at Berkeley change but at least shows an overlap of two out of five terms. Interestingly the zip code has made it into the lexical signature by now even though the content of the page has not changed in the last 11 years. The lexical signatures for Randy Katz's homepage in contrast do not show any term overlap. Correcting the typo in *Californa* likely contributed to the disappearance of the term since *California* is not a good discriminator in the entire index of a search engine. The Library of Congress example also shows two terms in common for both lexical signatures. Today the page is indexed with terms such as *detailurl* and *shortname* which are part of JavaScript code in the page and therefore not necessarily obvious to the user. The JCDL 2008 example shows the highest overlap of four terms. Only the email address replaced the less discriminating token

Table 9 Lexical Signatures Generated from Various URIs Over Time

URI	Lexical Signature	
	Past	Recent
`http://endeavour.cs.berkeley.edu/`	amplifies endeavour leverages charting expedition (late ’90s)	endeavour 94720-1776 achieve inter-endeavour amplifies (2011)
`http://bnrg.eecs.berkeley.edu/~randy/`	californa isrg culler rimmed gaunt (late ’90s)	randy eecs professor frameset katz (2011)
`http://www.loc.gov/`	library collections congress thomas american (2008)	library playersize detailurl shortname collections (2011)
`http://www.jcdl2008.org/`	libraries jcdl digital conference pst (2008)	libraries jcdl digital conference info@jcdl2008.org (2011)
`http://www.dli2.nsf.gov/`	nsdl multiagency imls testbeds extramural (late ’90s)	digital library dli2 2002 2003 (2009)

pst. The last example is also provided by Phelps and Wilensky. They generated the early lexical signature of the web page for the Digital Libraries Initiative 2 in the late 1990s. Today the URI returns a 404 error − the project has expired years ago. The recent lexical signature was created from the last available copy provided by the IA from 2009. We see no overlap between the lexical signatures.

These examples underline the intuition that the content of web pages changes over time and therefore their lexical signature changes. In the following experiment we modeled the decay of lexical signatures over time. The results will enable us to give a statement about the usefulness of a lexical signatures with respect to its retrieval performance. Ideally we would take snapshots of the entire web over a period of several years and evaluate the change. Since that is clearly infeasible for us we chose to utilize the IA and their best effort approach to make copies of the web. The IA provides copies of web pages starting in 1996 to the present and is therefore a great source for such a dataset. We use the “local universe” repository introduced in Chapter IV, Section 3.2 and visualized in Figure 17. For each URI we aggregate all terms per year and generate lexical signatures for each of those years. For example, the URI `http://www.perfect10wines.com` has observations in the IA in 2005, 2006 and 2007 (as Figure 23 shows) and so we generate three lexical signatures for this URI. The top 10 terms of each lexical signature along with their TF-IDF scores for this URI are shown in Table 10. This example shows a core of 8 terms (highlighted in bold) that occur in all three years. The ranking of the terms varies and the dynamics within the lexical signatures, meaning the rise and fall of words can be seen with terms such as *chardonnay* (ranked 6 in 2005



INTERNET ARCHIVE WaybackMachine

Enter Web Address: http:// All Take Me Back Adv. Search Compare Archive Pages

Searched for http://www.perfect10wines.com/ **58** Results

Note some duplicates are not shown. See all.
* denotes when site was updated.
Material typically becomes available here 6 months after collection. See FAQ.

Search Results for Jan 01, 1996 - Mar 02, 2008

1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008
0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	0 pages	12 pages	19 pages	22 pages	1 pages
									Jan 09, 2005 * Feb 02, 2005 Feb 03, 2005 Feb 04, 2005 Feb 10, 2005 Mar 06, 2005 Apr 05, 2005 May 19, 2005 Aug 28, 2005 Dec 14, 2005 * Dec 16, 2005 Dec 22, 2005	Jan 27, 2006 Jan 28, 2006 Feb 18, 2006 Feb 19, 2006 Apr 02, 2006 * Apr 12, 2006 * Apr 21, 2006 Jun 10, 2006 Jun 13, 2006 Jul 07, 2006 Jul 17, 2006 Aug 04, 2006 Aug 09, 2006 Aug 13, 2006 Aug 30, 2006 Sep 01, 2006 Oct 04, 2006 * Nov 04, 2006 * Dec 05, 2006 *	Feb 08, 2007 Mar 31, 2007 Apr 02, 2007 Apr 17, 2007 May 10, 2007 Jun 14, 2007 Jun 25, 2007 Jul 18, 2007 Aug 20, 2007 Sep 25, 2007 Sep 29, 2007 Oct 01, 2007 Oct 02, 2007 Oct 04, 2007 Oct 06, 2007 Oct 08, 2007 Oct 09, 2007 Oct 10, 2007 Oct 16, 2007 Oct 19, 2007 Nov 18, 2007 Dec 22, 2007	Feb 11, 2008 *

Home | Help

Internet Archive | Terms of Use | Privacy Policy

Fig. 23 58 Mementos of `http://www.perfect10wines.com` in the Internet Archive

and 9 in 2007) and *paso* (9 in 2005 and 3 in 2007). The example of Table 10 also shows that we did not apply stemming algorithms (*wine* and *wines*) nor eliminate stop words from the list of terms in this stage of the experiment. We conduct a term overlap analysis of the lexical signatures generated for all URIs at all available points in time and again we assume query term commutativity for our lexical signatures.

2.1 Results

We distinguish between two different overlap measures per URI:

1. **rooted** – the overlap between the lexical signature of the year of the first observation in the IA and all lexical signatures of the consecutive years that URI has been observed

2. **sliding** – the overlap between two lexical signatures of consecutive years starting with the first year and ending with the last.

For example, if a URI has copies in the IA in each year from 1996 through 2001, we would have rooted overlap values for the lexical signatures of 1996 and 1997, 1996 and 1998, 1996 and 1999, 1996 and 2000 and finally 1996 and 2001. For the sliding overlap we have data for 1996 and 1997,

Table 10 10-term Lexical Signatures generated for `http://www.perfect10wines.com` for 2005, 2006 and 2007

	2005		2006		2007	
	Term	**Score**	**Term**	**Score**	**Term**	**Score**
1	**wines**	8.56	**wines**	6.52	**wines**	5.25
2	**perfect**	5.00	**wine**	4.80	**wine**	4.50
3	**wine**	3.03	**perfect**	4.70	**paso**	4.50
4	**10**	2.60	**10**	3.45	**perfect**	4.10
5	**monterey**	2.24	**paso**	3.01	**robles**	3.75
6	**chardonnay**	2.24	**robles**	2.89	**10**	3.40
7	merlot	2.20	**monterey**	2.79	**monterey**	2.25
8	**robles**	1.99	**chardonnay**	2.79	cabernet	2.25
9	**paso**	1.99	ripe	1.86	**chardonnay**	2.25
10	blonde	1.38	vanilla	1.86	sauvignon	2.25

1997 and 1998, 1998 and 1999 etc. The term overlap is the number of terms two lexical signatures have in common e.g., if two 5-term lexical signatures have three terms in common its overlap would be $3/5 = 0.6$.

Tables 11 and 12 show the normalized (over the maximal possible) mean overlap values of all URIs where Table 11 holds the overlap values of what was introduced as rooted overlap and Table 12 holds values for the sliding overlap. All lexical signatures compared consist of 5 terms. In both tables the columns represent the year of the first observation in the IA e.g., all values for all URIs with observations starting in 1996 can be found in the column headed by 1996. The mean overlap of all URIs starting in 1996 between the starting year and 2001 can be thus be found in the first column and fifth row (the 2001-row) of Table 11. The overlap between 2003 and 2004 of all URIs with observations starting in 1999 can consequently be found in the fourth column (the 1999-column) and eight row (the $2003-2004$-row) of Table 12.

We generally observe low overlap scores for the rooted overlap (Table 11). We see an overlap of slightly above 10% between lexical signatures from 1996 and more recent ones. The maximum overlap value of 90% is scored for lexical signatures created in 2006 and 2007. Values are highest mostly in the first years after creation of the lexical signatures and then drop over time. We rarely see values peaking after this initial phase which means terms once gone (not part of the lexical signature anymore) usually do not return. This indicates that lexical signatures decay over time and become stale within only a few years after creation.

Due to the year by year comparison, it is not surprising that the sliding overlap values (shown in Table 12) are higher than the rooted overlap values. We can observe a different pattern here. Values often increase over time and quite frequently they peak in the more recent past. It almost seems that lexical signatures enter a “steady state” from a certain time on. This could be because of the way people did web sites in the late 90s. We need to point out that all values are mean values over all URIs and normalized by the maximum possible overlap. Especially for the early years due to the sparse set of observations these values may not be representative.

Table 11 Normalized Overlap of 5-Term Lexical Signatures – Rooted Overlap

compare	**Year of First Memento**										
to	**1996**	**1997**	**1998**	**1999**	**2000**	**2001**	**2002**	**2003**	**2004**	**2005**	**2006**
1997	0.33										
1998	0.13	0.33									
1999	0.13	0.20	0.56								
2000	0.13	0.33	0.49	0.51							
2001	0.20	0.27	0.31	0.46	0.58						
2002	0.13	0.33	0.33	0.32	0.48	0.64					
2003	0.13	0.13	0.40	0.40	0.47	0.54	0.66				
2004	0.13	0.13	0.36	0.35	0.40	0.53	0.60	0.66			
2005	0.13	0.07	0.38	0.37	0.37	0.42	0.50	0.63	0.58		
2006	0.13	0.20	0.31	0.35	0.38	0.48	0.51	0.46	0.62	0.80	
2007	0.20	0.20	0.27	0.29	0.37	0.44	0.50	0.37	0.52	0.60	0.90

Table 12 Normalized Overlap of 5-Term Lexical Signatures – Sliding Overlap

	Year of First Memento										
comparison	**1996**	**1997**	**1998**	**1999**	**2000**	**2001**	**2002**	**2003**	**2004**	**2005**	**2006**
1996-1997	0.33										
1997-1998	0.40	0.33									
1998-1999	0.73	0.27	0.56								
1999-2000	0.53	0.40	0.49	0.51							
2000-2001	0.47	0.87	0.56	0.62	0.58						
2001-2002	0.53	0.73	0.51	0.52	0.63	0.64					
2002-2003	0.60	0.73	0.67	0.55	0.67	0.64	0.66				
2003-2004	0.93	0.80	0.76	0.69	0.80	0.83	0.73	0.66			
2004-2005	0.87	0.80	0.73	0.66	0.82	0.68	0.83	0.74	0.58		
2005-2006	0.93	0.47	0.71	0.72	0.77	0.72	0.84	0.51	0.76	0.80	
2006-2007	0.87	0.53	0.80	0.68	0.83	0.76	0.81	0.49	0.68	0.80	0.90

Table 13 Lexical Signatures Generated from URIs Over Time Queried against Google at Different Points in Time. Results are Shown as Rank/Total Results (Year of the Query)

URI	**Past LS Queried**		**Recent LS**
	in Past	**Recently**	**Queried Recently**
`http://endeavour.cs.berkeley.edu/`	**1/1** (late '90s)	**4/194,000** (2011)	**1/139** (2011)
`http://bnrg.eecs.berkeley.edu/~randy/`	**1/<100** (late '90s)	**NA/11** (2011)	**1/9,340** (2011)
`http://www.loc.gov`	**1/174,000** (2008)	**2/356,000** (2011)	**1/17** (2011)
`http://www.jcdl2008.org`	**2/77** (2008)	**9/550** (2011)	**1/617** (2011)
`http://www.dli2.nsf.gov`	**1/1** (late '90s)	**NA/19** (2011)	**NA/8,670** (2011)

3 PERFORMANCE OF LEXICAL SIGNATURES

Table 13 shows the results of applying the *c2u* function meaning querying the lexical signatures of the URIs shown in Table 9 of Section 2 at different points in time. The query results of the lexical signatures by Phelps and Wilensky in the late 1990s can be obtained from their numerous presentations available on the web. The table shows the query results of the lexical signatures created in the past and queried in the past as well as very recently but also the results of the recently generated lexical signatures queried at the time of writing.

We can see that all lexical signatures created in the past performed very well in the past. All three lexical signatures by Phelps and Wilensky showed an excellent performance by returning the URI top ranked. Two of the three lexical signatures return the target URI as the only result. If we are not concerned about recall this is the optimal output of the *c2u* function. The lexical signature of the Library of Congress had a high recall but still returned the URI top ranked. The JCDL URI was returned ranked second with a low recall which means the lexical signature is still performing well.

Querying the old lexical signatures today shows a different picture. We do not find the DLI2 URI and the URI of Randy Katz's page returned at all. The DLI2 URI no longer exists and since it has been deleted from the search engine's index it can not be returned. The URI of Randy Katz's page however is still indexed and could have been returned. The three URIs that are returned are ranked in the top 10 which can be considered a good result of *c2u* given the dated *rr* output even though the recall is rather high. The newly generated lexical signatures perform much better with all indexed URIs returned top ranked and a low recall value.

This example proves that the performance of lexical signatures changes over time. An up-to-date output of the *rr* function potentially performs better in the sense of finding recent versions of a page. However, an old lexical signature could still be used for identifying an old version. We conduct an experiment querying the lexical signatures of our "local universe" against today's search engines and evaluate the results in terms of their performance.

We also are interested in investigating the length of well performing lexical signatures. Therefore we generate lexical signatures that differ in the number of terms they contain. Phelps and Wilensky [218] as well as Park et al. [214] chose 5-term lexical signatures assuming 5 would be good number regarding precision and recall when feeding lexical signatures back to Internet search engines. We chose a wider range starting at 2 terms going up to 10 terms and for comparison reasons we also created 15-term lexical signatures. The goal is to display a performance curve that gives an indication for a possibly dated and insufficiently performing lexical signature.

3.1 Results Over Time

We took the lexical signatures generated from all URIs of the local universe introduced in Chapter IV, Section 3.2 (over the time span of 12 years) to form queries which we issued to the Google search API between November 2007 and January 2008. Since the Google API, at the time we conducted this experiment, had a limit of 1000 queries per day, we only ask for the top 100 results. To evaluate the performance of lexical signatures we parse the result set of the lexical signature queries and identify the URI the lexical signature was created from. This filter works due to bunching in a rather liberal fashion. Even if the actual URI string is not identical, as long as the desired URI is one of the bunch, we consider it identified. For example, if the URI the lexical signature was generated from is `http://www.foo.bar` the bunch of URIs that would all count as a hit is:

- `http://foo.bar`
- `http://foo.bar/page.html`
- `http://www.foo.bar/page.html`.

The search results provided by the search engine APIs do not always match the result provided by the web interfaces ([192]) but we are using the Google API for all queries and thus are not forced to handle possible inconsistencies.

As a first step we analyze the result set by again distinguishing between our four scenarios for each URI. Either:

1. the URI is returned as the top ranked result or
2. the URI is returned in the top 10 but not as the top ranked result or
3. the URI is ranked somewhere between 11 and 100 or
4. the URI was not returned which means in our case is ranked somewhere beyond rank 100.

We consider a URI for the last case as undiscovered because numerous studies ([54, 133, 142, 143, 165]) have shown that the vast majority of Internet users do not look past the first few search results. These studies also show that users rarely click on search results beyond rank 10. We are aware of the potential discrimination of results ranked just beyond our threshold and there is an obvious difference between search results ranked 101 and, for example, rank $10,000$. However, we chose this classification for simplicity and do not distinguish between ranks greater 100.

Table 14 shows the performance statistics of all lexical signatures in today's search engine distinguished by the number of terms they consist of. It displays the relative amount of URIs returned

Table 14 Lexical Signature Length vs. Rank

	1	**2-10**	**11-100**	**≥101**	**MR**
2-term	24.3	14.9	13.2	47.6	53.1
3-term	40.2	15.0	**15.0**	29.8	36.5
4-term	43.9	15.7	11.4	**29.0**	33.8
5-term	47.0	**19.4**	3.4	30.2	**32.7**
6-term	51.2	11.4	3.4	34.1	36.0
7-term	**54.9**	9.4	1.5	34.2	35.5
8-term	49.8	7.7	2.2	40.4	41.9
9-term	47.0	6.6	0.9	45.5	46.4
10-term	46.1	4.0	0.9	49.0	49.8
15-term	39.8	0.8	0.6	58.9	59.5

with rank 1, ranked between 2 and 10, between 11 and 100 and beyond 100 therefore representing our four retrieval scenarios. The last column holds the mean values of all ranks for a particular n-term lexical signature. For all lexical signatures we can observe a binary pattern meaning the vast majority of URIs returns either ranked 1 or beyond 100. This pattern becomes even more obvious when comparing the top 10 results (including the top rank) and the number of undiscovered URIs. In none of the cases more than 15% of URIs are ranked between 11 and 100. We can see that the performance of 2-term lexical signatures is rather poor with almost 50% undiscovered URIs and a mean rank of more than 53. 3- and 4-term lexical signatures perform similarly with a slight advantage for 4-terms when it comes to top ranked results and a lower mean rank value. 5-terms lexical signatures have the overall best mean rank value. They also return more results top ranked and ranked in the top 10 than 4-term lexical signatures but less URIs ranked between 11 and 100. 7-term lexical signatures show by far the highest percentage of top ranked URIs. Their mean rank however is lower than with 5-term lexical signatures. The performance of 6-term lexical signatures is somewhere between 5- and 7 terms. The value for top 10 results are much lower than for 5-term lexical signatures. It appears this loss is split in half with one half going to the top ranked results and the other to the undiscovered results. The picture for 8-, 9- and 10-term lexical signatures is basically the same. Their performance is not very impressive and gets worse as more terms are added to the lexical signature as the values for the 15-term lexical signatures prove. The binary pattern however is best visible at these high-term lexical signatures.

Figure 24 shows the rank distribution of all URIs returned by 5-term lexical signatures over the 12 year period. The color black represents the rank beyond 100 and the lightest gray represents rank 1. All ranks in between are colored proportionally where (for reasons of readability) we clustered ranks 2-10, 11-20, 21-30, etc. into ten clusters respectively. Therefore we plot 12 different gray tones including rank 1 and rank $>$ 100. The x-axis again represents the date of the lexical signature of each URI and the URIs are numbered on the y-axis. Here the URIs are sorted. The primary ranking is done by earliest observation in the IA. Thus the URIs with observations in 1996 are plotted on the bottom, followed by URIs with observations starting in 1997 etc. This results in the step-shape of the figure. The secondary ranking effects the URI order within one year (within one step so to speak). Here all URIs are sorted by their mean rank in ascending order. This secondary ranking and

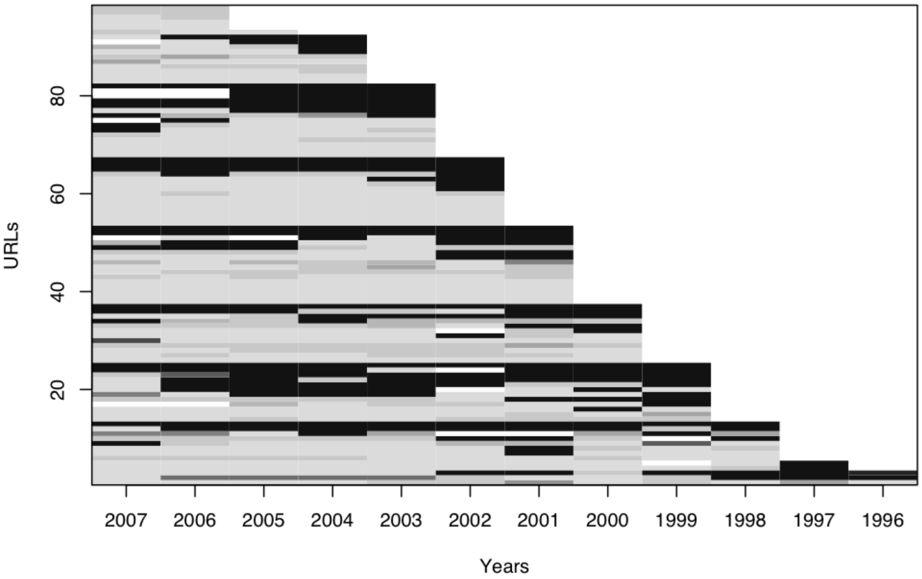


Fig. 24 Ranks of 5-term Lexical Signatures from 1996 to 2007

the color scheme mentioned above is the reason for the lighter colored bottom half and the darker upper half of each step.

Figure 24 gives visual confirmation of the numbers in Table 14. For 5-term lexical signatures we see roughly 50% top ranked results, 30% of the URIs are undiscovered (black) and 20% ranked in between 1 and 100 with a strong bias towards the top 10 (lighter gray). We can also see gaps for some URIs in the plot (white) which is simply caused by missing observations in the IA. Park et al. [214] classified the URIs returned in four categories in order to evaluate the performance of lexical signatures. With our above classification we did something very similar and subsume their four categories by applying a performance evaluation score that rewards results at the top of the result set and gives less value to results at the lower end of the result set. We use normalized Discounted Cumulative Gain (nDCG) for this purpose. We apply a binary relevance score, meaning we give the score 1 for an URI match and the score of 0 otherwise. Since we have binary relevance (i.e., one result has a value of 1 and all others have a value of 0), the IDCG will always have a value of exactly 1 which means nDCG is equal to DCG. Further, since only one member of the result set has a non-zero relevance score, and that one result always has a relevance score of exactly 1, we can simplify the DCG equation as shown in Equation 31, where i is the position of the target URI in the result set. We used this simplified DCG in previous work [264].

$$DCG* = \frac{1}{\log_2(1+i)} \tag{31}$$

If the target URI was undiscovered meaning not returned or ranked beyond rank 100, we assigned a nDCG value of 0, corresponding to an infinitely deep position in the result set.

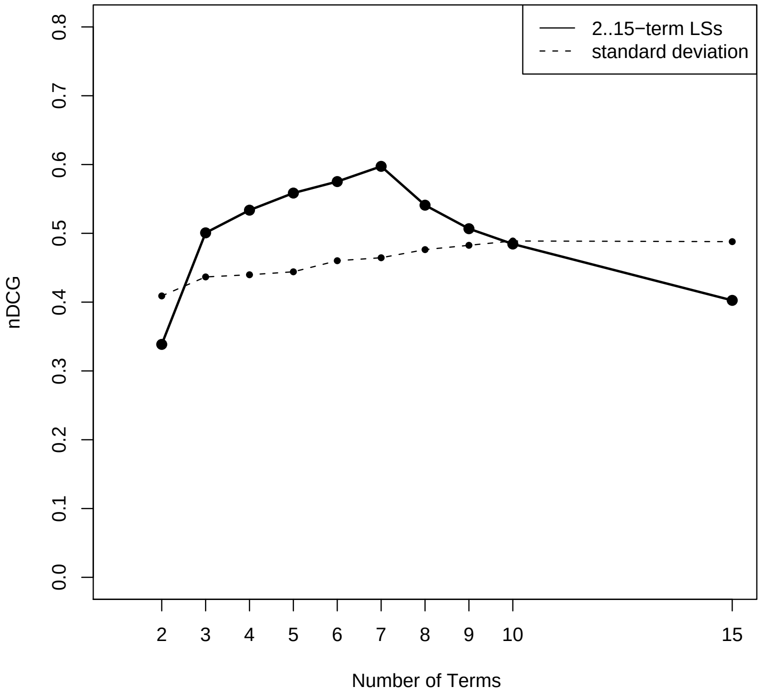


Fig. 25 Lexical Signature Performance by Number of Terms

Figure 25 shows the mean DCG values over all years. Just like in Table 14 we distinguish between lexical signatures containing $2 - 10$ and for comparison 15 terms. The figure proves that 7-term lexical signatures perform best with a mean DCG of almost 0.6. On the other end, 2- and 15-term lexical signatures are not performing well. The DCG values for 3- to 6-term lexical signatures steadily increase from 0.5 to 0.57. The values drop again for 8- through 10-term lexical signatures from 0.54 to 0.48. The figure also shows the standard deviation of the mean values drawn in as the thin dotted line. These values are high which is not surprising due to our binary relevance score paired with the binary retrieval pattern seen in Table 14.

Figure 26 displays the DCG values of selected lexical signatures over time. To keep the graphs readable we chose to display the most significant lexical signatures and do not draw a line for all n-term lexical signatures. Each data point represents the mean DCG score of all URIs of a certain year indicated by the values on the x-axis. We can clearly see the low scores of 2- and 10-term lexical signatures even though 10-term lexical signatures show a slight increase towards the more recent years. 5- and 7-term lexical signatures in contrast do much better as their score constantly increases over time reaching up to 0.56 (5-terms) and 0.6 (7-terms) in 2007. This figure proves that lexical signatures older than four or five years result in a mean DCG score of below 0.5. We consider this

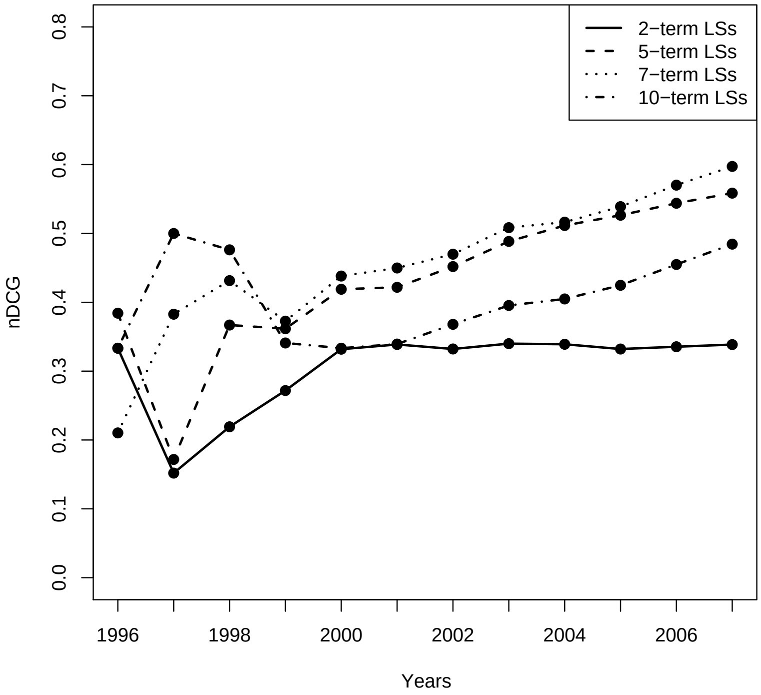


Fig. 26 Lexical Signature Performance Over Time

as our threshold and therefore declare the maximum age for a lexical signature that is expected to perform well to be at most five years.

The great fluctuation of the numbers for the early years in Figure 26 can be explained with the limited number of URIs and observations in the IA for that time. We do believe however that from roughly year 2000 on there is a pattern visible. The lines evolve much more steadily which may be because of an increase of observations in the IA from 2000 on (see Figure 17) in the sense of simply more URIs observed and also in the sense of more copies made per day/month of the same URIs.

Another interesting observation is the line for 2-term lexical signatures. Regardless of its low score it shows an almost flat line from year 2000 on. A possible explanation is that 2-term lexical signatures are in fact good for finding related pages (as shown in [125]) but do so constantly over time. That means 2-term lexical signatures constantly return relevant results with the URI of interest rarely top ranked but usually somewhere in the result set. Our intuition is that it provides good recall but its precision is rather poor and therefore the score is low.

3.2 Results in Different Search Engines

Unlike the experiment in Chapter IV, Section 3.5 we are here not interested in comparing the retrieval performance across search engines but rather generate lexical signatures from three search

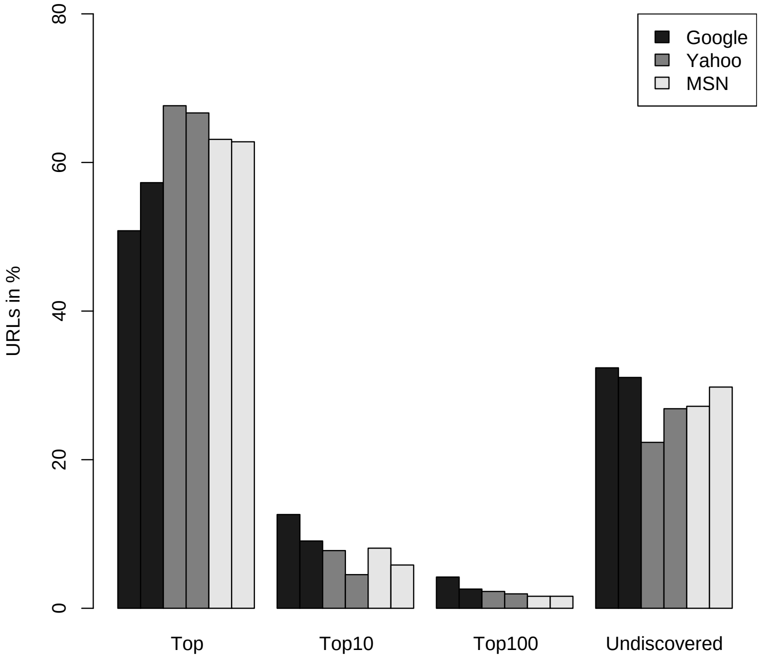


Fig. 27 5- and 7-Term Lexical Signature Retrieval Performance

engines and query them against the index they are based on. We use the Google, Yahoo! (BOSS) and MSN Live APIs to determine the DF values. Due to the results of our earlier research we query 5- and 7-term lexical signatures for each URI against search engines. As an estimate for the overall number of documents in the corpus (the Internet) we use values obtained from [37].

To obtain an increased dataset we again sampled from the Open Directory Project `http://dmoz.org` and randomly picked 500 URIs. We are aware of the implicit bias of this selection but for simplicity it shall be sufficient. We dismissed all non-English language pages as well as all pages containing less than 50 terms (this filter was also applied in [158, 214]). Our final sample set consists of a total of 309 URIs, 236 in the .com, 38 .org, 27 .net and 8 in the .edu domain. We downloaded the content of all pages and excluded all non-textual content such as HTML and JavaScript code.

The TF-IDF score of common words is very low given a sufficiently large corpus. Therefore these terms would not make it into the lexical signature, assuming the document contains enough other, less frequent terms. However, for the sake of minimizing the number of search engine queries needed to determine the DF values we dismiss stop words from the web pages before computing TF-IDF values.

Figure 27 shows the percentage of the 309 URIs retrieved top ranked, ranked in the top 10 and top 100 as well as the percentage of URIs that remained undiscovered when using 5- and 7-term lexical signatures. For each of the four scenarios we show three tuples distinguished by color, indicating the search engine the lexical signature was generated from and queried against. The left bar of each tuple represents the results for 5- and the right for 7-term lexical signatures. We again can observe the binary pattern with the majority of the URIs either returned in the top 10 (including the top rank) or remain undiscovered. If we, for example, consider 5-term lexical signatures submitted to Yahoo! we retrieve 67.6% of all URIs top ranked, 7.7% ranked in the top 10 (but not top) and 22% remain undiscovered. Hence the binary pattern: we see more than 75% of all URIs ranked between one and ten and vast majority of the remaining quarter of URIs was not discovered. Yahoo! returns the most URIs and leaves the least undiscovered. MSN Live, using 5-term lexical signatures, returns more than 63% of the URIs as the top result and hence performs better than Google which barely returns 51%. Google returns more than 6% more top ranked results with 7-term lexical signatures compared to when 5-term lexical signatures were used. Google also had more URIs ranked in the top 10 and top 100 with 5-term lexical signatures. These observations confirm our earlier findings.

4 SUMMARY

This chapter provides the results obtained from our study of the decay of lexical signatures as an implementation of the *rr* function over time. It further includes an analysis of their performance with respect to their age and the number of terms they contain. To investigate the degree of change over time we created lexical signatures of websites stretching over a 12 year time span and analyzed their overlap with a rooted and a sliding measure. The overlap measured with the rooted method is larger in the early years after creation of the web page and quickly decreases after three years. However, the sliding method shows some noise in the early years indicating substantial changes in the pages' content but somewhat stabilizes from 2003 on. This suggests that a lexical signature that is expected to perform well when queried against a search engine should not be older than five years. This result further indicates that lexical signatures, unlike proposed by Phelps and Wilensky, should not be created a priori if the incentive is to obtain a recent version of the missing page. The chances of having the content of a web page (and consequently its lexical signature) change dramatically especially after four years is very high.

Regarding the length of lexical signatures we found that 2-terms are by far insufficient. Slightly better perform 3- and 4-term lexical signatures but 5-, 6- and 7-term lexical signatures show the best performance numbers in terms of percentage of URIs returned top ranked, mean rank and nDCG. Which of these three to chose depends on the particular intention since 7 terms return the most top ranked results and show the highest nDCG but 5 terms have the best mean rank and leave fewer URIs undiscovered. We also show that including more than 7 terms worsens the performance values. For example, 15-term lexical signatures leave almost 60% of all URIs undiscovered.

These results, in aggregate with Chapter IV, constitute a framework for the creation of well performing lexical signatures. Its parameters will directly be implemented in Synchronicity introduced in Chapter X.

CHAPTER VI

TITLES

1 BACKGROUND

We have seen in the previous chapter that lexical signatures, depending on their age, can perform well as search engine queries with the intention of discovering web pages. However, the generation of lexical signatures is expensive. Recall that a TF-IDF value needs to be computed for each term (except stop words) occurring in the page. Computing IDF is costly since each term requires one query against a search engine to obtain an estimation for the DF value. The question therefore arises whether we can find a cheaper implementation of the rr function, or in other words a cheaper method to distill the "contextual aboutness" of a page. This method needs to result in a textual string suitable for our $c2u$ implementation and, when queried against a search engine perform equally well or even better than lexical signatures. Another question then is, if we find such a method, can we achieve a performance gain when combining it with lexical signature based queries?

In this chapter we investigate the performance of web pages' titles as a second rr function implementation and analyze some of their string characteristics. We maintain a few underlying assumptions regarding web pages' titles. We anticipate that a majority of web pages actually have titles. Furthermore we believe that titles are descriptive of the page's content and that titles, compared to the content itself, change infrequently over time.

2 PERFORMANCE OF WEB PAGE TITLES

To illustrate the concept behind this experiment we show two examples in Table 15. It displays the titles and lexical signatures obtained from two URIs. One page is about an airplane chartering organization and the other is a web page about a photographer named Nic Nichols. Both rr implementations output different strings of different length but $c2u$ − in this case the query against Google − returns in all cases the target URI top ranked. Both methods show great results in this example. The following sections describe an experiment investigating the performance of web pages' titles taken from URIs of a larger dataset. We further compare the titles' performance against lexical signature performances.

2.1 Title Extraction

Researchers such as Chakrabarti et al. [81] have found that up to 17% of HTML documents are lacking titles. This estimate is based on their corpus of one million random URIs. But even if we take this number as an upper bound, one can say that the majority of web pages contain titles and therefore this method is worth investigating. In a brief and somewhat brute force experiment we randomly picked $10,000$ URIs from `dmoz.org` and found that only 1.1% of URIs are lacking a title. This underlines our intuition that titles of web pages are commonplace. We used the same sample set of URIs introduced in the previous chapter since we have already downloaded all of the 309

Table 15 Example of Well-Performing Lexical Signatures and Titles Obtained from Two Different URIs

		Rank
URI	`www.aircharter-international.com`	
LS	*Charter Aircraft Jet Air Evacuation Medical Medivac*	1
Title	*ACMI, Private Jet Charter, Private Jet Lease, Charter Flight Service: Air Charter International*	1
URI	`www.nicnichols.com`	
LS	*NicNichols Nichols Nic Stuff Shoot Command Penitentiary*	1
Title	*NicNichols.com: Documentary Toy Camera Photography of Nic Nichols: Holgs, Lomo and Other Lo-Fi Cameras!*	1

URIs. We extract the titles by simply parsing the pages and extract everything between the HTML tags *¡title¿¡/title¿*.

2.2 Performance Results of Titles

Similar to the previous experiments we parse the top 100 returned results in search for the source URI. We again distinguish between our four retrieval scenarios:

1. the URI returns top ranked
2. the URI returns in the top 10 but not top ranked
3. the URI returns in the top 100 but not top ranked and not in the top 10
4. the URI does not return, neither top ranked, nor in the top 10 or top 100.

With these scenarios we evaluate our results as success at 1, 10 and 100. Success is defined as a binary value, as the target either occurs in the subset (top result, top 10, top 100) of the entire result set or it does not.

Figure 28 shows the percentages of retrieved URIs when querying the title of the pages. We queried the title once without quotes and once quoted, forcing the search engines to handle all terms of the query as one string. Each tuple is distinguished by color and the left bar shows the results for the non-quoted titles. It is surprising to see that both Google and Yahoo! return fewer URIs when using quoted titles. Google in particular returns 14% more top ranked URIs and 38% fewer undiscovered URIs for the non-quoted titles compared to the quoted titles. Only MSN Live shows a different behavior with more top ranked results (almost 8% more) for the quoted and more undiscovered URIs (more than 7%) using the non-quoted titles.

We can see however that titles are a very well performing alternative to lexical signatures. The top value for lexical signatures taken from Figure 27 was obtained from Yahoo! (5-term) with 67.6% top ranked URIs returned and for titles with Google (non-quoted) which returned 69.3% URIs top ranked.

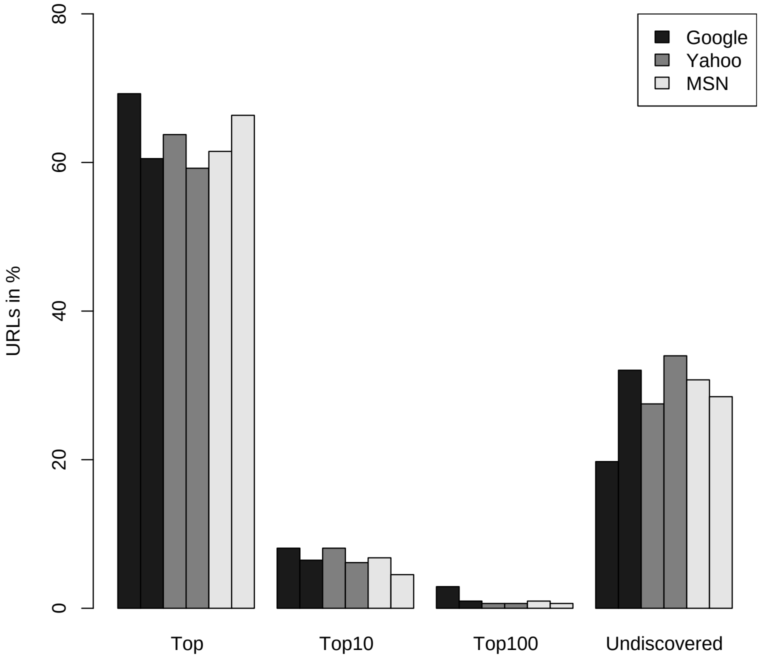


Fig. 28 Non-Quoted and Quoted Title Retrieval Performance

2.3 Combined Title and Lexical Signature Performance

The observation of well performing lexical signatures and titles leads to the question of how much could we gain if we combined both for the retrieval of the missing page? We modify the original $c2u$ function by adding a second iteration. The first iteration feeds lexical signatures into $c2u$. All URIs that remained undiscovered are subject for the second $c2u$ iteration. We feed their titles into the function and analyze the returned ranks of this second iteration in order to evaluate the potential performance gain. Table 16 summarizes the results shown in the sections above. It holds the relative numbers of URIs retrieved using one single method. The first, leftmost column indicates the method. $LS5$ and $LS7$ stand for 5- and 7-term lexical signatures and TI stands for titles. The

Table 16 Relative Number of URIs Retrieved with one Single Method from Google, Yahoo! and MSN Live

	Google				Yahoo!				MSN Live			
	1	**10**	**100**	> 100	**1**	**10**	**100**	> 100	**1**	**10**	**100**	> 100
LS5	50.8	12.6	4.2	32.4	**67.6**	7.8	2.3	22.3	63.1	8.1	1.6	27.2
LS7	57.3	9.1	2.6	31.1	**66.7**	4.5	1.9	26.9	62.8	5.8	1.6	29.8
TI	**69.3**	8.1	2.9	19.7	63.8	8.1	0.6	27.5	61.5	6.8	1.0	30.7

Table 17 Relative Number of URIs Retrieved with Two or More Methods Combined

	Google				Yahoo!				MSN Live			
	1	**10**	**100**	> 100	**1**	**10**	**100**	> 100	**1**	**10**	**100**	> 100
LS5-TI	65.0	15.2	6.1	13.6	**73.8**	10.0	2.3	14.0	71.5	10.0	1.9	16.5
LS7-TI	70.9	11.7	4.2	13.3	**75.7**	7.4	1.9	14.9	73.8	9.1	1.9	15.2
TI-LS5	73.5	9.1	3.9	13.6	**75.7**	9.1	1.3	13.9	73.1	9.1	1.3	16.5
TI-LS7	74.1	9.4	3.2	13.3	**75.1**	8.7	1.3	14.9	74.1	9.1	1.6	15.2
LS5-TI-LS7	65.4	15.2	6.5	12.9	**73.8**	10.0	2.6	13.6	72.5	10.4	2.6	14.6
LS7-TI-LS5	71.2	11.7	4.2	12.9	**76.4**	7.8	2.3	13.6	74.4	9.1	1.9	14.6
TI-LS5-LS7	73.8	9.1	4.2	12.9	**75.7**	9.1	1.6	13.6	74.1	9.4	1.9	14.6
TI-LS7-LS5	74.4	9.4	3.2	12.9	**75.7**	9.1	1.6	13.6	74.8	9.1	1.6	14.6
LS5-LS7	52.8	12.9	6.5	27.8	**68.0**	7.8	2.9	21.4	64.4	8.4	2.6	24.6
LS7-LS5	59.9	9.7	2.6	27.8	**71.5**	4.9	2.3	21.4	66.7	7.1	1.6	24.6

top performing single methods are highlighted in bold figures (one per row).

Table 17 shows in a similar fashion all reasonable combinations of methods involving lexical signatures and titles. A combination of methods formally translates to a sequence of $c2u$ iterations with varying rr implementations as their input. The sequences displayed in the leftmost column are sensitive to their order of inputs, i.e. there is a difference between applying 5-term lexical signatures first and 7-term lexical signatures second and vice versa. The top results of each sequence of methods are again highlighted in bold numbers.

Regardless of the sequence and their input, the best results are obtained from Yahoo!. If we consider all combinations of only two methods we find the top performance of 75.7% twice in the Yahoo! results. Once with $LS7 - TI$ and once with $TI - LS5$. The latter sequence is preferable for two reasons:

1. titles are easy to obtain and do not involve a complex computation and acquisition of DF values as needed for lexical signatures, and

2. this methods returns 9.1% of the URIs in the top 10 which is 1.7% more than the first sequence returns. Even though we do not distinguish between rank two and rank nine, we still consider URIs returned within the top 10 as good results.

The sequence $LS7 - TI - LS5$ accounts for the most top ranked URIs overall with 76.4%. While the 3-method sequence returns good results, there are not drastically better than, for example, the two methods mentioned above. The performance delta is not sufficient to justify the expensive generation of lexical signatures without using the easy to obtain titles first.

The last two rows in Table 17 show results for sequences of methods based purely on lexical signatures. The results again are good but it is obvious that a combination with titles (either as the first or second method) provides better results.

Yahoo! uniformly gave the best results and MSN Live was a close second. Google was third, only managing to outperform MSN Live once ($TI - LS5$) at the top rank.

2.4 Analysis of Title Characteristics

Given that the title of a page seems to be a good method considering its retrieval performance we further investigate the title characteristics of our dataset.

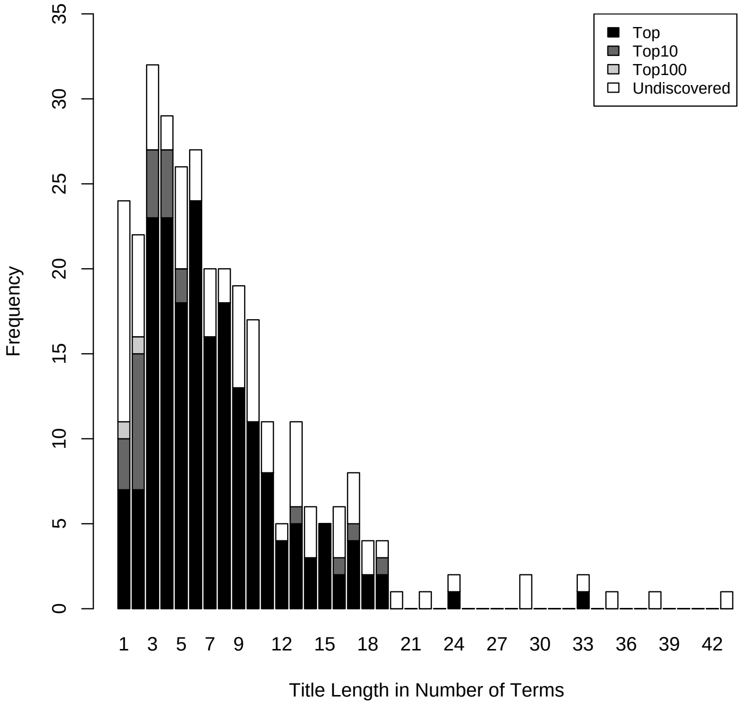


Fig. 29 Title Length in Number of Terms vs Rank

Title Length

The relation between title length in number of terms and the title's retrieval performance is shown in Figure 29. Each occurring title length is represented by its own bar and the number of times this title length occurs is indicated by the height of the entire bar. The shaded parts of the bars indicate how many titles of the according length performed in what retrieval class (the usual, top, top 10, top 100 and undiscovered). The titles vary in length between one and 43 terms. However, there is, for example, no title with length 21, hence its bar is of height null. Visual observation indicates a title length between three and six terms occurs most frequently and performs best compared to shorter titles and titles containing more terms.

The contrast of total title length in number of characters and rank is shown in Figure 30. Unlike in Figure 29 we left out non-represented title lengths here. While it seems difficult to determine a typical title length (it varies greatly between 4 and 294 characters) we only see 15 URIs with a title length greater or equal to 100 characters. Furthermore, only three URIs contain more than 200 characters in their title. Figure 30 does not reveal an obvious relationship between the number of characters and the rank returned for a title. However, we can see that very short titles (less than 10 characters) do not perform well. A title length between 10 and 70 characters is most common and the ranks seem to be best in the range between 10 and 45 characters.

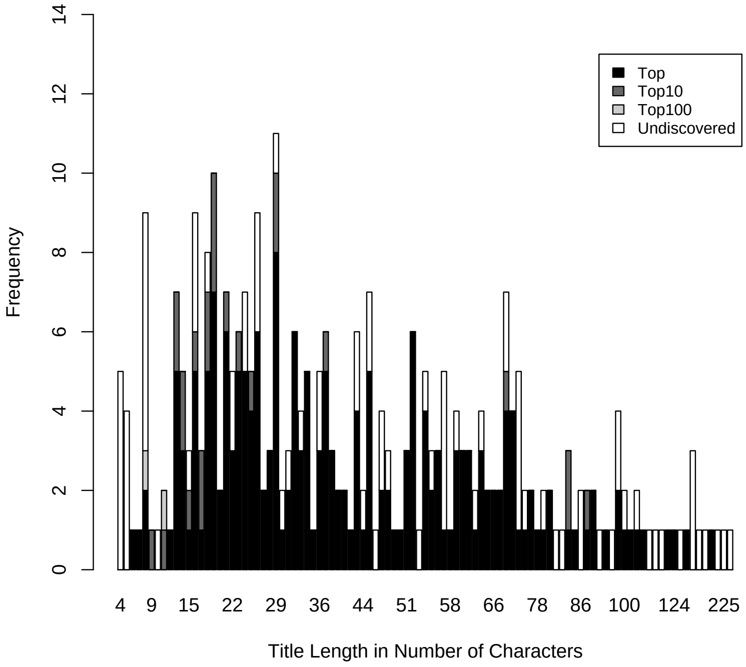


Fig. 30 Title Length in Number of Characters vs Rank

Title Term Characteristics

Besides the overall title length we are looking at the length of each title term. Figure 31 depicts on the left the mean number of characters per title term and their retrieval performance. Terms with an average of 5, 6 or 7 characters seem to be most suitable for well performing queries. On the bottom right end of the barplot we can see two titles that have a mean character length per term of 19 and 21. Since such long words are rather rare they perform very well.

It is not surprising that we observe stop words in the titles. As shown on the right in Figure 31 the performance of titles with more than one or two stop words suffers. Search engines generally filter stop words from the query and therefore (for non-quoted titles) it makes sense that, for example, the title with 11 stop words does not return its URI within the top 100 ranks.

3 QUALITY OF WEB PAGE TITLES

We have shown in the previous section that web pages' titles can perform very well as search engine queries to discover a page. However, titles are usually created by humans which intuitively makes us understand that not all titles are equally good. We further assume that titles, as well as the pages' content, undergo some changes over time and therefore, similar to lexical signatures, with time may become less useful when describing a page's content.

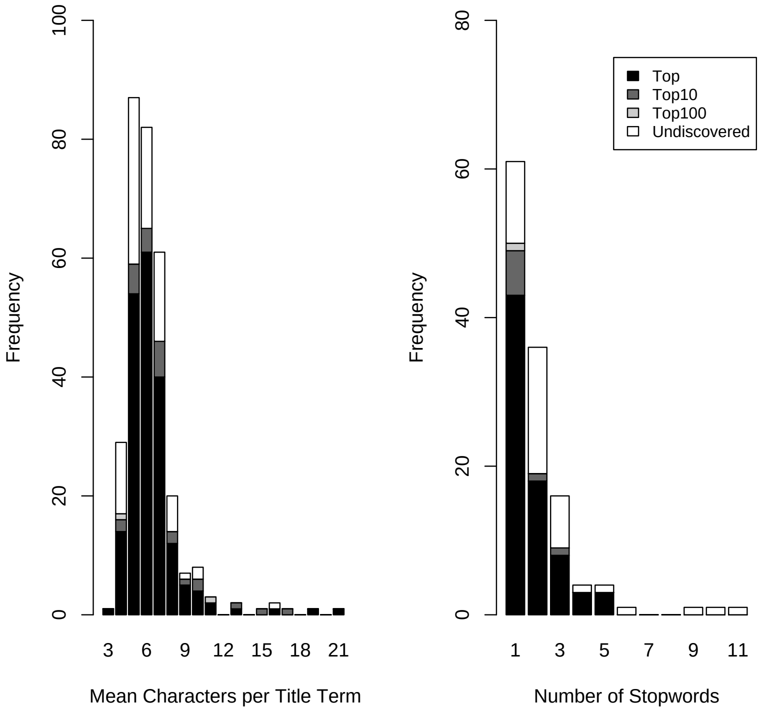


Fig. 31 Mean Number of Characters per Title Term and Number of Stop Words vs Rank

The examples displayed in Table 18 further motivate the research questions in this section. The examples illustrate the differences between titles and lexical signatures and their retrieval performance. We have already seen the first example in Table 15. The lexical signature from the URI `www.aircharter-international.com` as well as its title perform very well and return the URI as the top ranked result when queried against Google. Both methods result in query string that describe the page's content well and contain terms that (in combination) sufficiently discriminate against other pages. For this example URI one could have chosen either method and the results would have been equally good. Given the costs to generate a lexical signature however, using the title should be the first choice.

In the second example, the URI `www.redcrossla.org`, we obtain a lexical signature that contains terms which were part of the page at the time of the crawl but are less descriptive for the overall "aboutness" of the page. Instead, they represent the content of a single snapshot of the page at a certain point in time. Hence the output of the page's rr function is very sensitive to change of the page's content. Therefore the $c2u$ function does not return the page within the top 100 results and hence, as per our definition, remains undiscovered. The title of the page however captures the timeless essence of a web page of the Red Cross in Los Angeles and consequently performs much better. Submitted to $c2u$ it returns the URI top ranked. This example illustrates that

Table 18 Examples for Well and Poorly Performing Lexical Signatures and Titles

		Rank
URI	`www.aircharter-international.com`	
LS	*Charter Aircraft Jet Air Evacuation Medical Medivac*	1
Title	*ACMI, Private Jet Charter, Private Jet Lease, Charter Flight Service: Air Charter International*	1
URI	`www.redcrossla.org`	
LS	*Marek Halloween Ready Images Schwarzenegger Govenor Villaraigosa*	> 100
Title	*American Red Cross of Greater Los Angeles*	1
URI	`smiledesigners.org`	
LS	*Dental Imagined Pleasant Boost Talent Proud Ways*	1
Title	*Home*	> 100

despite the reliable TF-IDF based selection of the most salient terms of a page, a lexical signature is not automatically the best chosen *rr* implementation. This especially holds true for pages with frequently changing content such as news pages, forum pages, etc. A web page's title can be more robust since a title is understood to capture the overall topic of a page or a document. Our third example represents data taken from the URI `smiledesigners.org`, a web page of a dentist. The generated lexical signature contains terms that one could intuitively consider suitable for a search query and indeed *c2u* returns the URI top ranked. The title however is an unfortunate choice. While *Home* may be a good title within the site, it does not identify this page among others in the web. Submitted to *c2u* it does not return the URI within the top 100 results (but it is indexed with the term). This last example shows that not all titles are equally good for web retrieval and well suited as an *rr* implementation. Some user may simply not pay attention to what their title should be and others may just adopt the default of their web page generation tool or editor.

In this section we are investigating the quality of web pages' titles. We evaluate the relevancy of results returned from *c2u* when executed with a title. We obtain temporal snapshots of pages and their titles in order to investigate the evolution of titles and content over time. We are further searching for quality indicators that can be applied in real time to give a prediction of the performance of any given title.

3.1 Similarity of Search Results

A New Dataset

We were motivated to increase our corpus from Section 2.1. We started with randomly sampling 20,000 URIs from the Open Directory Project `dmoz.org`. Similar to the filters applied in [158, 214] we first dismissed all pages containing less than 50 terms from our 20,000 URI set. Second we applied a very restrictive off-the-shelf English language filter (the Perl package *Lingua::Identify* available through CPAN). This process shrank our corpus down to 6,875 pages. Table 19 shows the top level domains of the originally sampled URIs and of the final filtered dataset. We downloaded the content of all 6,875 pages and excluded all HTML elements.

Table 19 Sample Set URI Statistics

	.com	*.org*	*.net*	*.edu*	**Sum**
Original	15289	2755	1459	497	20000
Filtered	4863	1327	369	316	6875

Title Extraction and Copies from the Internet Archive

This dataset also supports our argument that titles of web pages are commonplace. Only 0.6% of all web pages (a total of 41) did not have a title. We extracted the titles of all pages by filtering all terms between the HTML tags *¡title¿¡/title¿*.

In order to investigate the temporal aspect of the title evolution we queried the URIs against the Internet Archive (IA). The IA and its crawler are not in competition with search engines. It rather is a best effort approach and all copied pages remain in a "quarantine period" for six to 12 months before they become accessible through the IA interface. Out of our $6,875$ pages the IA provided copies for $6,093$ URIs. We downloaded all available copies (more than $500,000$) and extracted the pages' content and titles.

Title and Lexical Signature Performance

For the sake of completeness we repeated parts of the experiment discussed in Section 2.2 with this larger dataset. Due to the increased number of URIs and hence number of queries against search engines we only queried the Yahoo! BOSS API. Also due to the results found in Section 2.2 we did not query with quoted titles. The results in Figure 32 confirm our earlier findings, the binary retrieval pattern and titles slightly outperforming 5- and 7-term lexical signatures.

3.2 Similarity of Search Results

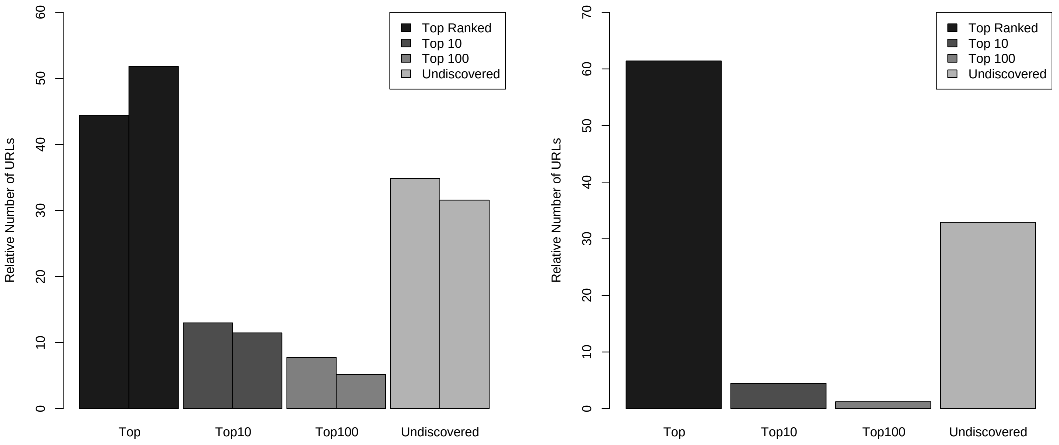
Thinking about web pages' titles as search engine queries lets us intuitively identify three special cases of search results:

1. *Aliases*, meaning two or more URIs resolve virtually the same content where the URIs may or may not canonicalize to the same value
2. *Duplicates*, meaning two or more pages hold duplicated content or a large subset of the other
3. *Title collisions*, meaning two or more pages share the same title but their content is very different.

To further illustrate these special cases let us explore the following examples: The Wikipedia page for "Lateef"[1] and "Lateef The Truth Speaker"[2] are the same and hence can be considered as aliases. The former has a note saying "(Redirected from Lateef the Truth Speaker)", but there is no HTTP notification about their equivalence. Note that Google indexes the former, and Yahoo! indexes the latter, however, neither search engine produces duplicate links.

[1] http://en.wikipedia.org/wiki/Lateef

[2] http://en.wikipedia.org/wiki/Lateef_the_Truth_Speaker



(a) 5- and 7-Term Lexical Signature Performance

(b) Title Performance

Fig. 32 Retrieval Performance of Lexical Signatures and Web Pages' Titles

The Wikipedia page for "baseball balk"[3] and the answers.com page[4] can be considered duplicates. The two pages have overlapping content, in part because they both quote from the rule book. The answers.com page additionally includes Wikipedia content as well as other sources[5]. Baeza-Yates et al. [61] explored the notion of genealogical trees for web pages where children would share a lot of content with their parents and are likely to become parents themselves. Our notion of duplicates is similar to this concept.

An example for title collisions are the following five pages:

```
http://www.globalrei.com/photos.php?property_ID=70694
http://www.globalrei.com/631-Westover-Aveune-a70694.html
http://www.globalrei.com/properties.php
http://www.globalrei.com/about.php
http://www.globalrei.com/globalrei/frm/3265/market_information/
```

All pages have the same title "Welcome to my new website!" but their content is very different. In fact the last URI even returns a customized 404 error page as introduced in Chapter III, Section 2.4.

In order to identify these three cases and reiterate our argument that web pages' titles perform well in search we investigate the similarity of the top 10 results with the originating URI (the URI whose title was used as a query against the search API). We used two methods to explore the similarity: normalized term overlap and k-shingles [77, 78] with a size of $k = 5$. We simplify the notion of retrieval to a binary scenario meaning either the URI was discovered or not. We define a

[3] http://en.wikipedia.org/wiki/Balk
[4] http://www.answers.com/topic/balk
[5] http://en.wikipedia.org/wiki/Answers.com

URI as discovered if it was returned within the top 10 search results including the top rank. For all other cases the URI is considered undiscovered. We are aware that we discriminate against URIs ranked between 11 and 100 with respect to our earlier measure of retrieval. However, the binary pattern which is obvious in Figure 32 supports this simplification.

We show both in Figure 33, the normalized term overlap (o) and shingle values (s) divided in five classes depending on their value:

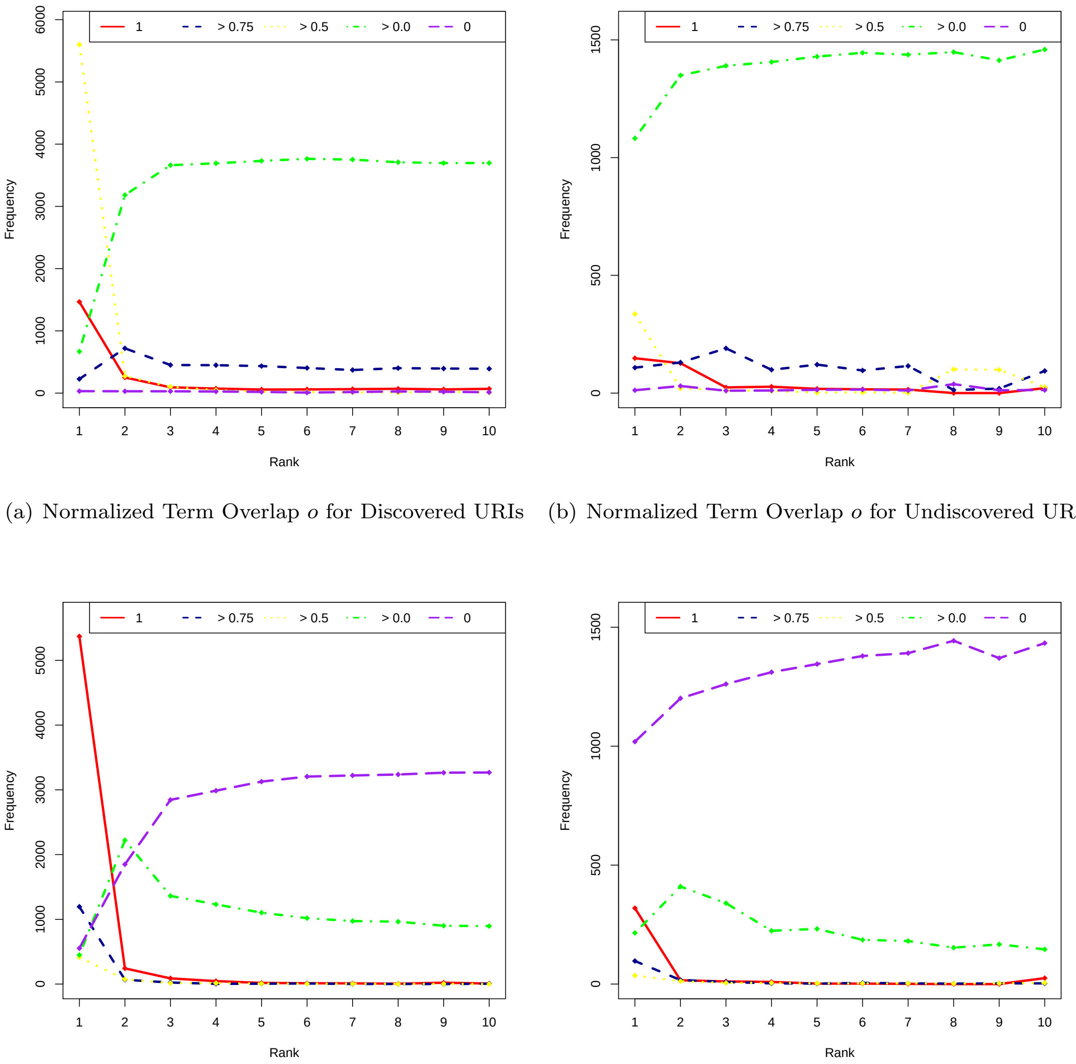
- $o, s = 1$
- $1 > o, s \geq 0.75$
- $0.75 > o, s \geq 0.5$
- $0.5 > o, s > 0.0$
- $o, s = 0$

Figure 33(a) displays the occurrence frequency of normalized term overlap values for all discovered URIs. The top rank is dominated by an overlap between 50% and 75%. The fact that only 1466 top ranked URIs have the perfect overlap despite the more than 60% top ranked URIs shown in Figure 32 indicates that the content of the pages has changed between the time we crawled the page and the time we queried the search engine with its titles. From rank three on the most frequent overlap value is between 1% and 50%. Figure 33(b) shows a similar graph for all undiscovered URIs. The lower overlap class of values between 1% and 50% throughout the ranks stands out. The perfect overlap is only noticeable for the top rank which indicates discovered aliases. The class with values between 50% and 75% occurs most frequently for the top rank as well which further indicates the discovery of duplicates.

A different and potentially better measure of document similarity is w-shingles. The shingle value of 1 ($s = 1$) is an indicator of strong similarity (note that it does not guarantee identical content) and a null value indicating no similarity between the shingles. Figure 33(c) shows five classes of shingle values by rank for discovered URIs. We see the dominance of the top rank with shingle value $s = 1$ which is not surprising considering the great amount of URIs discovered top ranked (see Figure 32). However, this optimal shingle value is achieved more often than number of URIs discovered top ranked. That indicates we discovered aliases and duplicates since for those cases we expect shingle values to be high. The zero value is rather low for the top rank, increases for rank two and three and then levels off. Figure 33(d) shows the same classes of shingle values for undiscovered URIs. As expected in this scenario the zero value occurs very frequently. However, the 300 occurrences of $s = 1$ for the top rank is surprisingly good. It indicates that here as well we have discovered a number of duplicates and aliases within the top ranks unlike the original URI.

3.3 Title Evolution Over Time

It is our intuition that web pages' titles change less frequently and less significantly than the web pages' content. The title supposedly reflects the general topic of a page which naturally changes less often than its content. With all from the IA provided copies downloaded we are able to investigate the temporal aspect of title changes. However, both the time intervals in which the IA makes copies



(a) Normalized Term Overlap o for Discovered URIs (b) Normalized Term Overlap o for Undiscovered URIs

(c) Shingle Values s for Discovered URIs (d) Shingle Values s for Undiscovered URIs

Fig. 33 Five Classes of Normalized Term Overlap (o) and Shingle Values (s) by Rank for Discovered and Undiscovered URIs. $o, s = 1$; $1 > o, s \geq 0.75$; $0.75 > o, s \geq 0.5$; $0.5 > o, s > 0.0$; $o, s = 0$

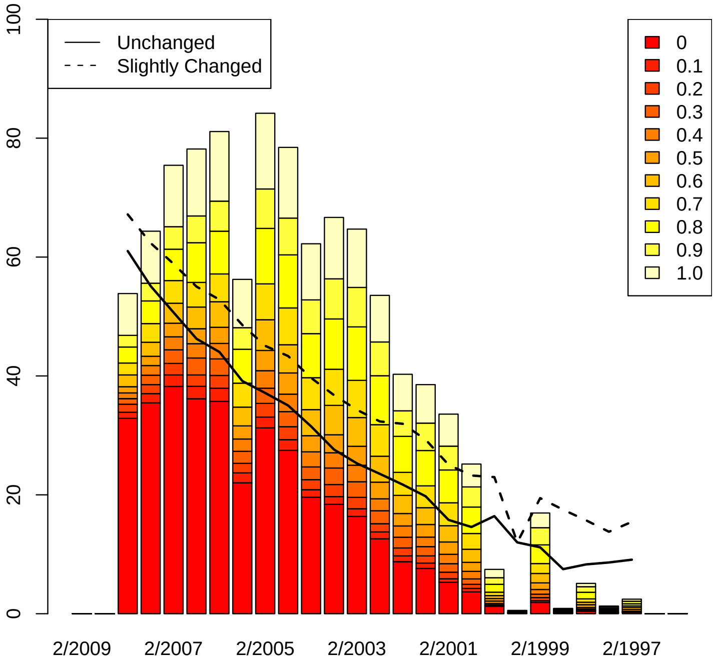


Fig. 34 Title Edit Distance Frequencies per Internet Archive Observation Interval

of web pages as well as the dates of the earliest and latest copies of pages available in the IA can vary. In order to be able to investigate the title evolution over time we need to generalize the available copies of each URI. We define 60 day time windows in which we pick one copy per URI as representative for the according time and URI. We define two such time windows per year, one in February and one in August. Since the IA provides copies of web pages from as early as 1996 we have a total of 27 time windows and hence a maximum of 27 representative copies per URI from the last 14 years.

The Levenshtein edit distance gives a measure of how many operations are needed to transform on string into another. We compute the edit distance between the titles obtained from the pages as they were downloaded in August of 2009 (our baseline) and our representative copies from the IA. We normalize the distance to a value between zero and one where one means the titles are completely dissimilar and a value of zero indicates identical titles. The edit distance distribution per time interval is shown in Figure 34. The time intervals are represented on the x-axis with the most recent on the far left decreasing to the right. The number of available copies in the early years of the IA is rather sparse. There are no copies of any of our web pages available in the first two time intervals (2/2009 and 8/2008). A possible explanation is the IA internal quarantine period mentioned earlier. However, for the third time interval we find copies of roughly 55% of all URIs. The graph reveals that about half of the available titles from the more recent copies up until 2/2006 are identical or at least very similar to the baseline. For example, we find copies of about 80% of all URIs for time interval 2/2007 and more than half of those titles have an edit distance of zero. This ratio drops for earlier copies. For the time intervals in 2002, for example, we see only about 30% of the available titles with a distance value of zero. We have to keep in mind though that copies for only about 40% of our URIs are available from that time. From 2006 on it seems that the percentage of low edit distance values decreases while the amount of higher distances increases. The solid line in Figure 34 indicates the probability, based on our corpus, that a title of a certain age is unchanged. Our data reveals that for copies as old as four years we have a 40% chance of an unchanged title. We define titles that have an edit distance value of ≤ 0.3 as titles with only minor changes compared to the baseline. The dashed line represents the chances for such titles given their age. We can see that for copies of web pages as old as 5.5 years we have a probability of at least 40% that the title has undergone only minor changes.

3.4 Title Evolution vs Document Changes

If a page's title changes less frequently over time than its content the title could constitute a reliable search engine query for discovering missing web pages. To prove this intuition we computed shingle values for all available IA copies in the above mentioned time intervals and our baseline version of the according page downloaded in August of 2009. We normalized these values so that zero indicates a very similar page and the value of one a very dissimilar page content. In order to compare these values with the edit distance of our titles in two dimensions we computed the average of all available copies in our time intervals per URI.

Figure 35 shows the average normalized edit distance on the x-axis and the average normalized shingle value of the according URI on the y-axis. Both values are rounded to the nearest tenth. The color indicates the overlap per point or in other words the amount of times a certain point was plotted. The palette starts with a basic green indicating a frequency of less or equal than 10 and transitions into a solid red representing a frequency of more than 90. The semi-transparent numbers represent the total amount of points in the according quarters and its halves. The pattern is very apparent. The vast majority of the points are plotted with an average shingle value of above 0.5 and an average edit distance of below 0.5. The most frequent point in fact is plotted more than $1,600$ times. It is (as an exception) colored black and located at the coordinates $[0, 1]$ meaning close to identical titles and very dissimilar content. The point at $[0, 0]$ is plotted 122 times and hence somewhat significant as much as some points with a shingle value of one and an edit distance of above 0.5. These points have transitioned to red.

Figure 35 supports our intuition that titles change less significantly over time than the pages' content. Therefore we claim titles to be the more robust technique with respect to content changes. That means if we, for example, wanted to find what corresponds to a five year old Memento, perhaps the lexical signature captures the aboutness at that specific time better but the title captures a "longer range aboutness" of a web resource.

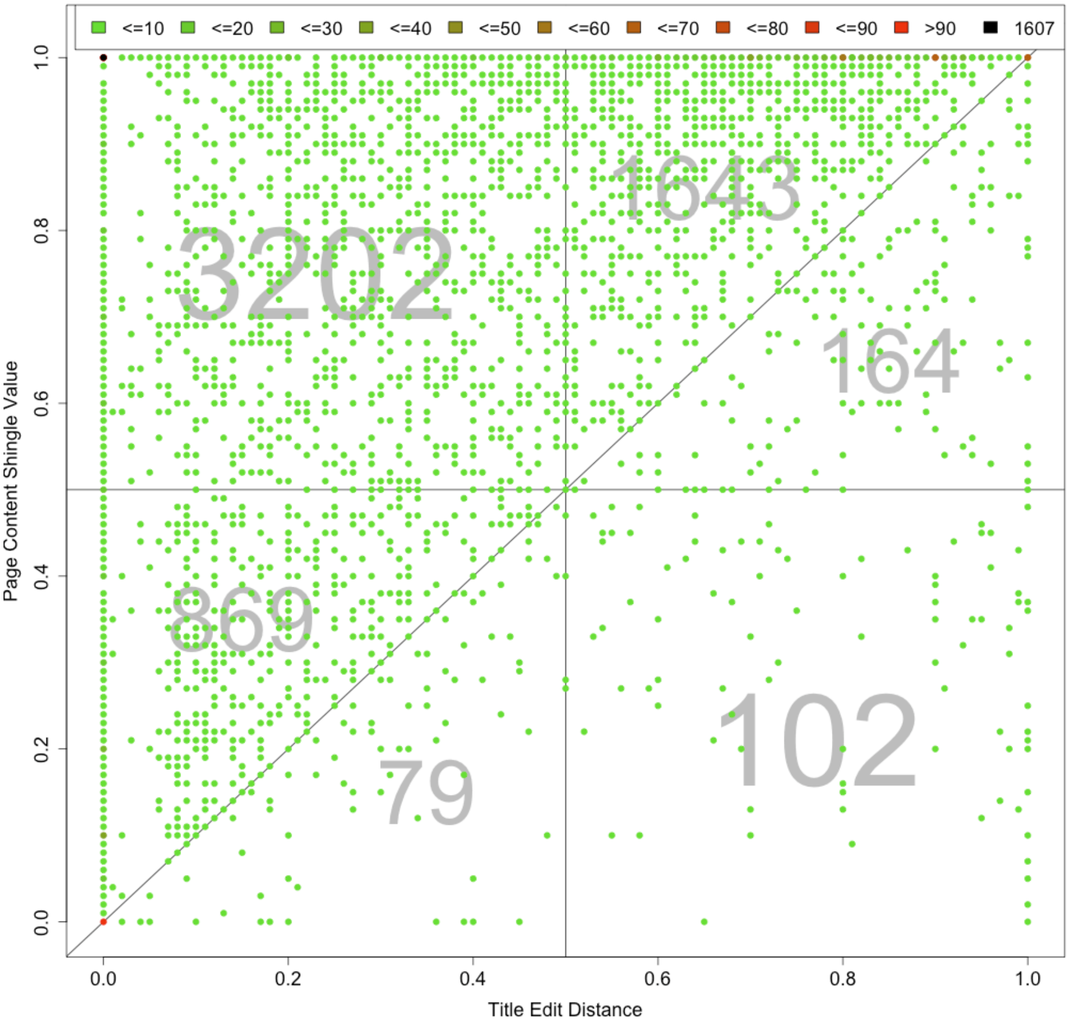


Fig. 35 Title Edit Distance and Document Changes of URIs

3.5 Title Performance Prediction

The examples shown in Table 18 support the intuition that not all titles are equally good in terms of their performance as search engine queries for the discovery of web pages. The title *Home* clearly does not perform well and we can imagine other possible titles that are equally non-descriptive such as *Index* or *Welcome*.

We are interested in a method to analyze any given title in real time and give a probability for its usefulness for our search. If we can identify a candidate bad title, it can be dismissed right away and we proceed with the generation of a lexical signature as the more promising approach for web page discovery.

Ntoulas et al. [211] used methods based on web pages' content to identify spam web pages. One of their experiments shows that web pages' titles consisting of more than 24 terms are likely to be spam. This result confirms that there are indicators by which we can predict the usefulness of titles for search. Figure 36(a) displays the composition of our titles with respect to the number of terms (on the x-axis) and number of characters (y-axis) they contain. The two different colors of the dots represent cases where URIs were found and not found. This graph supports the findings of Ntoulas et al. since it is visible that titles containing between two and 10 terms and less than 200 characters return more URIs discovered than undiscovered and hence can be considered good titles. The graph further shows that although search engines may enforce query limitations (number of query terms or number of characters) on their APIs, we can not say with certainty that queries that exceed the limitations will be unsuccessful. That means in case the servers silently truncate the queries the titles still may hold enough salient information in the first n characters/terms that do not exceed the possible limit to discover the URI. Figure 36(b) shows the same information for titles with nine or less terms. These titles account for more than 70% of all titles in the entire corpus.

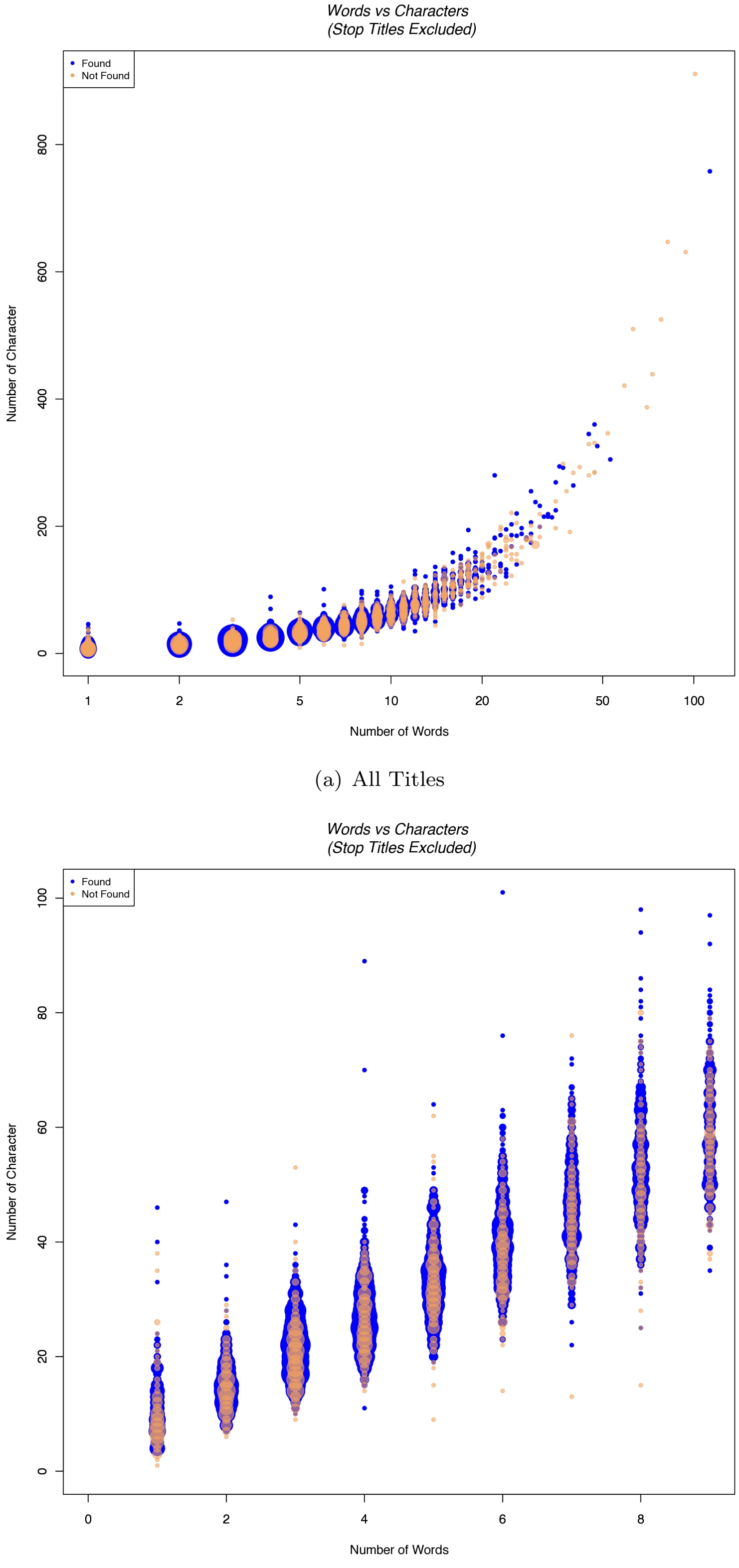
Most search engines automatically filter stop words (language dependent) from a query string since they are not crucial when it comes to describing a pages' content. For titles, however, our intuition is that we can identify additional terms that would not necessarily occur in a common stop word list for the English language but do not contribute to uncover the "aboutness" of a web page.

We analyze our corpus of $6,875$ titles and identify those that (used as the query string) do not lead to rediscovering the originating page. We call these titles **stop titles**. For the sake of simplification we narrow the rediscovery to a binary value meaning URIs that are returned within the top 10 search results including the top rank are considered discovered and all remaining URIs are considered undiscovered. Some of the most frequent stop titles in our corpus are *home*, *index*, *home page*, *untitled document* and *welcome*. We further argue that terms such as *main page*, *default page* and *index html* should be added to the list of stop titles even though they did not occur in our dataset. The experienced Internet user and website creator will recognize most of these terms as the default title setting of various web page generating tools. A complete list of stop titles from this experiment can be found in Appendix A.

With the list of stop titles our approach is to automatically identify bad titles. The trivial case is to match a given title with all of the stop titles and if we find a match the title is classified as bad. The second approach is to compute the ratio of stop titles and total number of terms in the title. The analysis on our corpus has shown that if this ratio is greater than 0.75 the likelihood of the title performing poorly is very high and hence the title should be dismissed. Figure 37(a) shows the sentinel value indicating the upper bound for the ratio based on our corpus and the binary discovered/undiscovered classification of all titles.

Table 20 shows the confusion matrix for the second approach based on our experiments. We can see that with the evaluated upper bound for the ratio (0.75) we obtain a total match of more than 71% and hence a mismatch of 28%.

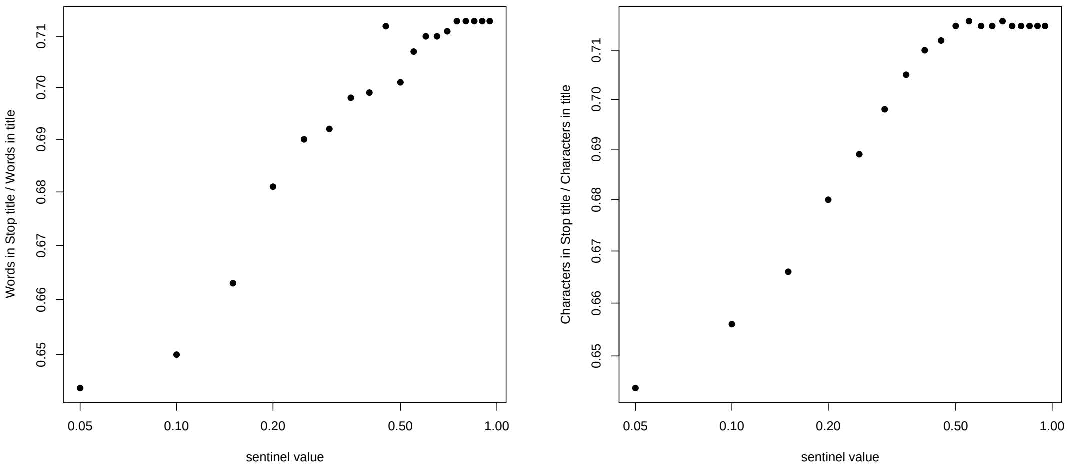
The third approach is based on number of single characters in the title. That means if a title contains stop titles we determine the ratio of number of characters in the stop title and number of total characters of the title. The analysis of our corpus has shown that a ratio greater than 0.75 predicts a poor performance (as also shown in Figure 37(b)). Table 21 shows the according confusion matrix to the third approach based on our experiments. The upper bound (also 0.75) accounts for similarly good numbers with a total match of more than 71% but also achieves 0% false positives.



(a) All Titles

(b) Titles with Nine or Fewer Terms

Fig. 36 Title Length in Number of Terms and Characters Distinguished by URI Found and Not Found



(a) Stop Titles and Total Number of Terms

(b) Stop Title Characters and Total Number of Characters

Fig. 37 Upper Bounds for Ratios of Number of Stop Titles in the Title and Total Number of Terms in the Title and Number of Stop Title Characters and Total Number of Characters in the Title

Table 20 Confusion Matrix for Stop Titles / Total Number of Words

		Actual	
		Found	**Not Found**
Predicted	**Found**	66.0%	0.42%
	Not Found	28.27%	5.28%

Table 21 Confusion Matrix for Number of Characters in Stop Titles / Total Number of Characters

		Actual	
		Found	**Not Found**
Predicted	**Found**	66.45%	0.0%
	Not Found	28.48%	5.07%

4 SUMMARY

In this chapter we evaluate the retrieval performance of web pages' titles as another implementation of the *rr* function and compare it to lexical signatures. We further investigate the evolution of titles over time and analyze the characteristics of poorly performing titles.

The results of using titles fed into *c2u* shown here lead us to the conclusion that titles of web pages are a strong alternative to lexical signatures. Almost 70% of all URIs have been returned as the top result from the Google search engine API when queried with their non-quoted title. Analyzing the top 10 results for all queries we found that regardless of whether the URI was discovered, potential URI aliases and duplicates were returned which also supports the argument that titles form well performing queries. Considering that we did not address issues such as URI canonicalization in this chapter we see our retrieval results as a lower bound.

However, our results also show that a sequence of *c2u* iterations with different input performs best. Querying the title first and then using the 5-term lexical signature for all remaining undiscovered URIs against Yahoo! provided the overall best result with 75.7% of top ranked URIs and another 9.1% in the top 10 ranks. Even though the sequence 7-term lexical signature, title, 5-term lexical signature returned 76.4% of the URIs in the top ranks we recommend the former sequence since titles are far cheaper to obtain than lexical signatures. Since titles are commonplace and we have shown that they perform similarly well compared to lexical signatures our recommended strategy is to query the title first and if the results are insufficient generate and query lexical signatures second. Yahoo! returned the best results for all sequences and thus seems to be the best choice even though Google returned better results when feeding only the title into *c2u*.

We have used copies of web pages from the Internet Archive to confirm our intuition that titles decay over time. We have provided evidence that the content of web pages not only changes more quickly but also more significantly than their titles. For example, the results show that for a four year old title we have an almost 50% chance that it has not changed at all or has changed only slightly. In comparison, in Chapter V we have seen that lexical signatures older than five years perform poorly with an nDCG value of below 0.5. We therefore argue that titles can be a more robust *rr* function implementation and a promising input for *c2u* with the intention to rediscover missing pages.

Lastly, we have shown that not all titles are "good titles", at least as an *rr* implementation. With a thorough analysis of the composition of all titles in our data set we have provided a guideline to automatically identify titles predicted to perform poorly for our purpose. We have distilled a list of "stop titles" that indicate the title's retrieval quality. Due to our results we conclude that if 75% or more of a given title are stop titles it is predicted to perform poorly.

Given these results we have found two solid methods to rediscover missing web pages, created a guideline of how to sequence them and became aware of their weaknesses. All these findings are utilized in Synchronicity introduced in Chapter X.

CHAPTER VII

TAGS

1 BACKGROUND

The previously introduced methods, namely a web page's title and the page's lexical signature, have both shown to perform very well for the purpose of rediscovering missing web pages. However, both methods are applicable only if a Memento of the missing page can be found. If that fails, meaning if we can not obtain a Memento for the missing page, we have no means to gain knowledge of its "aboutness" and we can not execute the rr function.

In this chapter we explore a third option. We investigate the retrieval performance of tags as an rr implementation. In particular we analyze tags created by Delicious users to annotate URIs. We see several promising aspects for using tags: unlike titles and lexical signatures tags may be available even if no old copy of a missing page can be found. That means even if we can not obtain the title or generate the lexical signature of the missing page we may find tags describing its content. Tags are created by many users, therefore represent the "wisdom of the crowd". They have been shown to be useful for search [71, 126] and shown to possibly contain terms that do not occur in the original (now missing) web page. Therefore tags could be suitable input for the $c2u$ function and could be beneficial for retrieving other, potentially relevant documents.

We do not expect tags to outperform titles and lexical signatures but we foresee an added value for the rediscovery of missing web pages in combination with the previously established sequences of $c2u$ iterations.

In previously generated corpora containing randomly sampled URIs we observed that tags were very sparse. In the corpora used for the experiments in Chapters V and VI which are also described in [161], for example, we only found tags for 15% of all URIs. This led us to the creation of a new, "tag-centric" corpus introduced here.

2 EXPERIMENT SETUP

2.1 Data Gathering

Heymann et al. [126] support the point that tags are very sparse in datasets based on randomly sampled URIs. They show that compared to a search engine's index the number of URIs annotated with tags is diminishing. Therefore we decided to reverse the approach and obtain tags and the URIs they annotate instead of first sampling URIs and then asking for their tags hoping to get a good sized sample set. Note that these URIs are not really missing but due to the sparseness of tags we use the obtained URIs and pretend they are missing. A few sources are available to obtain tags left by users to annotate URIs. The website `delicious.com` is probably the most famous and most frequently used one. We queried Delicious for 5000 unique URIs from their index using the Delicious "random tool"[1]. We are aware of the bias of our dataset towards the Yahoo! index (which

[1] `http://www.delicious.com/recent/?random=1`

Table 22 Tag Distribution for URIs Sampled from `http://www.delicious.com`

# of Tags	0	1-5	6-10	11-15	16-20	21-25	26-29	30
Frequency in %	0.42	1.44	2.36	4.48	6.66	6.86	4.04	**73.73**

we query against) especially in the light of Yahoo! integrating Delicious data into their index [8]. However, sampling from Delicious is an approach taken by various researchers [71, 126].

We eventually aggregated 4968 unique URIs from Delicious. We did get 11 duplicates and despite the fact that we sampled from the "random tool" which pulls from the Delicious index we obtained 21 URIs that did not have tags. We used screen scraping, instead of the Delicious API, to gather up to 30 tags per URI[2]. The order, which may be of relevance for web search, indicates the frequency of use for all tags. Table 22 shows the relative distribution by number of tags for all URIs. We obtain the maximum of 30 tags for almost three out of four URIs.

2.2 Performance Measure

We use the Yahoo! BOSS API for all queries and analyze the top 100 results. We apply three different performance measures for our evaluation. Since our dataset consists of live URIs one way of judging the performance of tag based search queries is to analyze the result set and monitor the returned rank of the URI of interest. This establishes a binary relevance case. More precisely, similar to our evaluation in [161] the first performance measure distinguishes between four retrieval cases where the returned URI is:

1. top ranked
2. ranked 2-10
3. ranked 11-100
4. considered undiscovered (ranked 101+).

We consider URIs not returned within the top 100 as undiscovered. We are aware of the possibility of discriminating against results returned just above that threshold but as mentioned earlier it is known that the average user does not look past the first few search results ([54, 142]). We also compute normalized Discounted Cumulative Gain (nDCG) for the result set as a measure to reward results at the top of the result set and penalize results at the lower end. We give a relevance score of 1 for an exact match of the target URI and a score of 0 otherwise. For comparison reasons we also include mean average precision (MAP) scores for our results with the same binary relevance scoring.

Secondly, we compute the Jaro-Winkler distance between the original URI and the top 10 returned URIs from the result set. The intuition is that some highly relevant pages have very similar URIs. The Jaro-Winkler distance is frequently applied to measure the similarity between short string such as names and is therefore well fitting for comparing our URIs.

[2] We have previously shown the Delicious API to be unreliable, see:
`http://ws-dl.blogspot.com/2011/03/2011-03-09-adventures-with-delicious.html`

Table 23 Relative Retrieval Numbers for Tag Based Query Lengths, nDCG and MAP

# of Tags	**Top**	**Top10**	**Top100**	**Undis**	**Mean nDCG**	**MAP**
4	7.2	11.3	9.6	71.9	0.14	0.11
5	9.0	11.3	**9.7**	69.7	0.16	0.13
6	9.7	**12.0**	9.0	**69.3**	0.17	0.14
7	10.5	11.5	8.7	**69.3**	**0.18**	0.14
8	**11.0**	10.8	8.1	70.1	**0.18**	**0.15**
9	10.3	9.9	8.0	71.9	0.17	0.14
10	9.7	8.9	6.4	75.0	0.15	0.13

As a third measure, we compute the Dice coefficient between the content of the original page and the content of the top 10 search results. This gives us a sense of the string based similarity between the original content and the returned results. A high coefficient means a high similarity which in turn can be interpreted as a high relevance to the query – the tags used to annotate the original URI.

3 RETRIEVAL PERFORMANCE OF TAGS

3.1 Length of Tag Based Search Queries

We determined the best performing lexical signature length in previous work [158] and as shown in Chapter V to be 5 and 7 terms. We initially assumed these parameters could be equally applied to tags. Hence our input to the *c2u* function consisted of 5 and 7 tags. It turns out our assumption was inaccurate and therefore we widened the spectrum. Table 23 shows query lengths varying from 4 to 10 tags and their performance in relative numbers with respect to our four retrieval categories introduced in Section 2.2 as well as their nDCG and MAP. The generally low mean nDCG and MAP values are due to the large number of undiscovered URIs. Table 23 shows that 8-tag queries return the most top ranked results (11%) and 7-tag queries, tied with 6-tag queries, leave the fewest URIs undiscovered. Is also shows that 7- and 8-tag queries are tied for the best mean nDCG while 8 tags have a slight edge at MAP. However, taking this data we can not find a statistical significance (p-value ≤ 0.05) between the performances of 5-, 6-, 7- and 8-tag queries. The performance of 4-, 9- and 10-tag queries is in comparison statistically significantly worse.

3.2 Relevance of Results

Our binary retrieval evaluation (the URI is either returned or not) is applicable since we know what the "right" result to the tag based query is – the URI. However, the results in Table 23 indicate that a large percentage of URIs remain undiscovered. Next, we investigated the relevance and similarity of the returned results for cases where the URI of interest is not returned.

URI Similarity

We compute the Jaro-Winkler distance between the original URI and the URIs of top 10 results to determine the similarity between URIs. Given the data from Table 23 we take the results of the

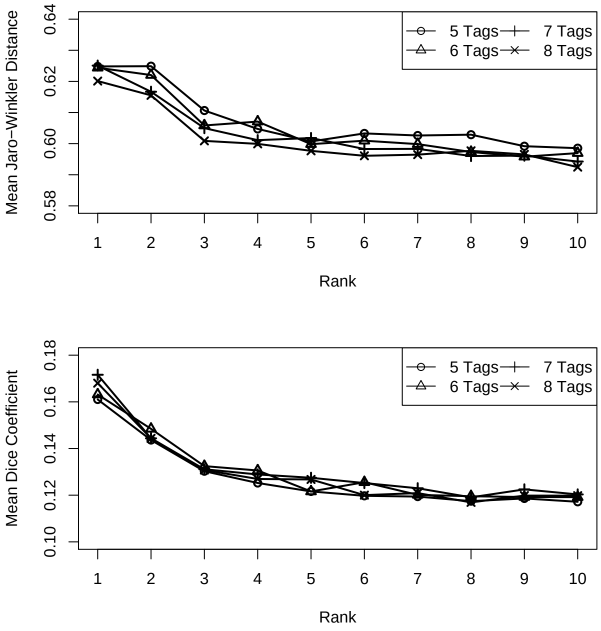


Fig. 38 Similarity Between URIs and Contents

five best performing tag based query lengths (5, 6, 7 and 8 tags) for this analysis. Figure 38 shows in the top graph the mean Jaro-Winkler distance for all URIs (y-axis) per rank (x-axis). Each of the four lines represent a query length but it seems insubstantial to distinguish between them. The mean Jaro-Winkler value is high. It varies between 0.59 and 0.62 with slightly higher values for the top two ranks. The values for ranks three through ten are almost indistinguishable. These results show very similar URIs in the top 10 indicating a high degree of relevancy for the returned results.

Content Similarity

Figure 38 shows in the bottom graph the Dice coefficient between the content of the URI the tags were derived from and the content of the top ten results. The intuition is that tags may not have the specificity to reliably return their URIs but contain enough information to return other relevant pages. This can especially be true for tags that do not actually occur in the pages. The graph also distinguishes by query length but the differences are diminishing. The mean Dice coefficient varies between 0.12 and 0.17. It is highest for the top two ranks and slightly decreases with higher ranks. The low mean Dice coefficients give an indication for a small degree of string similarity for the obtained results.

3.3 Performance Compared to Content Based Queries

In order to give a comparison for the performance of tags we also apply the previously introduced *rr* implementations, the title of the page and the page's lexical signature, and feed them into *c2u*. Table 24 summarizes the *c2u* outcome distinguished by our four retrieval cases, nDCG and MAP.

Table 24 Relative Retrieval Numbers for Titles, Lexical Signatures and Tags, nDCG and MAP

	Top	**Top10**	**Top100**	**Undis**	**Mean nDCG**	**MAP**
Titles	**60.2**	4.2	0.6	**34.9**	**0.63**	**0.62**
LSs	36.5	6.6	1.3	55.6	0.4	0.39
Tags	22.1	**15.4**	**10.2**	52.4	0.32	0.27

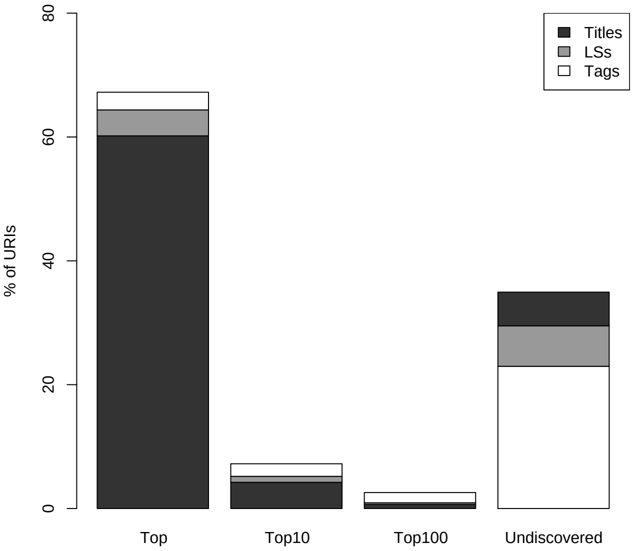
Note that the data in Table 24 is based on aggregated values meaning we merged the results for 5- and 7-term lexical signatures into one category and likewise for all tag based query lengths. We can see that titles outperform lexical signatures, supporting our earlier findings in Chapter VI and in [161, 162]. Both methods perform better than tags in terms of URIs returned top ranked, mean nDCG and MAP even though tags leave slightly fewer URIs undiscovered than lexical signatures. Tags return much more URIs in the top 10 and top 100 than any other method. One interpretation of this observation is that tags, possibly rather generic by nature, are often not precise enough to return the URI top ranked. but they do provide enough specificity to return the pages within the top 100 results. This can be beneficial in case the other methods do not return the URI of interest. This observation can be compared to the performance of lexical signatures generated with the first method introduced in Park et al. [214]. Following this method the generated lexical signatures contain terms in decreasing TF order only and IDF values are not considered. These TF-biased lexical signatures show the worst performance in returning the URI as the only result.

3.4 Combining Tags With Other Methods

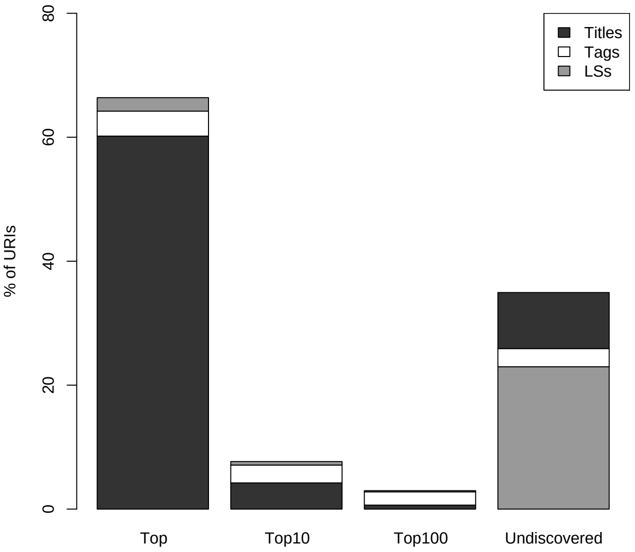
Tables 23 and 24 show that the overall retrieval performance of tags alone is not very impressive. However, what these tables do not show is the value of tags in a sequence with other *rr* implementations. In other words, does the union of the results of more than one method improve the retrieval performance? And speaking from the preservation point of view, can we rediscover more missing URIs with combining two or even all three of the methods?

Extracting a web page's title from the content is cheap; it costs just one request to the resource. In case the resource is a Memento it entails two requests: one to a Memento aggregator and the second to the archived resource itself. Lexical signatures are much more expensive to generate since each term, as a candidate to make it into the signature, requires the acquisition of a document frequency value. That means one request per unique term. Additionally we need to compute and potentially normalize term frequency (TF) values. Obtaining tags, similar to titles, is very cheap because it only requires one request to Delicious.

With this "cost model" in mind we define two sequences of *c2u* iterations with varying input: *Title-Lexical_Signature-Tags* (*T-LS-TA*) and *Title-Tags-Lexical_Signature* (*T-TA-LS*). Since titles perform best (as shown in Chapter VI and also demonstrated in previous work [161]) we maintain the priority for titles and query them as our first step in both sequences. As our second step in *T-LS-TA* we apply the lexical signature based method to all URIs that remained undiscovered (34.9% as shown in Table 24). We thirdly apply the tag based method to all URIs that are still undiscovered in *T-LS-TA*. The difference in the second sequence is that we apply the tag based method second



(a) Title-Lexical Signature-Tag Sequence



(b) Title-Tag-Lexical Signatures Sequence

Fig. 39 Performance of Titles Combined with Lexical Signatures and Tags

Table 25 Mean nDCG and Mean Average Precision for all Sequences of Methods

	TI	**TI-LS**	**TI-LS-TA**	**TI-TA**	**TI-TA-LS**
Mean nDCG	0.63	0.67	0.72	0.69	0.71
MAP	0.62	0.67	0.70	0.67	0.69

(to the 34.9%) and the lexical signature based method third.

Figure 39 shows the combined retrieval performance. The data of sequence *T-LS-TA* is shown in Figure 39(a) distinguished by contribution per method and separated in the previously introduced four retrieval categories. The first three bars (from left to right) are additive meaning the darkest part of the bars corresponds to the relative number of URIs returned by titles, the gray portion of the bars corresponds to the URIs not returned by titles but returned by lexical signatures and the white part of the bars represents the URIs neither returned by titles nor by lexical signatures. They are returned by tags only. Therefore these three left bars are to be read as if they were growing with the application of each additional method. The rightmost bar is to be read as if it was subtractive. For Figure 39(a) that means the dark portion of the bar represents the number of URIs undiscovered with titles (34.9%). The upper bound of the dark portion down to the upper bound of the gray portion represents the retrieval gain due to applying the second method. The height of the white portion of the bar corresponds to the final number of URIs that are left undiscovered after applying all three methods (23%) in the sequence *T-LS-TA*. Figure 39(b) displays the data in the same way for the sequence *T-TA-LS*. The color scheme remains the same with respect to the method meaning dark is still the title, gray still the lexical signature and white still represents tags.

The height of the gray bar for undiscovered URIs is of course identical to the corresponding white bar in Figure 39(a). The additive bar for the top ranked results is slightly higher in Figure 39(a) (67.2% vs. 66.4%) but the bars for the top 10 and top 100 results are slightly higher in Figure 39(b) (7.2% vs. 7.7% and 2.6% vs. 3.0%). The results for the sequence of methods in terms of mean nDCG and MAP are summarized in Table 25. The performance increase of both sequences is statistically significant as determined by the t-test with p-values below the 0.05 threshold. Tags perform similarly compared to lexical signatures for URIs that remain undiscovered with the title method. Since tags are so much cheaper to obtain than lexical signatures these results lead to the recommendation to use tags as the default secondary method for rediscovering missing web pages in case tags are available through Delicious. This condition is crucial since we have seen that tags were rather sparse for previously analyzed web page corpora.

4 GHOST TAGS

Previous research [71, 126] has shown that about half the tags used to annotate URIs do not occur in the page's content. We find a slightly higher value with 66.3% of all tags not present in the page. If we consider the top 10 tags only we find 51.5% of the tags not occurring in the page. This discrepancy intuitively makes sense since the ranking in Delicious is done by frequency of use which means that less frequently used tags are more likely to not appear in the page. However, these numbers only apply for the current version of the page. The tags provided by Delicious on the other hand are aggregated over an unknown period of time. The date of tags in Delicious can only be approximated but not reliably computed. It is possible that some tags used to occur in a previous version of the page and were removed or replaced at some point but still are available for that page through Delicious. We call these "ghost tags", terms that persist as tags after disappearing from the document itself.

To further investigate this aspect we use the Memento framework [259] to obtain old copies for all URIs that have tags not occurring in their content. For our dataset that applies to more than 95% of the URIs. Since we obtain different amounts of Mementos and different ages of the Mementos, we decided to only check tags against the first Memento meaning the oldest available copy of the page. We obtain Mementos of $3,306$ URIs some of which date back to 1996. Out of the 66.3% tags not present in the current page we find a total of 4.9% being ghost tags. They occur in about one third of the previous versions of our web pages. Figure 40 displays the distribution of tags (dark gray) and the Mementos they occur in (light gray) per year. Note that the y-axis is plotted in log-scale. The vast majority of our "ghost tags" is found in Mementos from recent years especially in 2009. Only a few if any at all are found prior to 2006. We also see noticeable numbers from 2011 which indicates a very short time between the publication of the tags at which time they did occur in the page and their disappearance from the page. The majority of these very recent Mementos were obtained from search engine caches. The observations from Figure 40 confirm 1) that ghost tags exist meaning some tags better represent the past content of a web page than the current and, 2) these ghost tags are found in the more recent past and rarely date back more than three years.

We then determine the importance of ghost tags for a page. We compare the tags' occurrence frequency in Delicious and their term frequency (TF) in the first available Mementos. We rank each ghost tag according to its Delicious and its TF rank and normalize the rank to a value between zero and one in order to avoid a bias towards a greater amount of available tags and longer documents. The closer the value gets to zero the higher the rank and the greater the importance. Figure 41 displays the Delicious rank on the x-axis and the TF rank on the y-axis. Each dot represents one ghost tag. If a dot is plotted more than once, its shade gets darker (18 dots are plotted twice, one three times and one five times). The semi-transparent numbers indicate the percentage of dots or ghost tags in the corresponding quadrants. The numbers confirm our first visual impression of the graph. A majority of ghost tags (34.7%) occur in the first quadrant meaning their normalized Delicious rank is ≤ 0.5 and so is their TF rank. This indicates a high level of importance of the ghost tags for the document and also for the Delicious user. One fourth of the ghost tags seem to be more important for the document than in Delicious since their ranking there is > 0.5. On the other hand for 22% of ghost tags the inverse holds true. In 18.1% of the cases we can claim that "only" infrequent terms became ghost tags. These results show the significance of ghost tags since one third of them were used very frequently in the document and still are used frequently in Delicious.

5 SUMMARY

In this chapter we investigated tags created by users to annotate URIs as a third rr implementation and as input to the $c2u$ function.

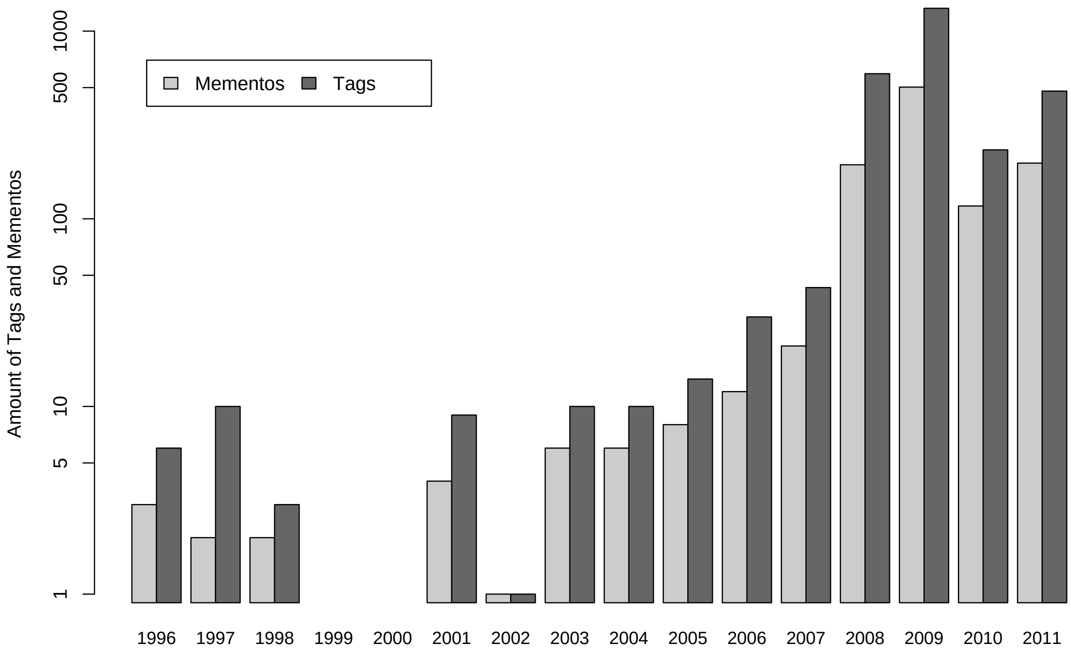


Fig. 40 Amount of Ghost Tags and Mementos Occurring in Previous Versions of Web Pages

Due to the sparsity of tags in the previously used datasets containing randomly sampled URIs we here used a set of URIs obtained directly from Delicious, the source of our tags. We investigated, similar to previous chapters, the length of well performing search engine queries based on tags and compared them with the previously introduced methods. We showed that a search engine query containing five to eight tags performs best. More than 20% of the URIs are returned in the top 10 ranks. We have further provided evidence for the top 10 results to be similar to the URI the queried tags were obtained from. But compared to querying the title of the page or its lexical signature tags alone do not perform well.

However, we here again find that a sequence of *c2u* iterations with tags and others methods can increase the overall retrieval performance. Due to the fact that titles perform best comparing all methods and given the low cost of obtaining a web pages' title we still consider titles to be our primary method. As a secondary method we have seen that both lexical signature and tags show similar performances. It further is a fact that tags are cheaper to obtain compared to the rather complex and time consuming creation of lexical signatures. Therefore we consider tags, if available, as our secondary method for the rediscovery of missing pages.

We have further introduced the notion of "ghost tags". Ghost tags are what we call terms obtained from Delicious that do not occur in the current version of the web page they are annotating but they do occur in a previous version of that page. Of course ghost tags are not restricted to

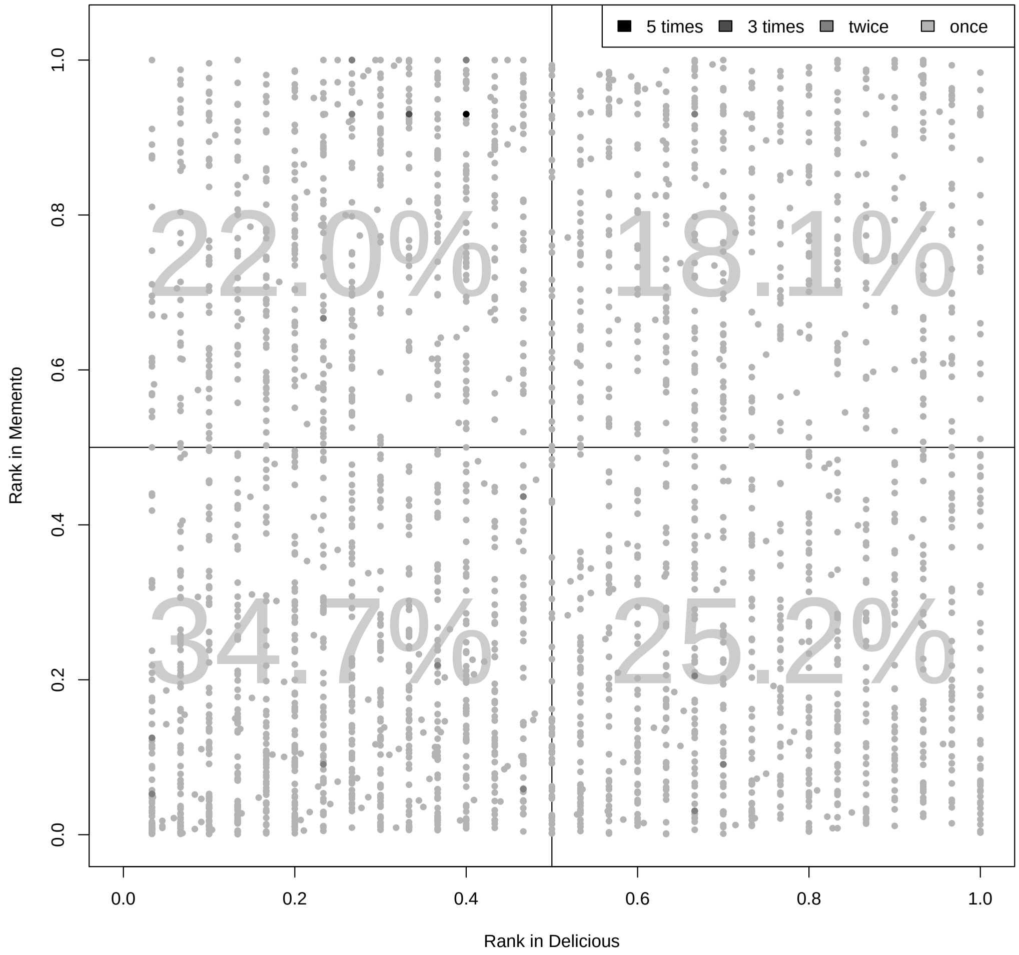


Fig. 41 Ghost Tags Ranks in Delicious and Corresponding Mementos

Delicious. We just chose to obtain tags from this one source but we do suspect ghost tags to be a ubiquitous phenomenon throughout various sources for tags.

We provided evidence that ghost tags are not just any terms, neither for the user nor for the document. More than one out of three ghost tags appear to be important for the user as well as for the document. We assessed the importance by analyzing the term's ranking in Delicious (determined by frequency of use) and its frequency of occurrence in the text.

The results of this chapter provide us with a guideline to implement tags as yet another method to rediscover missing pages in our software system. In particular we gained knowledge about the length of tag based queries. This method becomes particularly important if no Mementos of a missing URI can be obtained, which means no titles are available and we can not create a lexical signature.

CHAPTER VIII

LINK NEIGHBORHOOD LEXICAL SIGNATURES

1 BACKGROUND

Two of the *rr* implementations as input for the *c*2*u* function for rediscovering missing web pages introduced so far rely on obtained Mementos. The Memento's content is used to generate a lexical signature (Chapter V) and we extract its title (Chapter VI). If no Memento is available, both methods are unusable. In this chapter we investigate link neighborhood lexical signatures (LNLS) as a fourth *rr* function implementation. Like tags introduced in Chapter VII LNLSs can be used if no Mementos of a missing page are available. A LNLS of a web page is a lexical signature generated from the content of other web pages that link to the page of interest, also called their inlinks or backlinks.

We performed an experiment to evaluate constructing lexical signatures from link neighborhoods as seen in Klein et al. [163]. Since pages tend to link to related pages, our intuition is that the link neighborhoods contain enough of the "aboutness" of the targeted page to create a valuable *rr* implementation that also can be fed into the *c*2*u* function.

We constructed link neighborhoods by querying a search engine for listings of backlinks and tested several methods of calculating lexical signatures from those link neighborhoods to find the most effective signature based implementation.

Similar to Chapter V we identify in this chapter optimal values for LNLSs. We examine the effects of lexical signature size, backlink depth and backlink ranking as well as the radius within a backlink page from which terms for the link neighborhood lexical signature will be drawn.

2 CONSTRUCTING THE LINK NEIGHBORHOOD

We anticipated a large number of backlinks, which made us use the same corpus of 309 URIs introduced in Chapter V, Section 3.2 for our experiment.

For each URI, we queried the Yahoo! BOSS API to determine the pages that link to the URI ("backlinks"). We chose the Yahoo! BOSS API because it was previously shown to give more complete backlink results than other search engines [192]. We refer to the order in which these backlinks are returned as "backlink rank". By obtaining the backlinks of the backlinks we created a directed graph of depth two. Figure 42 graphically explains such a link neighborhood. The page on the right with the vertical lines represents the target page which is the page that a user linked to but is it no longer available. Using Yahoo!, we obtain (in this example) three pages that link to the target page. These are the first-level backlinks, represented in the center with horizontal lines. For each first-level backlink, we obtain its backlinks. These pages are represented with crossing lines and we call them second-level backlinks. In this manner we retrieved a total of $335,334$ pages, $28,325$ first-level and $306,700$ second-level backlink pages.

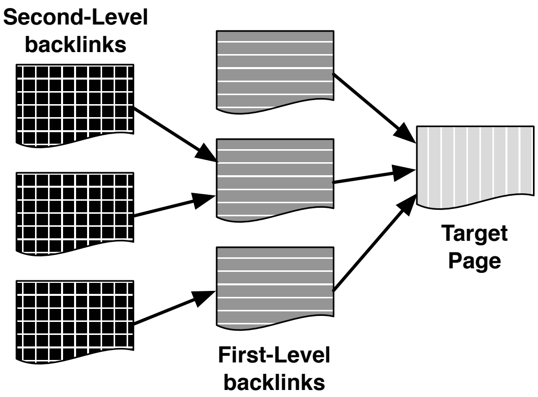


Fig. 42 Graphical Example for a Link Neighborhood

2.1 Pre-processing the Link Neighborhood

With a dataset of that size we need to guard against noisy inputs. We applied five filters to the backlink page representations before we calculated lexical signatures. Those filters were based on: content language, file type, file size, HTTP response code, and the presence of "soft 404s". Some of the filters have been used earlier, in particular in the local universe dataset first introduced in Chapter IV, Section 3.2. All applied filters are also described in our technical report [264].

Language

Since we used an American search engine which is likely to have coverage biased towards English-language pages [262] we made an attempt to discard non-English pages before the lexical signature calculation. We again use the Perl module *Lingua::Identify* to obtain percentage guesses for each page's language and dismissed less than 8% of all pages.

File Type

We discarded less than 4% of all pages due to their non-HTML representations. In cases where the server described the file type in the HTTP headers, this information was trusted. In other cases, the Unix *file* command-line program was used to guess the file type.

File Size

Similar to the filter applied earlier we discarded any pages that contained fewer than fifty terms after rendering to ensure enough input for a robust lexical signature. This filter accounted for dismissing less than 7% of all pages.

HTTP Response Code

We experienced a variety of errors while downloading all web pages. Fortunately almost 97% of pages returned the HTTP response code 200, which means success. We made a total of five attempts

Table 26 Pre-Processing Statistics

Filter	**Percent of URIs Discarded**
Content Language	7.36%
File Type	3.82%
File Size	6.86%
HTTP Result code	3.25%
Soft 404s	0.36%
Remaining	78.35%

to download the remaining pages. If none of the attempts resulted in success, those pages were dismissed.

Soft 404s

Pages that terminate with an HTTP 200 status code (possibly after multiple redirects) while returning an error message in the human-readable version of the page like "your page could not be found" are known as "soft 404s". Soft 404s have been introduced in Chapter III, Section 2.4. Since these pages are not "about" the same things as the pages that they link to, they were discarded (less than 1%). To determine if a page was a soft 404, we used a subset of the method laid out by Bar-Yossef et al. [65]. We retrieved the URI, plus two modified versions of the URI with a random 20-character string inserted. First, we inserted the random string immediately after the first slash after the domain name. Second, we inserted the random string immediately after the last slash in the path. For example, if the original URI was:

```
http://example.com/path/to/file.html
```

and the random string was XXXXXXXXXXXXXXXXXX, then we would have downloaded also

```
http://example.com/path/to/XXXXXXXXXXXXXXXXXXfile.html
```

and

```
http://example.com/XXXXXXXXXXXXXXXXXXpath/to/file.html
```

We used the Jaccard similarity coefficient to determine the similarity of the representation returned for the original URI and both modified URIs. If the original URI returned an HTTP error code, meaning anything other than a 200-level success code, then we could declare that the page was not a soft 404. If the first modified URI returned a HTTP error code, then the page could not be a soft 404, since malformed URIs in that path trigger real (hard) 404 responses. Otherwise we calculated the Jaccard similarity between the representation of the original URI and the first modified URI as well as the Jaccard similarity between the original URI and second modified URI. If both similarity coefficients were above our threshold of 0.9, then the original URI was deemed to be a soft 404 and discarded. Table 26 summarizes the percent of URIs that we caught in each pre-processing filter. We caught some URIs in more than one filter, but only list the filter here in which they were caught first.

3 CALCULATION OF NEIGHBORHOOD LEXICAL SIGNATURES

We seek to determine the effects of lexical signature size, backlink depth, backlink ranking, as well as the radius within a backlink page from which terms for the lexical signature should be drawn. For every possible combination for each of these factors we computed the TF-IDF value of every term in the appropriate section(s) of the appropriate pages. As usual the terms with the highest TF-IDF value are chosen for the lexical signature.

3.1 Backlink Depth

The two options for depth were:

1. to use only the first-level backlinks, those that link directly to the target page (marked with horizontal lines in Figure 42) and

2. to use first and second level backlinks (horizontal and crossing lines in Figure 42).

First-level backlinks might result in a lexical signature that more accurately describes the missing page since they are closer to the target page. However, in cases where few first-level backlinks exist, second-level backlinks might provide more information, leading to a better performing lexical signature.

3.2 Backlink Ranking

The backlinks returned from the Yahoo! BOSS API are ordered. To determine whether this ranking was helpful we tested the following three possibilities:

1. using only the top 10 backlinks,

2. using the top 100 backlinks and

3. using the top 1000 backlinks.

If fewer backlinks existed than allowed by the limit we use all available backlinks. If the rankings in backlink results from the BOSS API were helpful, then using only the top backlinks would provide a better lexical signature. If not, then using as many backlinks as possible might provide the better lexical signature since that means including more data.

3.3 Radius

We considered four possibilities for the radius within the backlink page from which lexical signatures would be drawn. lexical signatures are typically drawn from entire pages, and this was our first possibility. However, since a particular section of a page can be about a different topic than a page as a whole, we tested whether using only the relevant portions of a page would produce a better lexical signature. To find the “relevant” portion of a backlink page we used the link from the page to the target URI as a centerpoint and captured the ‘paragraph’ of context around the link. Such a link should exist, since the backlink was described by the BOSS API as a page that linked to the target URI. Hence the second option considered the anchor text plus the preceding five words and the following five words. We increased the radius around the centerpoint for the third option to include the link plus ten words on each side. The fourth option used only the anchor text itself. In cases where a given backlink page included multiple links to the same target URI, the text around every link was included. In cases where we did not find the link to the target URI, the backlink page was not included in the calculations.

3.4 Lexical Signature Size

We have previously shown that 5- or 7-term lexical signatures perform best. However, given that the lexical signatures in this experiment were being derived from a link neighborhood instead of the target page itself, we need to test the applicability of those standards. We stored the ten terms with the highest TF-IDF value and queried lexical signatures of sizes one, two, three, four, five, six, seven, and ten.

4 RESULTS

Just like for the evaluation of our results in Chapter V, Section 3.1 we use here again the normalized Discounted Cumulative Gain (nDCG). We set the relevance score to 1 for an exact match of the target URI, and 0 otherwise. We checked the first 1000 results and if the target URI was not found we assigned a nDCG value of 0, corresponding to an infinitely deep position in the result set.

4.1 Backlink Depth

Figures 43, 44 and 45 show average scores of methods based on anchor text, anchor text ±5 words and anchor text ±10 words, respectively, using the first- and second-level backlinks. First-level backlink methods are drawn in black and second-level methods in red. Since the method based on using the whole page performed exceptionally poor and its results do not change the overall outcome we do not include that graph. The x-axis is the number of terms included in the lexical signature and the y-axis is the mean nDCG. Note the dramatic decline in every case when second-level backlinks are included. This shows that second-level backlinks' relation to the target page is not tight enough to be useful in describing the target page. As such, our best-performing method includes only first-level backlinks.

4.2 Radius

We started with the assumption that some parts of a backlink page would use terms that are more closely related to the target page, and that the most relevant terms would be in or near the link to the target URI. As we can see in Figure 46 by far the best results arise from using only the terms in the anchor text itself to calculate the lexical signature. The anchor text ±5 words or ±10 words performed similar to each other and using the whole page performed the worst. Each step taken away from the anchor text, by broadening the radius to include words around the anchor or the entire page, yields increasingly poor results.

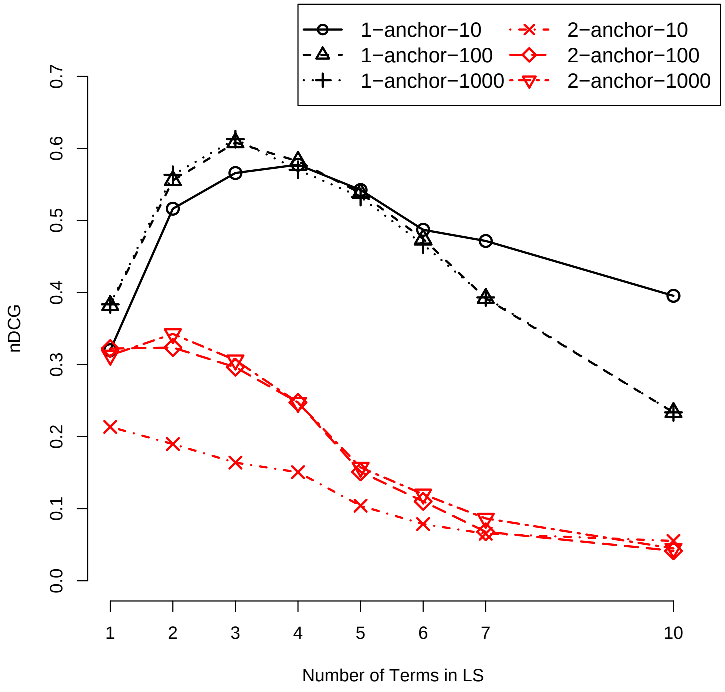


Fig. 43 First- and Second-Level Backlinks **Anchor** Radius Lexical Signatures with Various Backlink Ranks (shown as levels-radius-ranks)

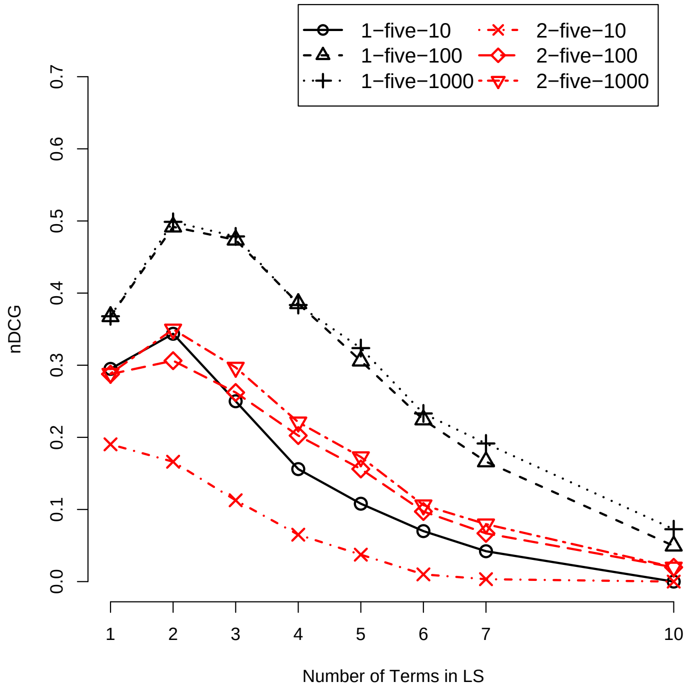


Fig. 44 First- and Second-Level Backlinks Anchor **Plus/Minus Five** Radius Lexical Signatures with Various Backlink Ranks (shown as levels-radius-ranks)

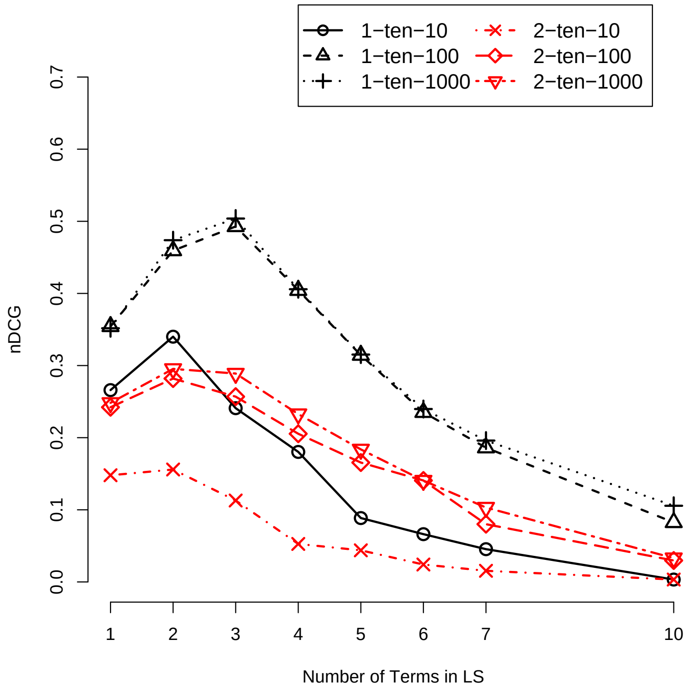


Fig. 45 First- and Second-Level Backlinks Anchor **Plus/Minus Ten** Radius Lexical Signatures with Various Backlink Ranks (shown as levels-radius-ranks)

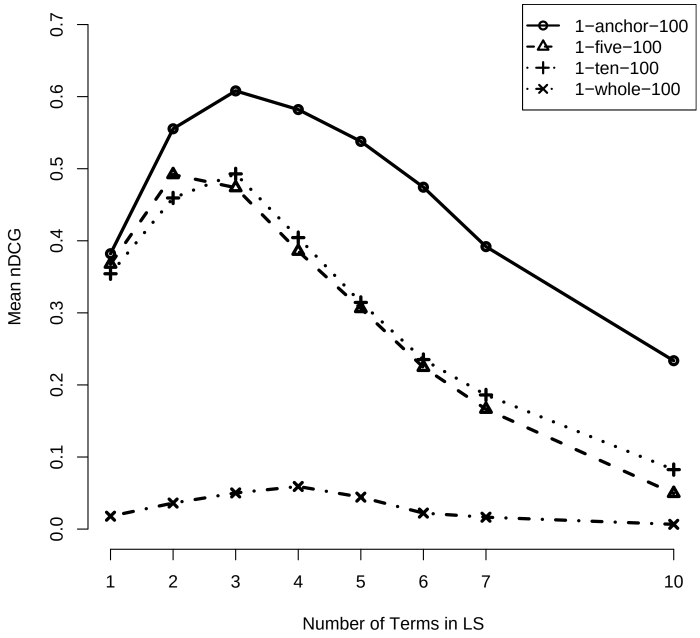


Fig. 46 Effect of Radius (First-Level Backlinks)

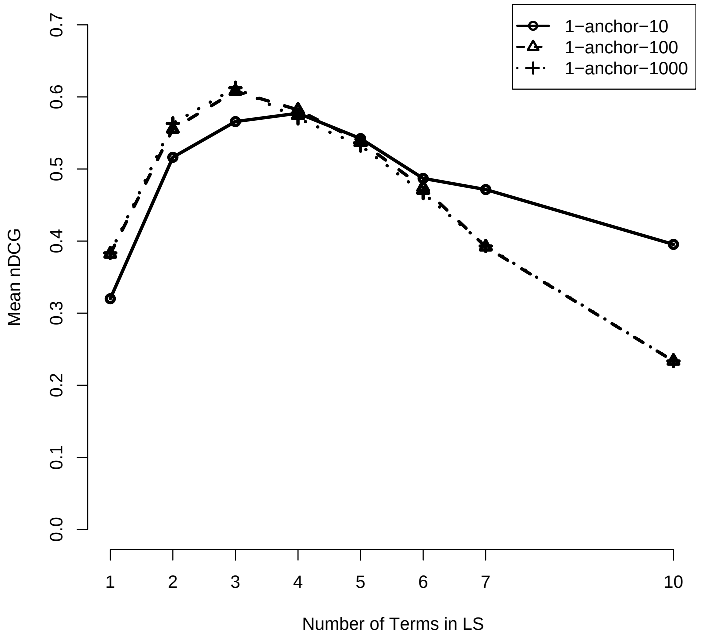


Fig. 47 Effect of Backlink Rank (First-Level-Backlinks)

4.3 Backlink Ranking

Figure 47 shows the three possibilities for backlink ranks: using only the top 10, top 100, or top 1000 backlinks. Note that using 100 and 1000 backlinks (and anchor text) results in the highest overall nDCG value of 0.61 obtained with 3-term lexical signatures. This accounts for more than 58% of all URIs returned as the top search result. The corresponding nDCG value using the top 10 backlinks only is 0.57. Considering this marginal delta in nDCG and the huge implied cost to acquire ten or one hundred times as many pages and generate a lexical signature based on an accordingly larger bucket of words, we consider using only the top 10 backlinks as the better tradeoff. The "return on investment" is better when sacrificing an nDCG drop of only 0.04. For lexical signatures of size greater than three terms the nDCG is equal or better using ten backlinks only. So while we do not exactly know how Yahoo! determines its ranking, we do know (considering all costs) that we are better off using only the top 10 backlinks of a URI.

4.4 Lexical Signature Size

We can further see in Figure 47 that the overall best performance is obtained using 3-term lexical signatures. However, with the above reasoning meaning to use the top 10 backlinks only, the best-performing lexical signature is four terms in length. Using ten backlinks, 4-term lexical signatures have an nDCG value of 0.58 and almost 56% of all URIs are returned top ranked. Using more terms yields poorer results. Using fewer terms and top 10 backlinks results in a slightly worse performance

Table 27 Result Rank and nDCG vs Lexical Signature Size (1-anchor-1000)

	# of terms in lexical signature							
Result Rank	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**10**
1	32.11	50.50	**58.19**	54.85	52.51	45.82	38.80	23.41
2-10	10.03	10.70	7.02	5.35	2.34	2.34	1.67	0.33
11-100	5.69	3.34	0.67	0.33	0.33	0.33	0.33	0.67
101-1000	4.35	1.67	0.00	0.00	0.33	0.33	0.33	0.00
> 1000	49.16	35.12	35.45	40.80	45.82	52.51	60.20	76.92
Mean nDCG	0.38	0.56	**0.61**	0.57	0.53	0.47	0.39	0.23

compared to top 100 and top 1000 backlinks. This result is noteworthy since we have previously found (as seen in Chapter V) that the best lexical signature size is five or seven terms. We consider the source of the terms that make up the lexical signature to be the reason for this disparity. In this method, the terms are drawn not from the target page itself, but from pages that link to it, which are likely to be "related". Using five or seven terms drawn from the backlink pages is likely to over-specify the backlink pages and their specific focus, rather than the content of the target page. The second lexical signature generation method introduced in Park et al. [214] shows a similar over-specification. These lexical signatures contain terms in decreasing order of their DF value only and TF is disregarded. By using fewer terms, we decrease the risk of including a term in the lexical signature that does not appear in the target page.

As explained before, the nDCG score represents the value of the result set to the user. To see the meaning of this score more clearly, Table 27 shows the percentage of URIs by their location in the result set. The data is obtained from using the first level backlinks, anchor text only and the top 1000 results since this combination accounts for the numerically best nDCG scores. Table 28 shows the same results for using the top 10 backlinks only, our preferred combination of parameters. The first row contains the percent of URIs that were returned top ranked in the result set. Each of these URIs received the maximum score of 1. The following four rows represent those URIs that were returned in the top 10 but not top ranked, on further pages, and those that were not rediscovered at all. The last row, for comparison, shows mean nDCG scores.

Note again that the best results were obtained using 3-term lexical signatures and the top 1000 backlinks. Over 58% of the missing URIs were found in the top spot of the result set. However, the top performance of our preferred method of using only the top 10 backlinks and 4-term lexical signatures accounted for almost 56% URIs returned at the top position. Less than 5% of the URIs were returned between rank one and 1000. These results confirm the "binary distribution" introduced in the previous chapter meaning the majority of the URIs are either returned top ranked or somewhere (if at all) beyond the top 1000.

5 SUMMARY

In this chapter we demonstrated the usability of link neighborhood based lexical signatures (LNLS) as our fourth implementation of the *rr* function that also can be an argument to the *c2u* function.

We worked with a previously used rather small corpus anticipating the size of the entire dataset to

Table 28 Result Rank and nDCG vs Lexical Signature Size (1-anchor-10)

	# of terms in lexical signature							
Result Rank	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**10**
1	25.08	45.15	52.51	**55.85**	52.84	47.83	46.49	39.13
2-10	9.03	9.70	7.02	3.34	2.01	1.34	1.00	0.67
11-100	8.03	4.68	2.01	0.67	0.67	0.33	0.33	0.33
101-1000	5.69	2.34	0.33	0.67	0.33	0.33	0.33	0.00
> 1000	52.17	38.13	38.13	39.46	44.15	50.17	51.84	59.87
Mean nDCG	0.32	0.52	0.57	**0.58**	0.54	0.49	0.47	0.40

increase dramatically when including all neighboring pages. We investigated these lexical signatures in terms of levels of backlinks to include, the importance of ranked backlinks, the radius of terms to include around the anchor text, and the length in number of terms.

The results for the backlink levels are clear: only the first level is to be considered. Including the second level washes away the context of the centroid page. Somewhat to our surprise using the anchor text only shows the best performance. We anticipated more gain by including five or ten words surrounding the actual anchor. Anchors are usually created manually and ideally very briefly describe the page the link is referring to which makes it a suitable search query. In addition it has been shown that anchor text is somewhat similar to queries generated by users for search engines.

The overall best performance is obtained by including the anchor text of the top 1000 backlinks and generating a 3-term LNLS. However, the implied costs are huge as we would have to download and parse 1000 pages and obtain DF value estimates for all terms. The more resource efficient alternative is to include just the top 10 backlinks and generate a 4-term LNLS. With respect to the bare retrieval numbers this setup performs second best but the difference is minimal. We therefore recommend using the "lightweight" setup since it seems to be the better trade-off. Given this outcome, we recommend the most resource efficient combinations of all parameters for the generation of LNLS. However, compared to the previously introduced rr implementations, LNLS are most complex to create as they require the most amount of queries against a search engine. Given that fact we do not consider any sequences of $c2u$ iterations including LNLS. We rather like to consider this implementation as a last resort for an $c2u$ input, in case all other iterations have failed.

Similar to tags link neighborhood based lexical signatures gain importance in case no Mementos are available. We still have a chance to obtain pages that link to the missing URI and create a query representing the aboutness of the page of interest.

CHAPTER IX

BOOK OF THE DEAD

1 BACKGROUND

In all previous experiments [158, 160, 161, 162, 163] we have been using data sets and corpora containing URIs that were not actually missing. We generated these sets of URIs by randomly sampling various sources and to evaluate our methods for the rediscovery of web pages we "pretended" they were missing. The main reason for this is the fact that there is no corpus of actually missing web pages available. Researchers in the area of information retrieval frequently utilize bounded data sets as introduced in Chapter III, Section 8 but even these URIs are not missing. They rather are often aggregated for a particular retrieval task such as identifying spam [39] or they are created to capture a particular fraction of the web such as the TREC BLOG08 test collection [38], which contains blog feeds.

In this chapter we introduce the *Book of the Dead*, a corpus of real missing web pages. We apply our methods to rediscover missing web pages to this corpus and evaluate the results with the help of the "Wisdom of the Crowds". We make the corpus available for researchers interested in conducting related experiments.

2 THE BOOK OF THE DEAD

One of the contributions of the Library of Congress [23] to web page preservation is aggregating and archiving pages related to certain topics. They apply web crawling techniques [57], specifically a method that is known as focused crawling [67, 68] in order to narrow the crawl to a particular topic. The objective is to rather go into depth than breadth. Focused crawling has been shown to be efficient for collecting large data sets specific to particular contexts [82, 221]. Using this approach the Library of Congress, for example, aggregated pages about various federal and state elections in 2004 and 2006 as well as web pages about the terrorist attacks in the US on September 11 in 2001.

In the process of crawling the Library of Congress encountered numerous web pages returning a response code that indicates some sort of error. Thankfully they agreed to sharing these URIs which enabled us to generate the Book of the Dead corpus.

The data set originally contained 1371 URIs that returned some response code other than 200 (meaning "OK"). We counted a total of 376 URIs returning a 400-level response code out of which 290 were 404 responses. A total of 36 URIs returned a 500-level response and the remaining 1031 URIs returned a recorded response code of 0. Since such a response code is not defined in the HTTP specifications [108] we chose to dismiss these URIs. We tested all URIs labeled with a 404 response and found a total of 57 now returning a 200 response. It is possible that the Library of Congress experienced transient errors or some other kind of temporary failures that caused them to record the URI as missing. However, the portion of the data set we are experimenting with contained 233 actually missing URIs all of which can be found in Appendix B.

2.1 The Aboutness of the Dead

Amazon offers a service called Mechanical Turk [1] which implements Human Intelligence Tasks (HITs). The concept of this "micro-task market" is that businesses, researchers and developers create HITs which individuals all over the world can work on. Creators of HITs benefit from the access to a distributed around the clock active workforce getting their tasks done in a short period of time. Workers on the other hand can work on their own schedule from a location they chose and of course can pick interesting tasks to work on. Creators of HITs pay the workers for completing the tasks (if satisfied) and Amazon gets a share as well. The Mechanical Turk is gaining popularity in various research communities. It, for example, has been shown to work efficiently and reliably for relevance assessment tasks [56] and user studies [155].

We utilize this service to gain knowledge about the content of the missing pages. We only gave the Mechanical Turk workers two pieces of information: the general topic of the set of URIs (elections and terror) and the URIs itself. We asked them to analyze the URI and give their best guess on what the page used to be about. This approach is based on utilizing the web and a virtually unlimited pool of users.

However, there are alternative methods for URI classification most of which are based on analyzing the content of the pages. Kan [148] introduced a different method. He uses the URI only to categorize web pages which is much faster since the pages do not need to be fetched an analyzed. As also shown in his later work together with Thi [149] they split the URIs into meaningful segments and identify their features. These features together with applied maximum entropy [66] methods enable them to model obvious patterns. They evaluate their approach with the WebKB corpus [43], a corpus commonly used for text classification experiments and show that it can perform as well as content based classification techniques.

Often an URI can give away the general idea of the content it dereferences. For example, it is not difficult to give a broad idea of what the URI `http://www.whitehouse.gov` is about. Most computer scientists will probably also know that the URI `http://www.cs.umass.edu` identifies. This holds true for political parties and candidates. For example, the URI `http://www.lptexas.org/2006/helm/` is taken from the Book of the Dead. Obviously it is about something happening in 2006 related to the state of Texas. We know the general topic of the corpus and can conjecture it is a political website for an election in 2006. Maybe we even know that "lp" stands for Libertarian Party and "Helm" is one of their candidates. The majority of the aggregated "aboutness" obtained from Mechanical Turk workers for this URI confirms this intuitive analysis (each bullet corresponds to an answer by a different worker):

- Texas 4th District Libertarian Party United States Congress House Elections Libertarian Party Texas Helm Kurt G. Political candidates United States Elections United States United States Politics and government 2001

- this web is about the analysing the votes

- This site is for Libertarian Party of Texas and contains news and events related to it.

- Libertarian party of Texas, candidate Helm for 2006 election

- Texas poker cards hold'em

Anticipating a broad spectrum of opinions in this task, we asked for ten guesses per URI assuming the aggregate of HITs would give a good idea of the "aboutness" of the missing pages. As expected, this process also returned less useful replies. For example, some workers did not seem to read the description and simply responded with "page is not working" or "dead link". However, the majority of responses seemed suitable and so we decided to manually filter the useless ones. Note that we acknowledge a potential bias here but we made sure that we did not judge the relevance of responses but rather just dismissed clearly misguided ones. This filter accounted for the set of responses to shrink by roughly one third. However, each URI remained with at least two descriptions from two different workers. The filtered and aggregated responses from the HITs together with its corresponding URIs are available for download [32].

3 REDISCOVER THE DEAD URIS

3.1 Applying Titles and Lexical Signatures

Considering all our methods, the application of titles and lexical signatures show the highest probability of success. Therefore we decided to apply both methods to the Book of the Dead corpus. We utilized the Memento framework and were able to obtain Mementos for 161 out of all 233 URIs. From the latest available Memento per URI we extract the title and generate the 5- as well as the 7-term lexical signature following the procedure described in earlier chapters. Out of the 161 Mementos 154 have a title and we are able to generate a 5- and 7-term lexical signature for 139 URIs. The remaining Mementos are not in compliance with the criteria for the generation of lexical signatures; mainly they do not contain more than 50 terms.

We query the applied methods against the Yahoo! search API and recorded the top 100 results. To assess the similarity and hence the relevance of the returned results we compute the Dice similarity coefficient introduced in Chapter II, Section 7.3 between the Memento of the missing URI and the top 100 returned results. Figure 48 displays the Dice coefficient for the 5-term lexical signature results in a matrix. All URIs are shown along the x-axis and the ranks of the corresponding results are shown along the left y-axis. The URIs are sorted by the mean Dice values over all ranks in decreasing order. The Dice values D are separated into four clusters which are represented as four different colors in the graph:

- $D = 0$ depicted in white
- $0 < D \leq 0.3$ in light gray
- $0.3 < D \leq 0.6$ in gray and
- $0.6 < D$ in black.

The light blue area mainly in the right part of the graph represents the URIs and ranks where no results were returned from the Yahoo! search API. Further the average Dice value over all ranks per URI is displayed with the green line and refers to the right y-axis. The red dots are representative for download errors meaning cases where our script was not able to grab the content of the search

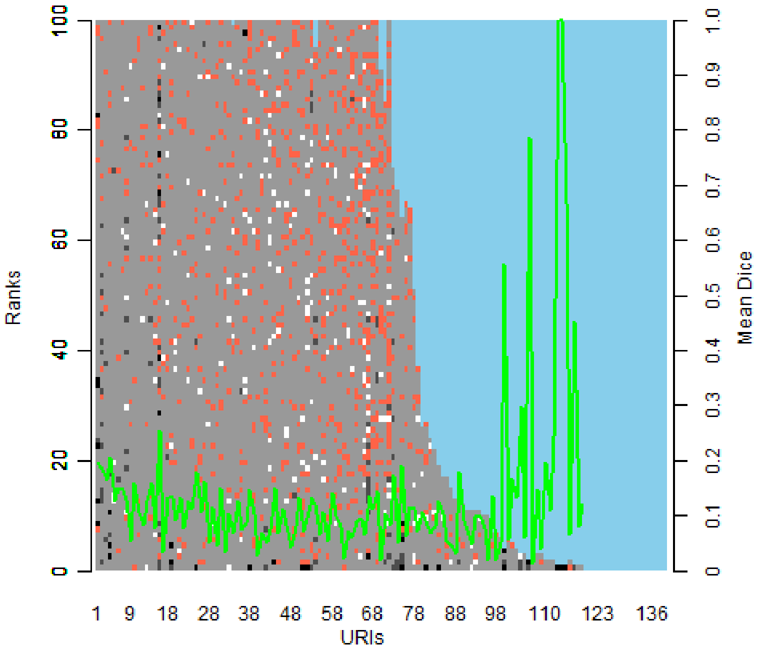


Fig. 48 Dice Similarity Coefficient for Top 100 5-Term Lexical Signature Results per Rank Including Mean Coefficient

result. The most likely reason being the download scripts do not accept HTTP cookies or send the same HTTP request headers as conventional web browser. Figures 49 and 50 show the same data for 7-term lexical signature and title queries, respectively.

The first obvious result is that almost all titles return some results whereas especially 7-term lexical signatures show no results for about 20 URIs. However, the majority of results that are returned for all three methods have fairly low Dice values. They fall in the second cluster meaning $0 < D \geq 0.3$ and are represented in light gray. Only a few dark gray and even fewer black results can be seen. The ones that do occur however are mostly ranked in the top 10 (close to the x-axis) which is not surprising since we would expect search engines to rank relevant results high. The average coefficient per URI confirms the picture of overall low Dice values. It varies greatly for URIs with fewer results returned. The few very similar results indicate cases where the missing page has moved to a different URI but the content has not changed dramatically. The dark gray and in the best case maybe even some of the light gray results indicate content-wise relevant pages. Overall we do not see a lot of completely dissimilar results (white dots) which supports the overall good performance of the two methods.

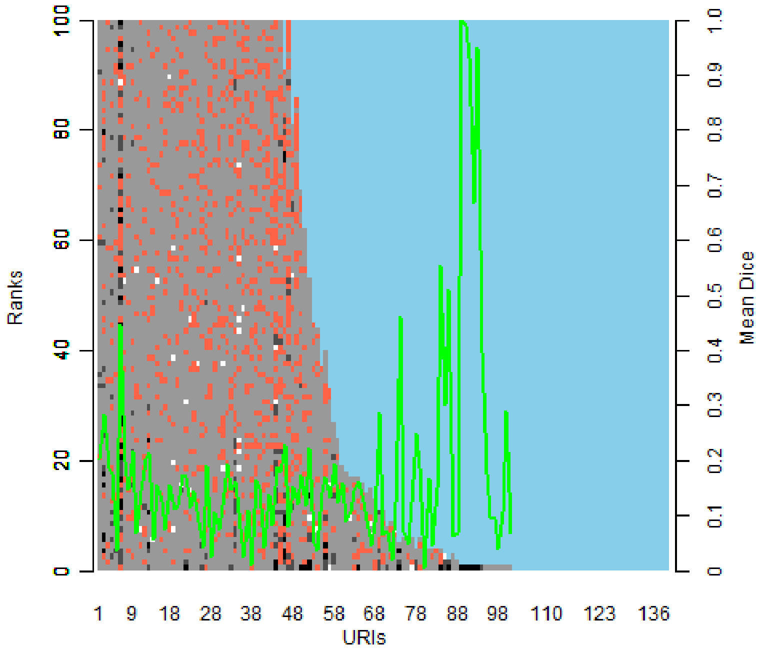


Fig. 49 Dice Similarity Coefficient for Top 100 7-Term Lexical Signature Results per Rank Including Mean Coefficient

3.2 Similarity of Search Result URIs

Another indicator for a successfully rediscovered missing web page is a high degree of similarity between the discovered URI and the URI of the missing page. The Jaro distance, introduced in Section 7.3 of Chapter II, is commonly used for the comparison of short strings such as names of individuals or companies. It is also suitable for the comparison of URIs. For example, consider the missing URI of the PSP conference in 2003 introduced in Table 1 of Chapter I
`http://www.pspcentral.org/events/annual_meeting_2003.html`
and its new URI
`http://www.pspcentral.org/events/archive/annual_meeting_2003.html`
shown in Table 2 of the same chapter. These URIs are very similar and their Jaro distance value of 0.91 confirms that. In this example the content is nearly unchanged which supports the point that a high Jaro distance indicates a high level of relevancy.

Following this intuition we compute the normalized Jaro distance for all top 100 results of all three methods. Similar in style with the graphs from the previous section Figures 51, 52 and 53 show the Jaro distances for the results distinguished by method. The light blue sections again show ranks of URIs without results. The Jaro distance values J are separated into four clusters which are represented as four different colors in the graph:

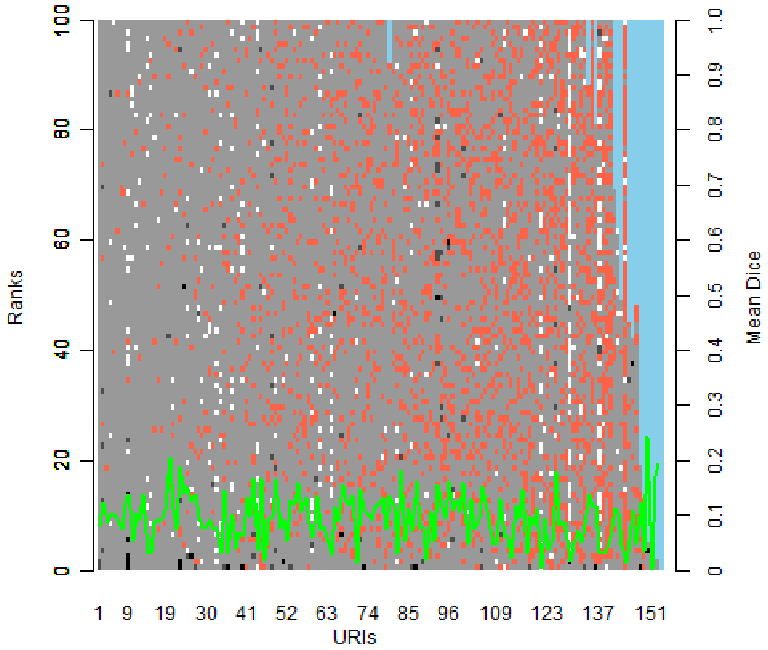


Fig. 50 Dice Similarity Coefficient for Top 100 Title Results per Rank Including Mean Coefficient

- $J = 0$ depicted in white
- $0 < J \leq 0.3$ in light gray
- $0.3 < J \leq 0.6$ in gray and
- $0.6 < J$ in black.

All three figures show a moderate Jaro distance with the average (green line) falling between 0.3 and 0.5 for most URIs. The majority of returned URIs are drawn with a gray dot meaning $0.3 < J \leq 0.6$ as J is part of the third cluster. Another interesting observation that is most obvious in Figure 53 is that very similar URIs are not necessarily returned in the lower ranks, e.g. the top 10. They are much more distributed over all ranks and seem to be just as likely to occur in the higher ranks as they occur in the lower ranks. This observation is somewhat expected since we do not anticipate search engines to discriminate by URIs but rather rank the results by relevancy and "importance" of a page (as determined, for example, by PageRank [76] and potentially many more criteria). The average Jaro distance per URI fluctuates the least in this graph due to the most amount of results returned. The line is dropping to the right due to the sorting of the URIs by decreasing mean Jaro distance value. However, we do not claim that URI similarity by itself can be considered reliable evidence for a successful rediscovery of a missing page. The Hypertext 2006 conference web page,

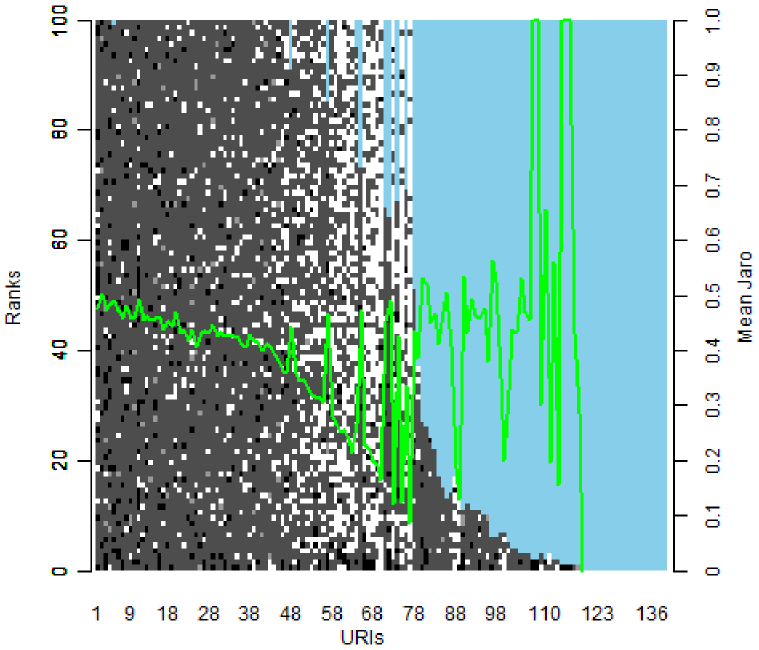


Fig. 51 Jaro Distance for Top 100 5-Term Lexical Signature Results per Rank Including Mean Distance

also shown in Chapter I, provides a negative example even though the URIs are identical. In general this assumption likely does not hold for URIs that have not changed at all or that have changed only slightly over time but their content has changed dramatically and is no longer relevant to its original as seen in the Hypertext example.

3.3 Relevancy of Search Results

The results of the previous section are satisfying only to a certain extent. They do not provide very clear indicators for strong similarity or dissimilarity of the returned results. The majority of the values were falling somewhere in the middle of the corresponding scale. For this reason we again utilized the Mechanical Turk for a human relevance evaluation.

In Section 2.1 we created HITs to determine the “aboutness” of all missing pages. Now we create HITs asking the users to judge the relevance between the “aboutness” and the returned results. For simplicity we restrict the process to the top 10 results returned from each method. This threshold is chosen somewhat arbitrarily and rather serves the sake of simplicity. We ask each HIT to be judged with one out of the following four scores:

- 0: URI is broken

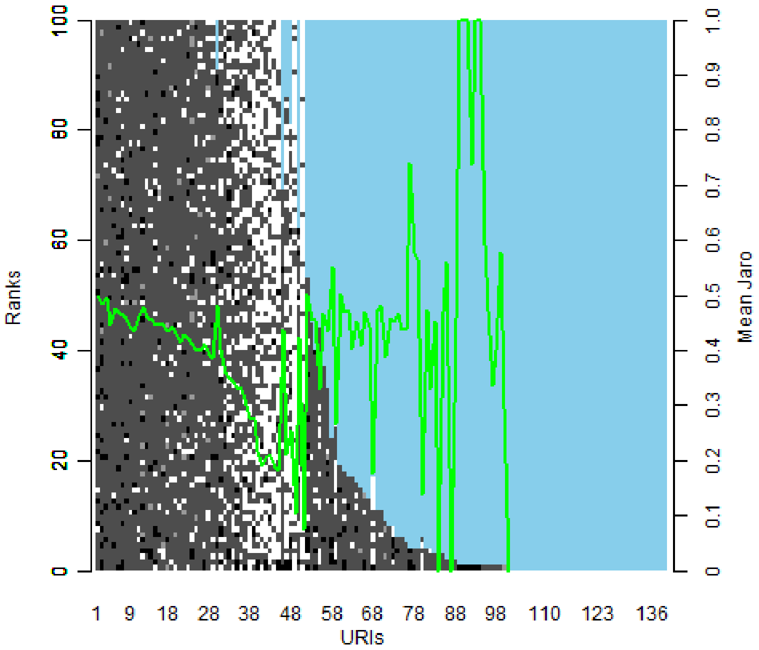


Fig. 52 Jaro Distance for Top 100 7-Term Lexical Signature Results per Rank Including Mean Distance

- 1: URI is not relevant
- 2: URI is somewhat relevant
- 3: URI is highly relevant.

Figure 54 shows the relevance scores for our applied three methods. All three subfigures display the relevance scores with ten groups of four bars each. Each group represents a rank in the top 10 from left to right. Each bar within a group represents a particular score. The height of the bars indicates the relative frequency of occurrence over all URIs referring to the left y-axis. The values of the solid line refer to the right y-axis and represent the average relevance score of the corresponding rank. The dominant relevance class over all methods and ranks is “somewhat relevant” with the score of 2. This becomes apparent with the corresponding bars peaking at about 60% and the average score for methods being very close to 2 over all ranks. The number of broken result URIs on the other hand is diminishing for all methods and ranks. Figure 54(a) representing the title relevance scores and Figure 54(b) for the 5-term lexical signatures shows a fairly balanced ratio between scores 1 and 3. Both bars fall in the 20% range for all ranks. The picture is similar for 7-term lexical signatures shown in Figure 54(c) even though we see some slight imbalance particular for ranks one, four and eight. However, the average relevance score here is also around 2 even though the line is not quite

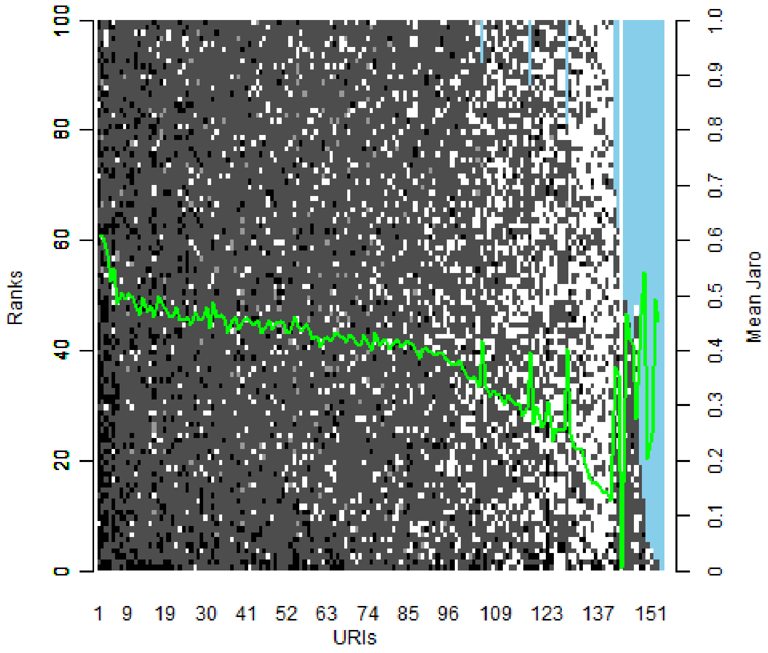


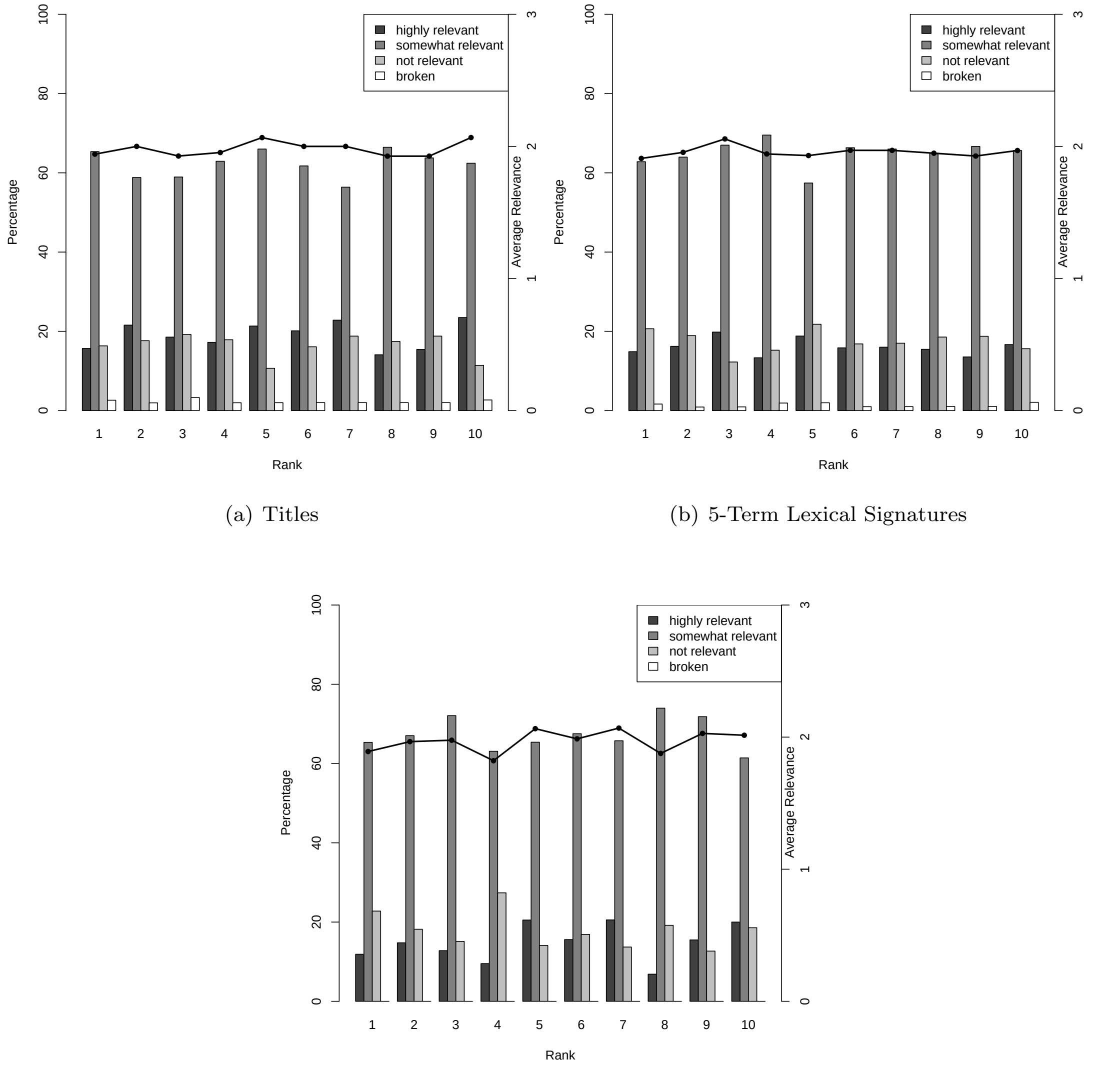
Fig. 53 Jaro Distance for Top 100 Title Results per Rank Including Mean Distance

as level as for the other two methods.

With a given relevance value for each of the top 10 results we also compute nDCG values for all methods. As mentioned earlier nDCG penalizes low relevancy in the higher ranks and high relevancy in the lower ranks. Since we created the HITs to judge relevance for the top 10 results only we can rewrite our nDCG values as $nDCG$@10.

Figure 55 displays four lines three of which represent our three applied methods. The green dotted line shows the nDCG values for the title results, the red dotted line for the 5-term lexical signature results and the blue dotted line for the 7-term lexical signature results. All three lines are sorted independently of each other in decreasing order of $nDCG$@10 values. The solid black line represents the union of all three lines meaning it plots the maximum nDCG value of any of the three methods for each URIs. Note that the y-axis starts at 0.5. The lines stretch to different values on the x-axis because they have different numbers of results returned for each method.

Obviously the $nDCG$@10 for all methods is very high. For all methods half of the URIs that have results returned show an $nDCG$@10 at or above 0.9. For the union of all methods this even holds true for two thirds of the URIs. However, we also see a steep drop at the tail of the line in particular for 7-term lexical signature results but also visible for the other two methods. That means we have only a small number of URIs with the relevancy of their results being well below the relevancy of the majority of results of other URIs. These kind of outliers do not surprise but we



(a) Titles

(b) 5-Term Lexical Signatures

(c) 7-Term Lexical Signatures

Fig. 54 Relevance of Results by Method

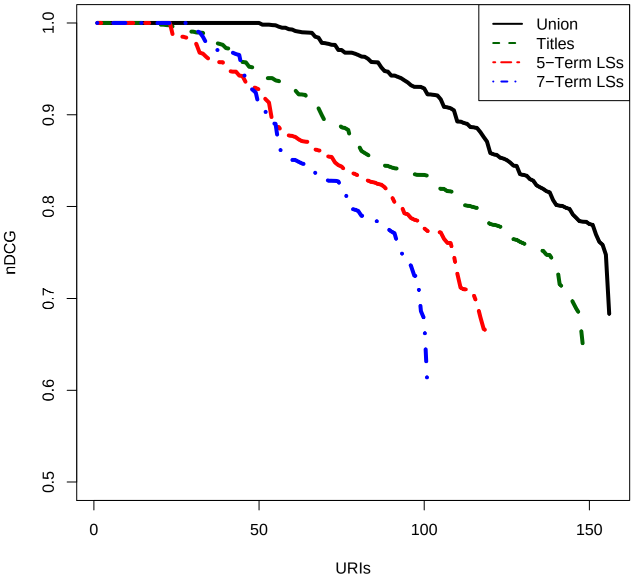


Fig. 55 nDCG per URI

rather expected to find them in our corpus. It is possible that their "aboutness" resulting from the HITs is not very specific or the URIs does not give away what the page could have been about and therefore it is difficult to judge a positive relevance for the results.

Figure 56 also shows the $nDCG$@10 values for the title results but has the values for the other two methods ordered accordingly. That means we see the same line for the title results as seen in Figure 55 and each point on that line has corresponding point for the 5- and 7-term lexical signature methods if the URI has results. In case of a tie it is broken up by the value for 5-term lexical signatures and by the remaining value for 7-term lexical signatures thirdly if a tie still exists.

This concept implies that some points from the title method do not have a corresponding point from the 5- and 7-term lexical signature methods and vice versa. The valuable observation of this graph is that even with a decreasing relevancy for the titles results we see high relevancy numbers for the other two methods. This is confirmed by the solid black line in Figure 55. It becomes particularly obvious for URIs 90 through 150 where we see only a few relevancy points below the titles line (mostly from the 5-term lexical signature method) but a large number of relevancy points above the solid line. From URI 120 on its seems primarily the 7-term lexical signature method that provides better results.

4 SUMMARY

In this chapter we introduce the *Book of the Dead*, a corpus of real missing web pages. The corpus is based on an aggregation of web pages created by the Library of Congress during a focused crawl

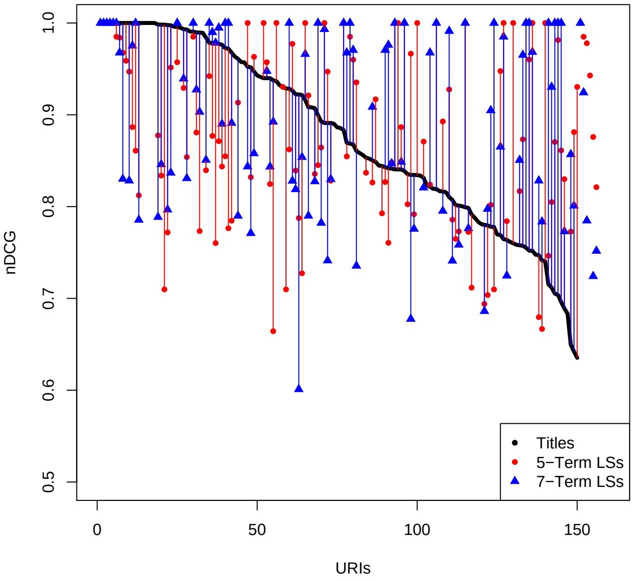


Fig. 56 nDCG per URI Ordered by Titles

around the topic of general elections in the US in 2004 and 2006 as well the terrorist attacks on the US on September 11 2001. To the best of our knowledge no corpus of missing web pages is available to researchers. We therefore conducted all experiments from Chapters IV through VIII on corpora containing randomly sampled web pages that are not really missing.

We apply two of our most promising methods for the rediscovery of missing web pages to the corpus and evaluate the results. We find that the title method returns results for most URIs while lexical signatures, especially 7-term lexical signatures, leave roughly 13% of URIs without results. Overall, most of the results show a fairly low content similarity to the Mementos of the missing pages the methods were applied to. The majority of Dice values fall between 0 and 0.3. However, we see a few results with high similarity scores which mainly occur in the top 10 results. The similarity between the URIs of the missing pages and their results are slightly higher. We used the Jaro distance and measured values between 0.3 and 0.6 for the most top 100 results.

We further utilize Amazon's micro-task service called "Mechanical Turk" to obtain a general and ideally unbiased idea of what each of the missing URIs were about. With this information we use the Mechanical Turk again to assess the relevancy of the top 10 results for all URIs and methods. The results are promising in the sense that the vast majority of results were judged as relevant and only about 20% were considered irrelevant regardless of the method applied. The ranking of results also shows encouraging numbers with nDCG values of 0.9 and higher for half of the title results, for example. Our results also prove that even if the relevancy of the results of one method is not great one or both of the other methods often provide highly relevant results. This means that the union of results is highly relevant with nDCG values of 0.9 or above for at least two out of three URIs.

Aside from these results, the "Book of the Dead" is a first approach to provide a comprehensive corpus including a description of the aboutness of the content that the URIs used to dereference to. We make this dataset available for download and hope it can provide a baseline for further relevant research. The Book of the Dead is available at the URI `http://bit.ly/Book-of-the-Dead`.

CHAPTER X

SYNCHRONICITY

1 BACKGROUND

In Chapters V through VIII we have introduced four methods to automatically rediscover missing web pages in real time. We have characterized parameters which when applied to these methods promise to show the best performance. For example, we have shown that a best performing lexical signature should consist of five or seven terms and that a title containing $\geq 75\%$ stop titles should be dismissed since its projected performance is poor.

In this chapter we introduce *Synchronicity*, a Mozilla Firefox add-on that supports the Internet user in (re-)discovering missing web pages in real time. Synchronicity is the prototype implementation of the above mentioned methods and further enables the user to browse previous versions of any given web page as long as they are available. The term *Synchronicity* was coined by the Swiss psychiatrist Carl Gustav Jung (1875-1961). It is a philosophical concept commonly defined as several events that appear to be causally unrelated but are observed to occur together in a meaningful manner [146]. We define our methods introduced in the previous chapters as such events and them working together constitutes the meaningful manner − the rediscovery of a missing web page.

The logo of Synchronicity is a shrimp. It is displayed in the top right corner of each panel when running the software. The idea is derived from the 1984 movie *Repo Man* [19] where in one scene a supporting character called Miller explains the philosophical concept of synchronicity (without actually using the term) to the main character Otto. Miller argues that there is a “lattice of coincidences” and gives the following example: Otto is supposed to think of a plate of shrimp and suddenly, out of the blue, someone else would mention “plate” or “shrimp” or “plate of shrimp” without any explanation. According to Miller there would be no point in looking for one since it all was part of an “cosmic unconsciousness”. This scene inspired the choice for a shrimp as the icon and logo for our Synchronicity add-on.

2 IMPLEMENTATION

Synchronicity is implemented as an add-on to the Mozilla Firefox web browser [34]. The user needs to download and install the extension and after restarting the browser the software is operational. Synchronicity places its icon in the status bar and whenever a 404 error occurs the icon will change color and appear with a bright red background. This indicates to the user that the error was caught and Synchronicity could be used to help out.

Synchronicity utilizes the *Memento* framework introduced in Chapter II, Section 4 for time based access of web resources. That means every time a URI results in a 404 error the system obtains the URI’s TimeMap provided by Memento. From the TimeMap, Synchronicity can grab and parse valid Mementos to obtain their titles and generate their lexical signatures. In its current state Synchronicity considers a Memento valid if the HTTP response code is not 404.

Synchronicity can also be activated without having observed a 404 error by simply double clicking on its icon. This is useful in case of a soft 404 error or other unexpected or unwanted content of a given URI as shown, for example, in Figure 2 in Chapter I.

3 DOCUMENT FREQUENCY SERVICE

We have shown in Chapter IV that estimating DF values for all terms candidate to make it into a lexical signature is a crucial aspect of the computation of accurate TF-IDF values. We have concluded from this chapter that screen scraping a search engine's web interface is a feasible and reliable approach to address this issue. Consequently Synchronicity relies on the screen scraping approach for its lexical signatures.

It appears to be a waste of resources to query a search engine for each term every single time this term occurs. A more reasonable approach is to implement a cache that stores previously queried terms and their obtained DF values. This way we mainly achieve two things. First, the load on the search engine is reduced since we avoid duplicate queries. Secondly, we expedite the TF-IDF computation since the DF service can respond faster to requests for DF values that are already stored in the cache and does not need to query the search engine.

The caching system is implemented as a Perl [35] based Common Gateway Interface (CGI) [227]. It is served through an Apache Web Server [31] running on a Linux [28] computer hosted by the Computer Science Department at Old Dominion University. The terms and their DF values are stored in an instance of a MySQL [25] database running on the same computer.

The CGI script is designed to take one term as an input parameter. It grabs the term from the URI and checks its availability in the database. If the term is found the script returns its DF value. If the term is not yet stored in the database the script queries it against the Yahoo! API and obtains the estimated DF value. It then stores the term and the DF value in the database before returning the DF value in response to the requested URI.

The web server is configured to rewrite an URI ending with

```
^/getdf/(.*)
```

to

```
/cgi-bin/getdf/getdf.cgi?term=$1
```

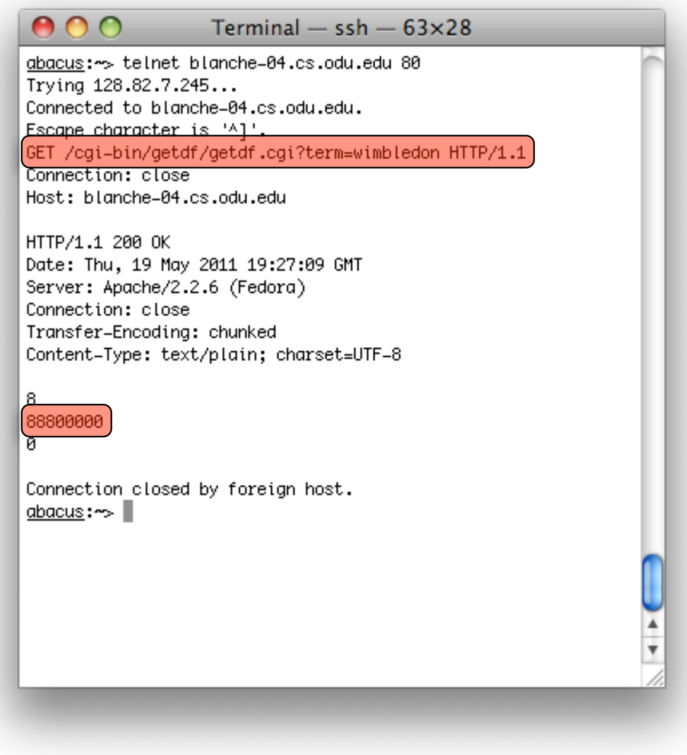
That means, for example, if someone would like to obtain the DF value of the term *Wimbledon* she would request the URI

```
http://blanche-04.cs.odu.edu/getdf/wimbledon
```

and the web server would rewrite it to

```
http://blanche-04.cs.odu.edu/cgi-bin/getdf/getdf.cgi?term=wimbledon
```

The DF value is the only data (besides the usual HTTP headers) returned which makes it easy for applications to parse it. Figure 57 shows the response of the web server to the request shown above. Since the HTTP protocol version 1.1 is used the response is chunked. The term *Wimbledon* returns a DF value of $88,800,000$.



```
abacus:~> telnet blanche-04.cs.odu.edu 80
Trying 128.82.7.245...
Connected to blanche-04.cs.odu.edu.
Escape character is '^]'.
GET /cgi-bin/getdf/getdf.cgi?term=wimbledon HTTP/1.1
Connection: close
Host: blanche-04.cs.odu.edu

HTTP/1.1 200 OK
Date: Thu, 19 May 2011 19:27:09 GMT
Server: Apache/2.2.6 (Fedora)
Connection: close
Transfer-Encoding: chunked
Content-Type: text/plain; charset=UTF-8

8
88800000
0

Connection closed by foreign host.
abacus:~>
```

Fig. 57 Telnet Session to Obtain Document Frequency Value

4 OPERATION

Synchronicity as seen in Klein et al. [156] triggers whenever a 404 error is encountered but it is up to the user to load the Synchronicity panel into the browser. The panel can also be triggered manually if desired. Synchronicity offers six options to support the Internet user in (re-)discovering missing web pages. Figure 58 shows Synchronicity's flow diagram with all options arranged. On startup the system displays two tabs named *Archived Version* and *New Version*. The user can switch between the tabs at any time. The purpose of the *Archived Version* tab is to display all available Mementos for the given URI. It provides two visualizations, a *TimeGraph* and a *TimeLine*. The TimeGraph is a dynamically generated but yet static image for the purpose of giving an overview about how many Mementos are available from what year. By chosing a date the user can display a particular Memento. The TimeLine mainly serves the purpose of browsing all Mementos. The TimeLine is "zoomable" into time and the user can and view what archive the Memento was derived from and when it was created. The user has the option to filter single archives and of course also display any particular Memento.

The *New Version* tab offers five methods to rediscover the missing page at its new URI or discover a replacement page that satisfies the user's information need. The first four of these methods are

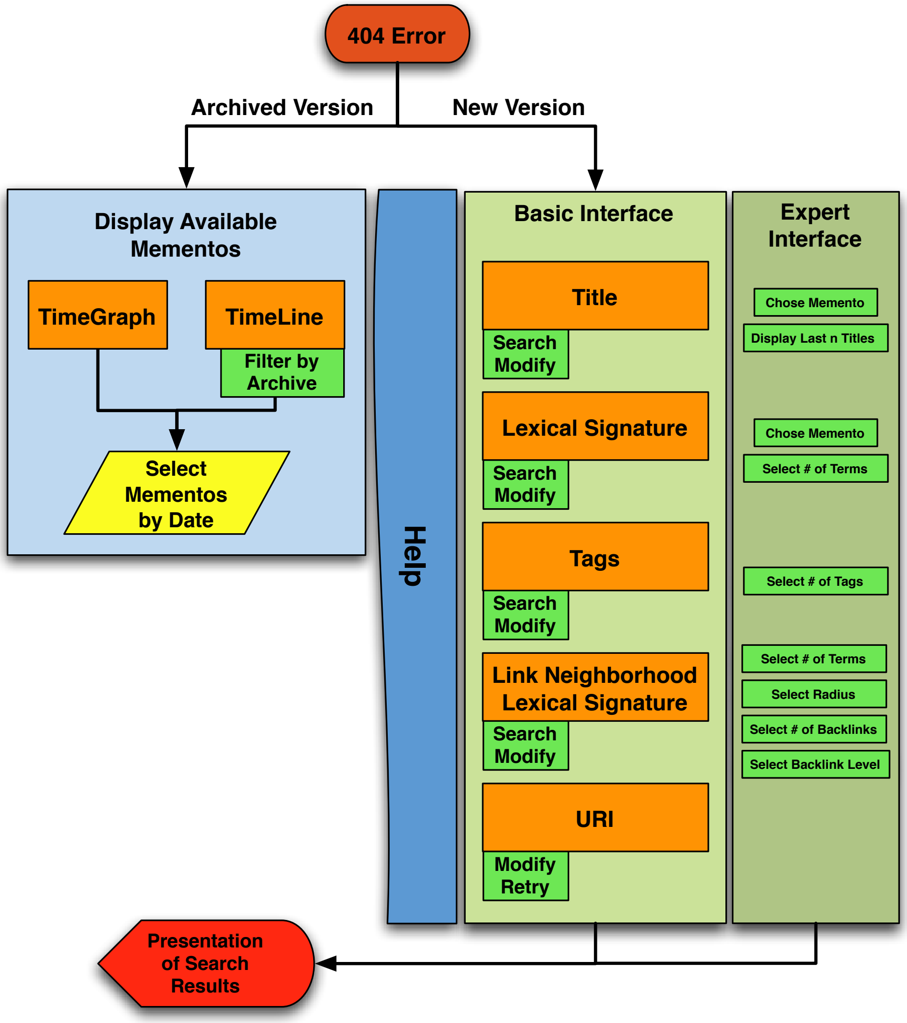
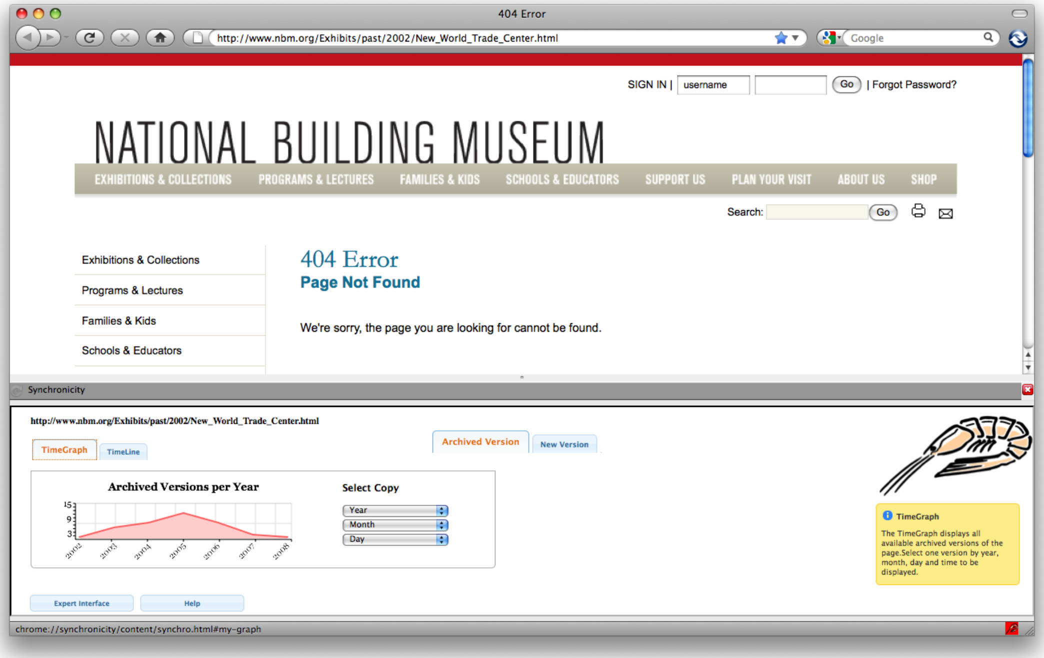


Fig. 58 Synchronicity Flow Diagram

based on generating search engine queries and represent the implementation of our results from Chapters V through VIII. The output of all methods can be modified by the user and queried against any of the three major search engines Google, Yahoo! and Bing. The fifth method is based on URI modifications meaning Synchronicity displays the URI that caused the 404 error and the user can edit it and retry. The search results are displayed in the main browser window.

Synchronicity displays a small yellow *InfoBox* at all times giving a brief explanation what the system can do at the corresponding panel. It, for example, explains how to zoom the TimeLine and what to do with the generated lexical signature. The add-on also offers a *Help* button which when clicked opens a new browser window with detailed supporting information for the user.

The user further is able to switch between two interfaces. The system by default offers the *Basic Interface* which includes all the functionality just described but hides more complex configuration



404 Error

http://www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html

SIGN IN | username Go | Forgot Password?

NATIONAL BUILDING MUSEUM

404 Error

Page Not Found

We're sorry, the page you are looking for cannot be found.

Synchronicity

http://www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html

TimeGraph

The TimeGraph displays all available archived versions of the page.Select one version by year, month, day and time to be displayed.

chrome://synchronicity/content/synchro.html#my-graph

Fig. 59 Synchronicity Displaying the TimeGraph for the Missing URI `www.nbm.org/Exhibits/past/2002/New\_World\_Trade\_Center.html`

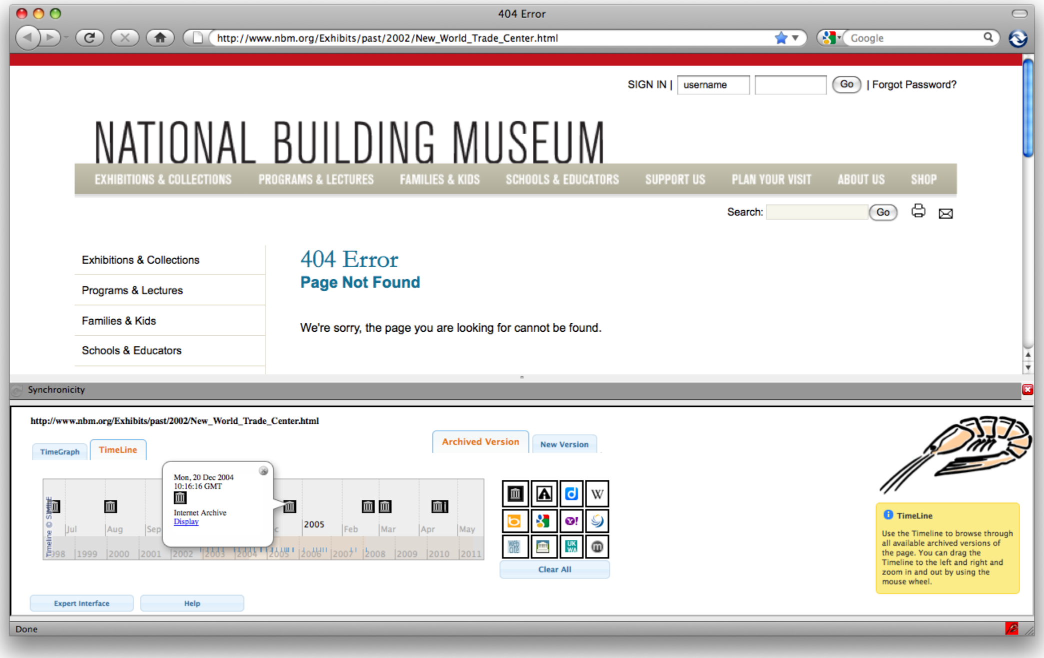
options from the user. The *Expert Interface* is described in detail in Section 5.

An important benefit of Synchronicity is that it works while the user is browsing and provides results in real time. Some methods may take longer than others but depending on the user's motivation she may be willing to invest more time to obtain the desired result if no other methods succeeded.

4.1 Mementos

The retrieved Memento TimeMap contains references to all available Mementos of the missing URI. Synchronicity visualizes all available Mementos in two different ways. First, it offers a *TimeGraph* which displays a static graph of the overall number of Mementos found per year. Figure 59 shows a screenshot of the TimeGraph for the missing URI `http://www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html`. The URI is part of the Book of the Dead. The TimeGraph gives the user an overview of when the majority of Mementos were created. Right next to the graph the add-on offers three drop-down boxes which dynamically fill with the dates and times of all Mementos. Here the user can pick one particular Memento by its creation time and have it displayed in the main browser window.

The second visualization is called *TimeLine*. It is located behind the accordingly labeled tab. All Mementos are displayed on a TimeLine implemented as a Simile Widget [29] as shown in Figure 60.



404 Error

http://www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html

SIGN IN | username Go | Forgot Password?

404 Error
Page Not Found

We're sorry, the page you are looking for cannot be found.

http://www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html

Mon, 20 Dec 2004
10:16:16 GMT
Internet Archive
Display

TimeLine

Use the Timeline to browse through all available archived versions of the page. You can drag the Timeline to the left and right and zoom in and out by using the mouse wheel.

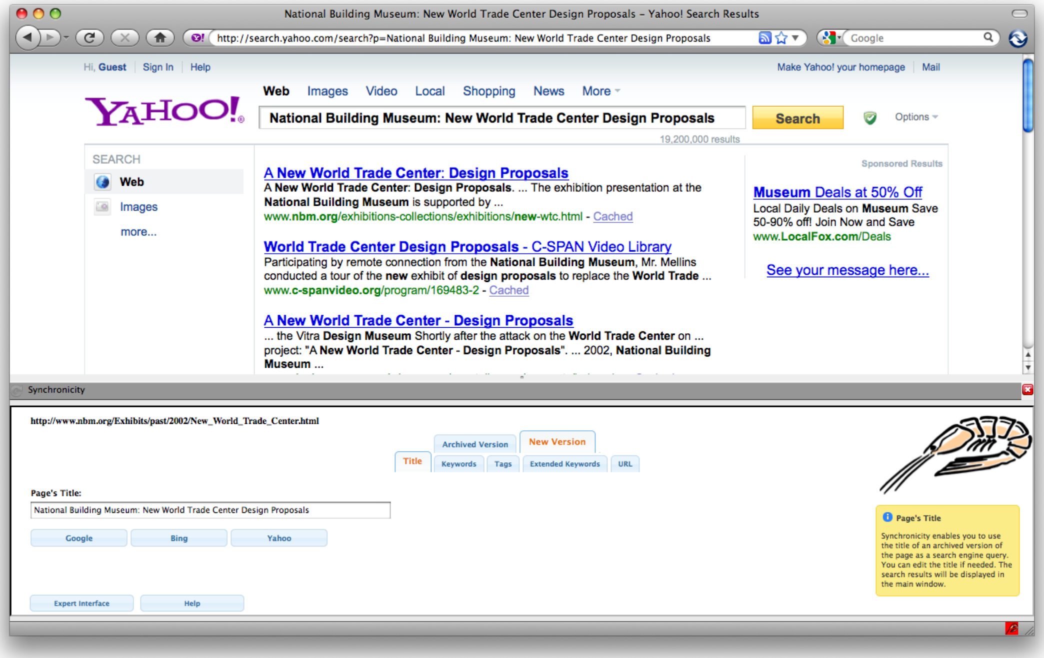
Done

Fig. 60 Synchronicity Displaying the TimeLine for the Missing URI www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html

Each Memento is placed at the proper time it was created and it is represented with the icon of the archive is was derived from. The TimeLine consists of two “bands” – a lower and an upper band. Each band can be dragged to the left and right which visualizes the “travel through time”. The lower band is displayed on a year granularity and also shows the colored section spanning from the first to the last available Memento. This section includes the time span currently displayed in the upper band. The upper band by default starts at the month granularity but it is, unlike the lower band, “zoomable”. That means a user can zoom into any given Memento and display the year, date and time it was created. As usual for Simile time lines this can be done with a click wheel mouse or multi-touch genstures on laptops. By clicking on one icon a small bubble appears that summarizes the meta information of this Memento. It further offers a link to display the archived version in the main browser window. The bubble is also visible in Figure 60.

The TimeLine further shows a matrix of all possible archives providing Mementos represented by their icons. By clicking on one icon the TimeLine filters out all its Mementos. By clicking it again its Mementos re-appear. Located just below the matrix is a button to clear all Mementos from the TimeLine. This is useful if the user wants to, for example, only display Mementos from the National Archives but does not want to filter all other archives by clicking on every single icon. After clicking the *Clear All* button its label changes to *Show All* which reverses the previous action.

If the user's information need is satisfied with displaying a Memento either with the help of the TimeGraph or the TimeLine nothing else needs to be done. Note that in case no Mementos are



National Building Museum: New World Trade Center Design Proposals – Yahoo! Search Results
http://search.yahoo.com/search?p=National Building Museum: New World Trade Center Design Proposals
Google
Hi, Guest | Sign In | Help
Make Yahoo! your homepage | Mail
Web Images Video Local Shopping News More
YAHOO!
Search
Options
SEARCH
Web
Images
more...
Sponsored Results
Synchronicity
Archived Version
New Version
Title
Keywords
Tags
Extended Keywords
URL
Google
Bing
Yahoo
Expert Interface
Help

National Building Museum: New World Trade Center Design Proposals

19,200,000 results

A New World Trade Center: Design Proposals
A **New World Trade Center**: **Design Proposals**. ... The exhibition presentation at the **National Building Museum** is supported by ...
www.**nbm.org**/exhibitions-collections/exhibitions/**new**-wtc.html - Cached

World Trade Center Design Proposals - C-SPAN Video Library
Participating by remote connection from the **National Building Museum**, Mr. Mellins conducted a tour of the **new** exhibit of **design proposals** to replace the **World Trade** ...
www.**c-spanvideo.org**/program/169483-2 - Cached

A New World Trade Center - Design Proposals
... the Vitra **Design Museum** Shortly after the attack on the **World Trade Center** on ... project: "A **New World Trade Center - Design Proposals**". ... 2002, **National Building Museum** ...

Museum Deals at 50% Off
Local Daily Deals on **Museum** Save 50-90% off! Join Now and Save
www.**LocalFox.com**/Deals

See your message here...

http://www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html

Page's Title:
National Building Museum: New World Trade Center Design Proposals

Page's Title
Synchronicity enables you to use the title of an archived version of the page as a search engine query. You can edit the title if needed. The search results will be displayed in the main window.

Fig. 61 Synchronicity Displaying the Search Results in Yahoo! with the Title of an Obtained Memento

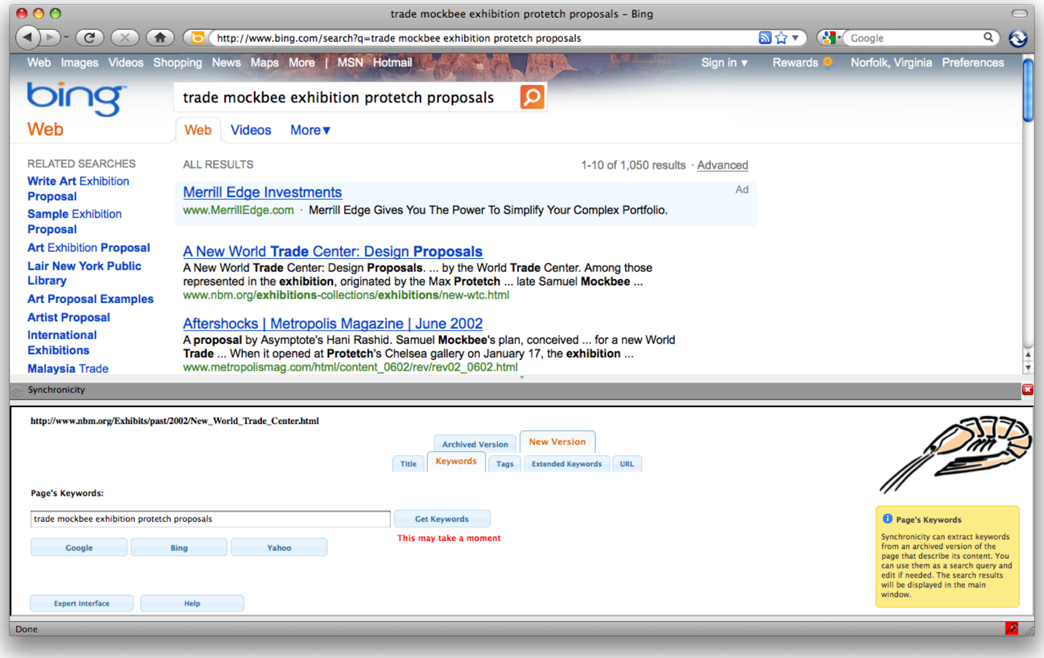
available for any given URI, Synchronicity displays an appropriate message.

4.2 Search with Title

If an archived version of the missing page is not sufficient for the user she can click on the *New Version* button which leads to a new panel offering five different methods to obtain a satisfying version of the missing page.

The primary method, as argued in Chapter VI, is the Memento's title as a textual feature describing the aboutness of the missing page. By default Synchronicity obtains the title of the latest available Memento. Obtaining this title requires just one request and therefore it is done automatically when the add-on loads. That means there is no delay in obtaining the title and it is shown automatically when the user clicks on the *New Version* button. The title can be used as a search engine query in Google, Yahoo! and Bing and the search results are displayed in the main browser window. Figure 61 shows the screen capture of Synchronicity having issued the title of our missing example URI into Yahoo!. The title obtained from the latest available Memento is *National Building Museum: New World Trade Center Design Proposals*. The top ranked resturned search result is indeed the new location of the missing page `www.nbm.org/exhibitions-collections/exhibitions/new-wtc.html`.

At any point the user can edit the obtained title and add or remove terms and re-query. In



http://www.bing.com/search?q=trade mockbee exhibition protetch proposals

trade mockbee exhibition protetch proposals

Write Art Exhibition Proposal
Sample Exhibition Proposal
Art Exhibition Proposal
Lair New York Public Library
Art Proposal Examples
Artist Proposal
International Exhibitions
Malaysia Trade

1-10 of 1,050 results

Merrill Edge Investments
www.MerrillEdge.com · Merrill Edge Gives You The Power To Simplify Your Complex Portfolio.

A New World **Trade** Center: Design **Proposals**
A New World **Trade** Center: Design **Proposals**. ... by the World **Trade** Center. Among those represented in the **exhibition**, originated by the Max **Protetch** ... late Samuel **Mockbee** ...
www.nbm.org/**exhibitions**-collections/**exhibitions**/new-wtc.html

Aftershocks | Metropolis Magazine | June 2002
A **proposal** by Asymptote's Hani Rashid. Samuel **Mockbee**'s plan, conceived ... for a new World **Trade** ... When it opened at **Protetch**'s Chelsea gallery on January 17, the **exhibition** ...
www.metropolismag.com/html/content_0602/rev/rev02_0602.html

http://www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html

Page's Keywords:

trade mockbee exhibition protetch proposals

This may take a moment

Page's Keywords

Synchronicity can extract keywords from an archived version of the page that describe its content. You can use them as a search query and edit if needed. The search results will be displayed in the main window.

Fig. 62 Synchronicity Displaying the Search Results in Bing with the Lexical Signature of an Obtained Memento

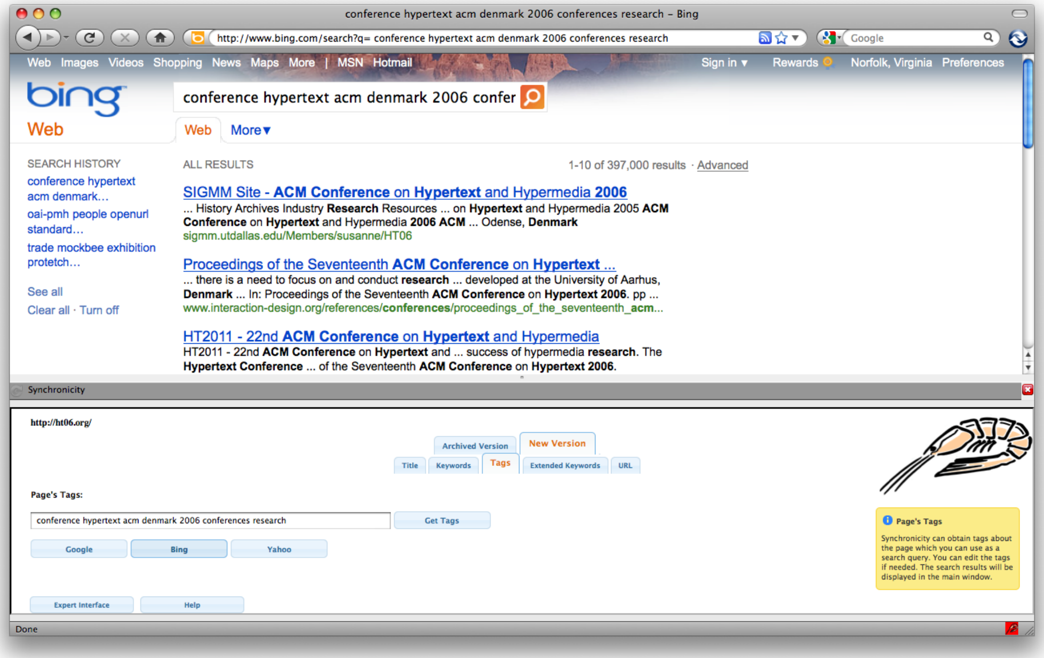
compliance with our findings in Chapter VI, Synchronicity informs the user if the obtained title is predicted to perfrom poorly and recommends switching to the next option right away.

4.3 Search with Keywords

As a secondary method Synchronicity offers the generation of a lexical signature from the textual content of the retrieved Mementos. We chose the label "Keywords" as it is, unlike "lexical signatures", a commonly well understood term but basically refers to the same concept. Here again the system uses the last available Memento. By default and in agreement with our results from Chapter V, Synchronicity generates 5-term lexical signatures. Depending on the length of the document and bandwidth of the network connection this process may take some time. A proper warning is displayed right below the button that the user needs to click on in order to generate a lexical signature. The system indicates that it is "busy" by displaying a rotating shrimp similar to the well known hour glass cursor in the Microsoft Windows environment or the spinning "rainbow wheel" known from Macintosh systems.

Figure 62 provides a snapshot of Synchronicity displaying the search results from Bing when queried with the lexical signature of the missing example URI. The generated lexical signature is *trade mockbee exhibition protetch proposals* and just like the title it returnes the new URI top ranked.

Similar to the previous method the user is encouraged to use the lexical signature as a search


conference hypertext acm denmark 2006 conferences research – Bing
http://www.bing.com/search?q= conference hypertext acm denmark 2006 conferences research
Google
Web Images Videos Shopping News Maps More | MSN Hotmail
Sign in ▾
Rewards
Norfolk, Virginia
Preferences
bing
Web
Web
More ▾
Synchronicity
Archived Version
New Version
Title
Keywords
Tags
Extended Keywords
URL
Get Tags
Google
Bing
Yahoo
Expert Interface
Help
Done

conference hypertext acm denmark 2006 confer

SEARCH HISTORY

conference hypertext acm denmark…

oai-pmh people openurl standard…

trade mockbee exhibition protetch…

See all

Clear all · Turn off

ALL RESULTS

1-10 of 397,000 results · Advanced

SIGMM Site - **ACM Conference** on **Hypertext** and Hypermedia **2006**

… History Archives Industry **Research** Resources … on **Hypertext** and Hypermedia 2005 **ACM Conference** on **Hypertext** and Hypermedia **2006 ACM** … Odense, **Denmark**

sigmm.utdallas.edu/Members/susanne/HT06

Proceedings of the Seventeenth **ACM Conference** on **Hypertext** …

… there is a need to focus on and conduct **research** … developed at the University of Aarhus, **Denmark** … In: Proceedings of the Seventeenth **ACM Conference** on **Hypertext 2006**. pp …

www.interaction-design.org/references/**conferences**/proceedings_of_the_seventeenth_**acm**…

HT2011 - 22nd **ACM Conference** on **Hypertext** and Hypermedia

HT2011 - 22nd **ACM Conference** on **Hypertext** and … success of hypermedia **research**. The **Hypertext Conference** … of the Seventeenth **ACM Conference** on **Hypertext 2006**.

http://ht06.org/

Page's Tags:

conference hypertext acm denmark 2006 conferences research

Page's Tags

Synchronicity can obtain tags about the page which you can use as a search query. You can edit the tags if needed. The search results will be displayed in the main window.

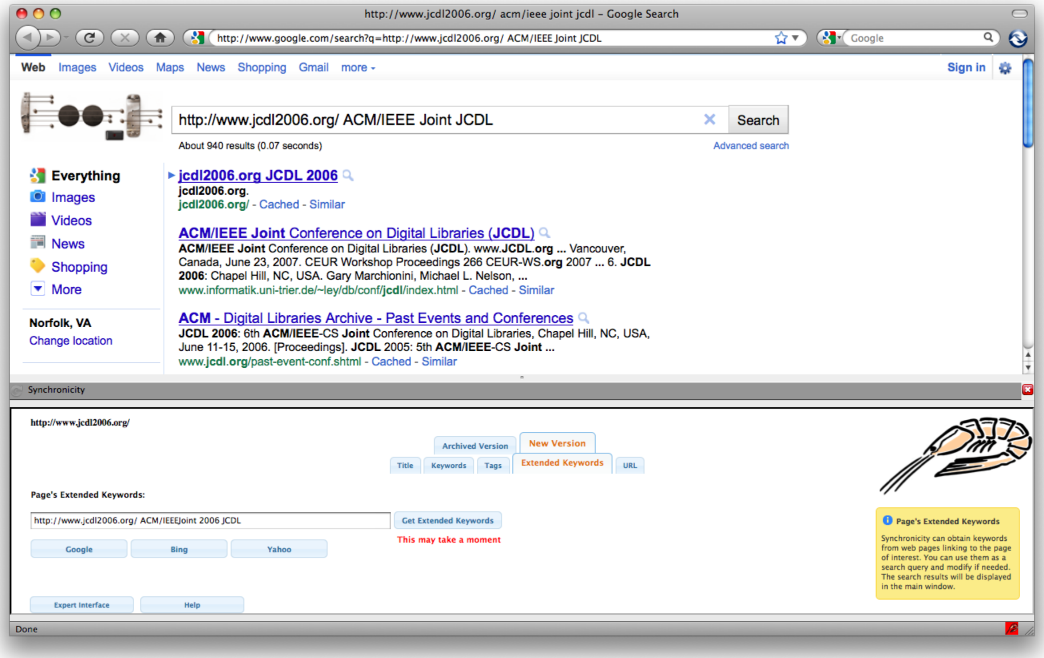
Fig. 63 Synchronicity Displaying the Search Results in Bing with the Obtained Tags from Delicious

engine query string and she can modify the lexical signature as desired. The search results will again be displayed for the user to explore.

4.4 Search with Tags

The next tab in the *New Version* panel offers the use of tags. Unlike titles and lexical signatures, tags can also be applied for cases where no Mementos are available. Synchronicity queries the social bookmarking site *Delicious* to obtain tags that users have used to annotate the now missing page. The system applies our results from Chapter VII by offering seven tags by default. Here again can the user edit the query string at any time. In case no tags can be obtained Synchronicity will give proper feedback to the user.

As mentioned in Chapter VII the sparcity of tags prevents them from being a primary method to rediscover missing web pages. There are no tags available for our sample URI. However, tags are available for the Hypertext web page introduced in Chapter I. Delicious users have created the tags *conference hypertext acm denmark 2006 conferences research* to annotate the URI `ht06.org`. Synchronicity and the obtained tags queried against Bing can be seen in Figure 63. This additional example is particularly interesting since someone has "parked" the domain some time ago and therefore the latest available Memento dereferences to the current unrelated content. Having Synchronicity extract the title or generate the lexical signature may in this case result in "incorrect"



http://www.google.com/search?q=http://www.jcdl2006.org/ ACM/IEEE Joint JCDL

http://www.jcdl2006.org/ ACM/IEEE Joint JCDL

About 940 results (0.07 seconds)

Norfolk, VA

jcdl2006.org JCDL 2006
jcdl2006.org.
jcdl2006.org/ - Cached - Similar

ACM/IEEE Joint Conference on Digital Libraries (JCDL)
ACM/IEEE Joint Conference on Digital Libraries (JCDL). www.JCDL.org ... Vancouver, Canada, June 23, 2007. CEUR Workshop Proceedings 266 CEUR-WS.org 2007 ... 6. JCDL 2006: Chapel Hill, NC, USA. Gary Marchionini, Michael L. Nelson, ...
www.informatik.uni-trier.de/~ley/db/conf/jcdl/index.html - Cached - Similar

ACM - Digital Libraries Archive - Past Events and Conferences
JCDL 2006: 6th ACM/IEEE-CS Joint Conference on Digital Libraries, Chapel Hill, NC, USA, June 11-15, 2006. [Proceedings]. JCDL 2005: 5th ACM/IEEE-CS Joint ...
www.jcdl.org/past-event-conf.shtml - Cached - Similar

http://www.jcdl2006.org/

Page's Extended Keywords:

http://www.jcdl2006.org/ ACM/IEEEJoint 2006 JCDL

This may take a moment

Page's Extended Keywords

Synchronicity can obtain keywords from web pages linking to the page of interest. You can use them as a search query and modify if needed. The search results will be displayed in the main window.

Fig. 64 Synchronicity Displaying the Search Results in Google with the Link Neighborhood Lexical Signature of a Page

input for search engine queries since by default the add-on would just parse the latest Memento. However, the available tags can be used for search. They also represent an example for ghost tags as introduced in Chapter VII since they better describe a previous version of the page than the current one. As long as users do not go back and delete their tags, they can be especially useful for such cases.

Even though the results shown in Figure 63 do not contain the replacement page suggested in Table 2 of Chapter I they do contain relevant content about the Hypertext conference.

4.5 Search with Extended Keywords

The most complex option Synchronicity offers is using the content of pages linking to the missing page. Link neighborhood lexical signatures represent another method applicable for cases where no Mementos of the missing URI can be found. Synchronicity queries search engines to obtain backlinks (pages linking to the missing page) and generates a lexical signature from this link neighborhood. With "Extended Keywords" we again chose a more commonly used label for the tab. The default parameters are adjusted to our findings from Chapter VIII meaning the system uses the top 10 backlinks from the first level only and it considers only the anchor text to generate a 4-term link neighborhood based lexical signature. This signature also serves as a search engine query and can be modified by the user.

Figure 64 shows the Google search results for the link neighborhood lexical signature of the web page for the JCDL conference 2006. Its URI `www.jcdl2006.org/` does not result in a 404 error per se but it displays nothing but an empty page with the URI on it. The conference content is no longer available and hence the page can be considered missing. In fact this URI is an example of a soft 404 as introduced in Chapter III, Section 2.4. Its link neighborhood lexical signature as generated by Synchronicity is *http://www.jcdl2006.org/ ACM/IEEEJoint 2006 JCDL*. It results as seen in Figure 64 in relevant results with the probably most accurate replacement page being `www.jcdl.org/archived-conf-sites/jcdl2006/` ranked fifth in the result set.

4.6 URI Modification

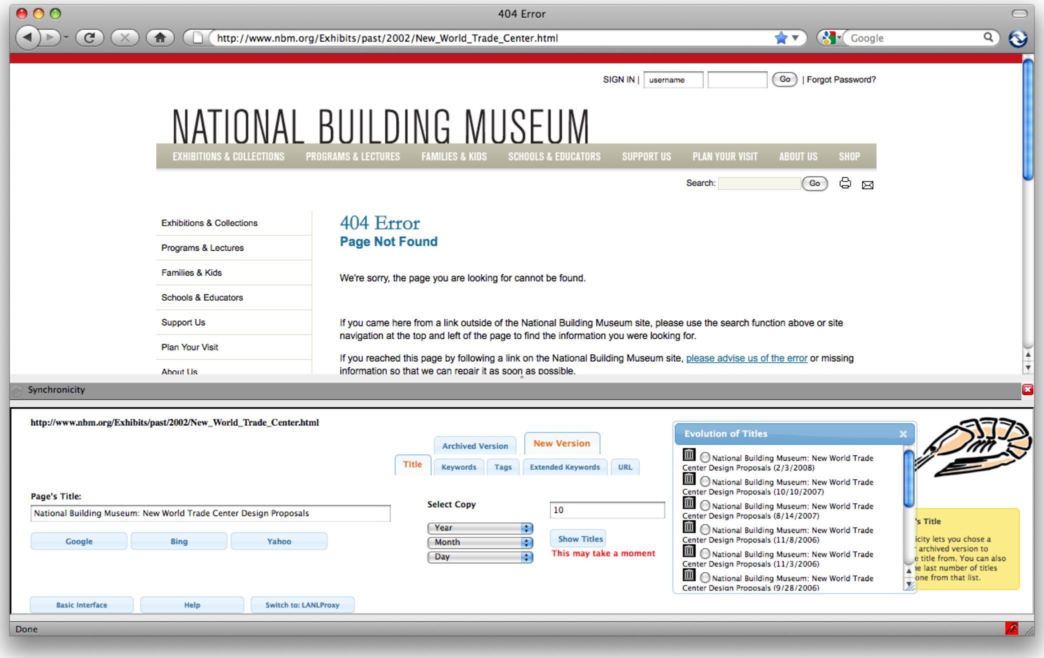
The fifth method that Synchronicity offers to the user is to modify and retry the actual URI that caused the 404 response. The URI is automatically loaded into the proper text field where it can be edited. The intuition is that shortening a long URI may help to at least find a new starting point to browse for the desired resource. The root of the URI, for example, may still be responsive and the user can possibly navigate from there to the desired page. This concept is not new, it, for example, was implemented in the customized 404 page seen in Figure 8 in Chapter III.

5 ADVANCED OPERATION

Synchronicity by default starts with what we call the *Basic Interface*. However, the system offers an additional interface aimed at the more advanced user. It is called *Expert Interface* and can be activated by clicking on the corresponding button. The button can be used to toggle between the two interfaces. Its label changes depending on what interface is currently activated. The user can switch between interfaces at any point during the interaction with the add-on will arrive at the corresponding tabs of the proper interface.

The functionality for both the TimeLine and the TimeGraph remains the same for the expert user. However, the user is now able to switch to another Memento proxy. There are currently two proxies available, one at Old Dominion University and one at the Los Alamos National Laboratory and the user can chose between the two. This may have an effect on the amount of Mementos available and their sources.

The *New Version* tab offers additional functions to the expert user. The main idea is that all parameters that have been investigated in Chapters V through VIII can be manually changed here. For the Basic Interface we implemented default values that have been shown to perform best for all methods. However, the expert user has now the chance to modify these parameters. For the title and keyword tab the expert user can chose one particular Memento to grab the title and generate the lexical signature from. Recall that the Basic Interface automatically choses the latest available Memento. The title tab further offers a function to display the titles of the last n number of Mementos. The value of n by default is set to ten but it can be modified. The window displaying the titles gives the user an impression of the evolution of titles over time and enables her to pick one particular title for the search. In addition the icons of the corresponding archives are displayed next to the titles. Figure 65 shows the additional options for the title based web page rediscovery. The three drop down boxes are available to pick one particular Memento to obtain the title from.



http://www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html

404 Error

Page Not Found

We're sorry, the page you are looking for cannot be found.

If you came here from a link outside of the National Building Museum site, please use the search function above or site navigation at the top and left of the page to find the information you were looking for.

If you reached this page by following a link on the National Building Museum site, please advise us of the error or missing information so that we can repair it as soon as possible.

http://www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html

Page's Title: National Building Museum: New World Trade Center Design Proposals

Select Copy: Year, Month, Day; 10

This may take a moment

Evolution of Titles

- ○ National Building Museum: New World Trade Center Design Proposals (2/3/2008)
- ○ National Building Museum: New World Trade Center Design Proposals (10/10/2007)
- ○ National Building Museum: New World Trade Center Design Proposals (8/14/2007)
- ○ National Building Museum: New World Trade Center Design Proposals (11/8/2006)
- ○ National Building Museum: New World Trade Center Design Proposals (11/3/2006)
- ○ National Building Museum: New World Trade Center Design Proposals (9/28/2006)

's Title

icity lets you chose a archived version to e title from. You can also e last number of titles one from that list.

Fig. 65 Synchronicity Displaying its Expert Interface and the Options for a Title Based Rediscovery of a Page

This feature can be helpful if the content of Mementos vary drastically and one particular Memento promises to provide a better title than others. The small window to the right lists the titles of the last ten Mementos and the icons of their archives for our initial sample URI. This option is useful to follow the evolution of titles over time.

The number of terms forming a lexical signature and number of tags for search was by default set to five and seven respectively. The expert user can change these values as desired.

The tab for search with extended keywords offers the most options to the expert user. All four parameters that were evaluated in Chapter VIII can be changed here:

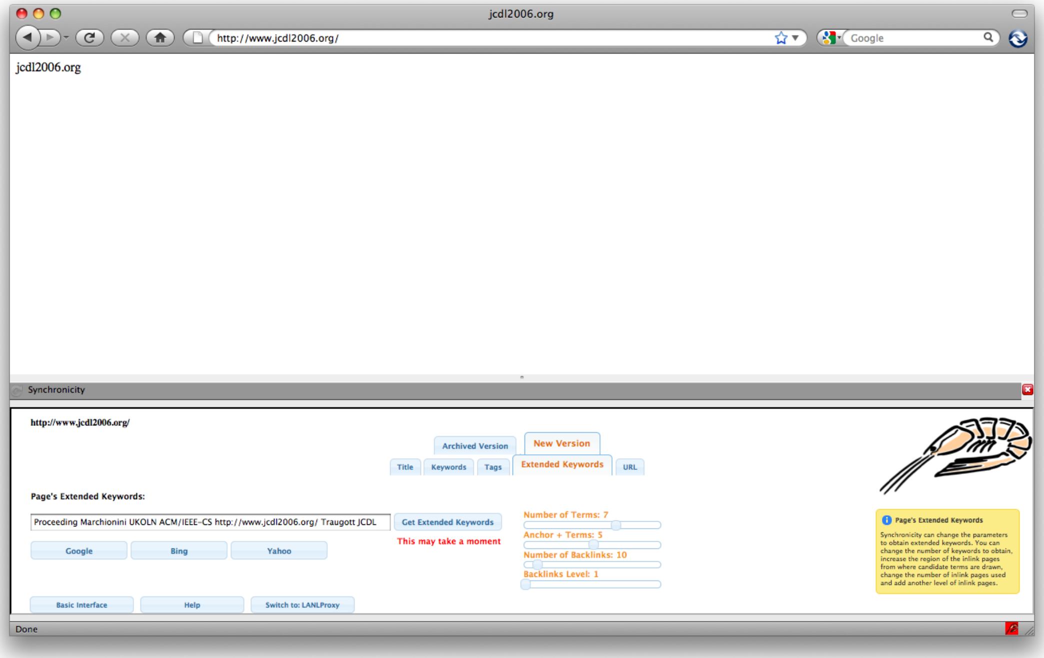
1. the number of terms for the link neighborhood based lexical signature (default is set to four)

2. the radius (anchor text $\pm n$ terms where n by default is zero)

3. the number of backlinks (default is set to ten) and

4. the backlink level (default is set to one).

By increasing any of these values the computation of the neighborhood based lexical signature will take more time. This is most significantly noticeable when including the second level backlink. However, the argument holds true here too: if the user is in real need of the page or seeks to gain knowledge what the page was about, she might be willing to invest that extra time.



http://www.jcdl2006.org/

Page's Extended Keywords: Proceeding Marchionini UKOLN ACM/IEEE-CS http://www.jcdl2006.org/ Traugott JCDL

This may take a moment

Number of Terms: 7

Anchor + Terms: 5

Number of Backlinks: 10

Backlinks Level: 1

Page's Extended Keywords

Synchronicity can change the parameters to obtain extended keywords. You can change the number of keywords to obtain, increase the region of the inlink pages from where candidate terms are drawn, change the number of inlink pages used and add another level of inlink pages.

Fig. 66 Synchronicity Displaying its Expert Interface and the Parameters to Generate the Link Neighborhood Lexical Signature of a Soft 404 Page

Figure 66 shows the screen capture of the Expert Interface providing sliders to adjust the paramters for the link neighborhood lexical signature generation. In this example Synchronicity created *Proceeding Marchionini UKOLN ACM/IEEE-CS http://www.jcdl2006.org/ Traugott JCDL* as a 7-term signature including ±5 terms around the anchor text. Figure 66 also shows the JCDL 2006 page considered missing in the previous example of Figure 64.

All options for all methods available through the Expert Interface are summarized in Figure 58.

6 SUMMARY

In this chapter we introduce *Synchronicity*, a Mozilla Firefox add-on that supports the user in rediscovering missing web pages in real time. Synchronicity utilizes Memento to obtain previous versions of a web page. It enables a user to browse through all available Mementos aggregated from various web archives and search engine caches. She can evaluate Mementos by the time they were archived and filter the displayed copies by particular archives.

Accessing an archived version of a page may already be sufficient to fulfill a user's information need. However, Synchronicity offers a number of addidtional methods to discover the missing page at its new location or good enough replacement pages. The obtained Mementos build the foundation for applying the title and lexical signature based rediscovery methods introduced in Chapters V and VI. The software further implements the tag based and link neighborhood lexical signature based methods introduced in Chapters VII and VIII. These methods are applicable if no Mementos of an URI are available. All four methods result in a query string that can be issued against search engines and the search results are displayed for the user to browse. While creating this string Synchronicity by default implements parameters that have been shown to perform best for all methods in the previous chapters. The user can modify the generated query string at any time and re-query against one of the three major search engines (Google, Yahoo! and Bing).

Synchronicity has an easy to use and straight forward interface but also gives an more exerienced user the option to modify all default parameters for the generation of the search queries. Not only can she, for example, modify the number of terms in a tag based query but also pick a particular Memento to obtain a title or generate a lexical signature from. The title evolution can be made visible by chosing to display the titles of a specified number of last Mementos.

Synchronicity is open source software and it is available for download via the Mozilla website [30]. Synchronicity is also available at the more marketable URI `http://bit.ly/no-more-404`.

As a Firefox add-on it is easy to install on all platforms and can support the user in overcoming the browsing detriment of 404 “Page not Found” errors.

CHAPTER XI

FUTURE WORK AND CONCLUSIONS

1 ASPECTS FOR FUTURE WORK

We see several aspects where the work presented here can be expanded in the future.

1.1 Interplay With Memento

This work focuses on providing alternative inputs for the *c2u* function established in Chapter I. The input is generated by any of the introduced implementations of the *rr* function and it typically returns a set or URIs. The *rr* function captures the "aboutness" of a document in a compact form. That means at the beginning of this process we distill the contextual essence of a document and at the end we obtain a set of URIs valid at time t_{now}.

The Memento framework on the other hand starts with a URI and a time t and returns (for example) a URI that identifies a previous version of a resource. That means Memento is helpful in studying the evolution of a concept at URI_1 over time. But if the concept moves from URI_1 to URI_2 Memento can not follow, but rather reports a "false dead end" of the concept even though it lives on and evolves at URI_2.

The challenge in the future is to bring these two orthogonal processes together and research how we can find the same or similar concepts at different URIs in web archives. That means given a concept it should return URI_1 @ t_1, URI_2 @ $t_2, ..., URI_n$ @ t_n. Such a system will enable users to map concepts into archives, a process that currently is not possible.

For example, if a user wants to tell a story about the Deepwater Horizon oil spill in the Gulf of Mexico she has to provide Memento with a URI (and a time) in order to obtain previous versions of that URI. She might be aware of `cnn.com`, for example, which would certainly be a good starting point but she is not able to retrieve archived information from sources (URIs) she is not aware of. A possible "concept-aware" Memento would be able to support the user in exploring a topic across URIs and across different archives hence enabling her to tell a rich story.

Related to this aspect, Memento could benefit from a contextual analysis of the previous version considered to be returned. For example, if a user uses Memento to travel back in time for the URI `http://ht06.org` she might be surprised to see that previous versions of that resource are contextually irrelevant to the current context. Note that in this example the previous copies are considered to be the original content of the Hypertext conference in 2006 and the current version provides unrelated content as seen in Chapter I.

Ultimately we can think of an enhanced Memento service giving the user a preview of the content to expect or maybe even an automatic dismissal of irrelevant versions of resoures. That would improve time travel in the web since it could prevent the user from unwanted surprises.

1.2 Study of Ghost Tags

We introduced the notion of "ghost tags" in Chapter VII as tags that better describe a previous version of a page than the current one. The method we used to explore these tags can be extended in several dimensions. We discovered ghost tags by analyzing the first Memento meaning the oldest available archived version of a page only. There is obviously potential to investigate more Mementos which would most likely result in the discovery of a higher precentage of ghost tags. It also would be interesting to observe when terms "ghostify" meaning at what point they are deleted from the document. This could be put into perspective with the term's importance within the document and within Delicious at that time.

We only used the bookmarking service Delicious to obtain tags about URIs. Even though Delicious is arguably the most popular of these services we could utilize other annotation sites such as Connotea [6], Historious [5] or Google Bookmarks [13].

1.3 Synchronicity Improvements

Synchronicity, even though more than a prototype, has not been developed beyond the beta status yet. Its functionality can be enhanced in several directions.

The TimeGraph already provides some metadata about the discovered Mementos and the archives they are available from. It would benefit from additionally displaying the amount of Mementos discovered both in total numbers and distinguished by archive.

The lexical signatures are being generated based on the DF service introduced in Chapter X. This implies the user has no choice but to use our DF values based on the Yahoo! index. It would be good to offer alternatives and let the advanced user chose the index. The user could additionally benefit from being able to chose between lexical signatures generated by different TF and DF weighted functions as evaluated in Park et al. [214].

The currently latest version of Mozilla Firefox includes a lightweight database system called IndexedDB. A great advancement to Synchronicity will be to have the user build their own local DF index. For example, the extension could store each [DF value, term] pair it uses and therefore over time build a local index providing faster access to the data.

Synchronicity does not yet distinguish between MIME types but rather assumes all encountered resources to be HTML formatted. Even though it may be able to provide Mementos of non-HTML resources functions such as title extraction and keyword generation may not work properly.

Synchronicity's internal list of stop titles is static right now. Surely users will encounter other titles that can be categorized as stop titles and hence should be able to add these to the list.

We also consider a command line implementation of Synchronicity for future work. Such a system would not be affected by user interface constrains and could serve as a reference implementation to demonstrate the entire spectrum of results. This system would particularly be useful for curators of collections of digital resources when automatically testing for missing pages and looking for alternatives. Synchronicity could be used for batch processing and would hence expedite these processes.

Synchronicity implemented as a web service is another plausible system for future development. Instead of having a client side system that users need to download and install we could offer a centralized service accessible through the web and easy to use for everyone.

The disadvantage of a client sided Synchronicity is that users can not benefit from the (re-)discovery experience of other Synchronicity users. We consider a scenario where different users encounter the same missing page as very likely and hence it would be convenient if they could share and learn from each other's outcome. We envision a (web) service that provides alternative URIs to known missing pages. Synchronicity could access that service (much like a cache) and offer an alternative URI, if available, to the user before it starts its own (re-)discovery process. Such a service would most likely imply a $n : m$ much rather than a $1 : 1$ mapping.

1.4 Enhanced Link Neighborhood Lexical Signatures

We have shown in Chapter IX that the anchor text of neighboring pages is most useful to generate well performing lexical signatures. Similar to stop words and the previously defined stop titles we can also imagine the existence of "stop anchors". This aspect should be investigated using a larger corpus but positive results could be implemented in Synchronicity as well.

We have only considered the top 10 , top 100 and top 1000 backlinks to draw the anchor text from. It remains the subject of future research to investigate the optimum number of backlinks. It is entirely possible that including some number of pages between ten and 100 proves to be better than only ten.

Alternatively it might result in a better performance to set the threshold as the number of anchor terms instead of number of backlink pages. For example, the parameter could be to aggregate 50 anchor terms (all candidate to make it into the link neighborhood lexical signature) from however many backlink pages it takes.

2 CONCLUSIONS

The Internet as an information space is huge and with that scale there will always be broken references. Maybe it is the bookmark created some time ago or a link from a poorly maintained website, but linkrot is verly likely to remain an undesired part of our web browsing experience.

In this dissertation we introduce several methods to overcome this detriment. Our idea is based on the intuition that pages often do not completely disappear but rather just move to a different URI. We approach the problem while it happens which means in real time we try to discover the page of interest at its new URI or offer satisfying alternative pages while the user is browsing.

We utilize the Memento framework to obtain previous versions of now missing pages. We distill the "aboutness" of these archived copies in form of their titles (Chapter VI) and lexical signatures (Chapters IV and V) and apply it as search engine queries. Alternatively, especially if no Mementos are available, we obtain tags (Chapter VII) and generate lexical signatures based on the missing page's link neighborhood (Chapter VIII) to issue further search engine queries. This work shows that with the resulting set of URIs we have a very good chance of fulfill the user's information need.

Synchronicity, our web browser extension, brings all these methods together to support the user while browsing the web. It is not in competition with search engines but rather attempts to provide the best possible input in order to fully utilizes their performance.

Of course we can not prevent 404 “Page not Found” errors from happening and there is no single web supervising institution that can, but with this work we contribute to overcoming the unwanted dead ends while browsing the web.

2.1 Contributions

This dissertation makes a number of significant contributions to the field of digital preservation, in particular the preservation of web pages at the time they are noticed missing by applying information retrieval techniques.

1. This work identifies the ubiquitous problem of 404 “Page not Found” errors and offers a complex solution bundling several content and link based methods.

2. The dissertation introduces a reliable real time approach to estimate document frequency values and evaluates it against existing well performing baselines. That includes a thorough correlation analysis between document frequency and the commonly used term count values.

3. This research results in a framework for the generation of lexical signatures. It experimentally validates the optimal length and maximum age to ensure a satisfactory outcome when using lexical signatures as search engine queries.

4. This dissertation further provides guidance to use a web page’s title for rediscovering missing web pages. It analyzes the change of titles over time and compares it to page content change. It introduces the notion of stop titles and provides an initial list of such (in a retrieval sense) poor titles.

5. This work experimentally evaluates the retrieval performance of tag based search engine queries, particularly with respect to an optimal length. This research is responsible for the discovery of ghost tags.

6. Furthermore this work explores link neighborhood based lexical signatures and offers a framework for their creation. It specifies optimal parameters such as length, number of backlinks and their backlink level to include as well as the textual backlink radius.

7. The software contribution of this dissertation is the implementaion of Synchronicity as a Mozilla Firefox extension. It brings all theoretical contributions together and is available to users as open source software.

8. The Book of the Dead is another noteworthy contribution of this work. The corpus is available to researchers and it includes the determined contextual “aboutness” of all missing URIs.

REFERENCES

[1] Amazon Mechanical Turk. http://www.mturk.com

[2] Architecture of the World Wide Web. http://www.w3.org/TR/webarch/

[3] Bing Search API. http://msdn.microsoft.com/en-us/library/dd251056.aspx

[4] Bitly. http://bit.ly/

[5] bookmarking done right: historious, your personal search engine - historious. http://historio.us/

[6] Connotea: free online reference management for clinicians and scientists. http://www.connotea.org/

[7] Delicious. http://www.delicious.com

[8] Delicious Integrated Into Yahoo Search Results. http://techcrunch.com/2008/01/19/delicious-integrated-into-yahoo-search-results/

[9] Enquiro Research. http://www.enquiroresearch.com/

[10] Errorzilla - Useful error pages for Firefox. http://www.jaybaldwin.com/Blog.aspx?cid=4

[11] Google - Crawl Errors Now Reports Soft 404s. http://googlewebmastercentral.blogspot.com/2010/06/crawl-errors-now-reports-soft-404s.html

[12] Google - Farewell to Soft 404s. http://googlewebmastercentral.blogspot.com/2008/08/farewell-to-soft-404s.html

[13] Google Bookmarks. https://www.google.com/bookmarks/

[14] Google Custom Search API. http://code.google.com/apis/customsearch/v1/overview.html

[15] Google SOAP Search API. http://code.google.com/apis/soapsearch/reference.html

[16] Google's Soft 404s are Inaccurate and Often Times Outdated. http://x-pose.org/2010/06/googles-soft-404s-are-inaccurate-and-often-times-outdated/

[17] Google's Soft 404s Errors Also Include 5xx Errors. http://www.seroundtable.com/archives/022396.html

[18] How does Google calculate the number of results? http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=70920

[19] IMDb - Repo Man (1984). http://www.imdb.com/title/tt0087995/

[20] Internet Archive Wayback Machine. http://waybackmachine.org/

[21] JSON. http://www.json.org/

[22] Latest on Yahoo! Search BOSS . http://www.ysearchblog.com/2011/02/08/latest-on-boss/

[23] Library of Congress. http://www.loc.gov/

[24] Memento: Adding Time to the Web. http://www.mementoweb.org/

[25] MySQL - The World's Most Popular Open Source Database. http://www.mysql.com/

[26] NII Test Collection for IR Systems. http://research.nii.ac.jp/ntcir/index-en.html

[27] ODP Open Directory Project. http://www.dmoz.org

[28] redhat.com — The World's Open Source Leader. http://www.redhat.com/

[29] Simile Widgets. http://www.simile-widgets.org/timeline/

[30] Synchronicity. https://addons.mozilla.org/en-US/firefox/addon/synchronicity/

[31] The Apache HTTP Server Project. http://httpd.apache.org/

[32] The Book of the Dead. http://ws-dl.blogspot.com/2011/06/201-06-17-book-of-dead-corpus.html

[33] The Coral Content Distribution Network. http://www.coralcdn.org/

[34] The Mozilla Project. http://www.mozilla.org/

[35] The Perl Programming Language. http://www.perl.org/

[36] The R Project for Statistical Computing. http://www.r-project.org/

[37] The size of the World Wide Web. http://www.worldwidewebsize.com/

[38] TREC BLOG08 Test Collection. http://ir.dcs.gla.ac.uk/test_collections/blogs08info.html

[39] TREC SPAM Track Guidelines. http://plg.uwaterloo.ca/~gvcormac/spam/

[40] W3C SOAP Tutorial. http://www.w3schools.com/soap/default.asp

[41] W3C XML Tutorial. http://www.w3schools.com/xml/default.asp

[42] WaCKy. http://wacky.sslmit.unibo.it/doku.php

[43] WebKB, The 4 Universities Data Set. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

[44] Yahoo! - Why is your crawler asking for strange URLs that have never existed on my site? http://help.yahoo.com/l/uk/yahoo/search/webcrawler/slurp-10.html

[45] Yahoo! BOSS Search API. http://developer.yahoo.com/search/boss/

[46] Abiteboul, S., Cobena, G., Masanes, J., Sedrati, G.: A First Experience in Archiving the French Web. In: Proceedings of ECDL '02, pp. 1–15 (2002)

[47] Abrams, M., Standridge, C.R., Abdulla, G., Fox, E.A., Williams, S.: Removal Policies in Network Caches for World-Wide Web Documents. In: Proceedings of SIGCOMM '96, pp. 293–305 (1996)

[48] Adamic, L.A., Huberman, B.A.: Power-Law Distribution of the World Wide Web. Science **287**(5461), 2115 (2000)

[49] Adamic, L.A., Huberman, B.A.: Zipf's Law and the Internet. Glottometrics **3**, 143–150 (2002)

[50] Adar, E., Dontcheva, M., Fogarty, J., Weld, D.S.: Zoetrope: Interacting with the Ephemeral Web. In: Proceedings of UIST '08, pp. 239–248 (2008)

[51] Adar, E., Teevan, J., Dumais, S.T.: Large Scale Analysis of Web Revisitation Patterns. In: Proceeding of CHI '08, pp. 1197–1206 (2008)

[52] Adar, E., Teevan, J., Dumais, S.T.: Resonance on the Web: Web Dynamics and Revisitation Patterns. In: Proceedings of CHI '09, pp. 1381–1390 (2009)

[53] Adar, E., Teevan, J., Dumais, S.T., Elsas, J.L.: The Web Changes Everything: Understanding the Dynamics of Web Content. In: Proceedings of WSDM '09, pp. 282–291 (2009)

[54] Agichtein, E., Zheng, Z.: Identifying "Best Bet" Web Search Results by Mining Past User Behavior. In: Proceedings of KDD '06, pp. 902–908 (2006)

[55] Ainsworth, S.G., AlSum, A., SalahEldeen, H., Weigle, M.C., Nelson, M.L.: How Much of the Web Is Archived? In: Proceedings of JCDL'11 (2011)

[56] Alonso, O., Mizzaro, S.: Can We Get Rid of TREC Assessors? Using Mechanical Turk for Relevance Assessment. In: SIGIR '09: Workshop on The Future of IR Evaluation, pp. 15–16 (2009)

[57] Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., Raghavan, S.: Searching the Web. ACM Transactions of Internet Technology **1**, 2–43 (2001)

[58] Ashman, H.: Electronic document addressing: Dealing with change. ACM Computing Surveys **32**(3), 201–212 (2000)

[59] Ashman, H., Davis, H., Whitehead, J., Caughey, S.: Missing the 404: Link integrity on the world wide web. In: Proceedings of WWW '98, pp. 761–762 (1998)

[60] Baeza-Yates, R., Caldern-Benavides, L., Gonzlez-Caro, C.: The Intention Behind Web Queries. In: Proceedings of SPIRE '06, pp. 98–109 (2006)

[61] Baeza-Yates, R., Álvaro Pereira, Ziviani, N.: Genealogical trees on the web: a search engine user perspective. In: Proceedings of WWW '08, pp. 367–376 (2008)

[62] Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)

[63] Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., Su, Z.: Optimizing Web Search Using Social Annotations. In: Proceedings of WWW '07, pp. 501–510 (2007)

[64] Bar-Ilan, J., Mat-Hassan, M., Levene, M.: Methods for Comparing Rankings of Search Engine Results. Computer Networks **50**(10), 1448–1463 (2006). doi: http://dx.doi.org/10.1016/j.comnet.2005.10.020

[65] Bar-Yossef, Z., Broder, A.Z., Kumar, R., Tomkins, A.: Sic Transit Gloria Telae: Towards an Understanding of the Web's Decay. In: Proceedings of WWW '04, pp. 328–337 (2004)

[66] Berger, A.L., Pietra, V.J.D., Pietra, S.A.D.: A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics **22**, 39–71 (1996)

[67] Bergmark, D.: Collection Synthesis. In: Proceedings of JCDL '02, pp. 253–262 (2002)

[68] Bergmark, D., Lagoze, C., Sbityakov, A.: Focused Crawls, Tunneling, and Digital Libraries. In: Proceedings of ECDL '02, pp. 91–106 (2002)

[69] Berners-Lee, T.: Cool URIs don't change (1998). `http://www.w3.org/Provider/Style/URI.html`

[70] Bharat, K., Broder, A.: A technique for measuring the relative size and overlap of public web search engines. Computer Networks ISDN Systems **30**(1-7), 379–388 (1998)

[71] Bischoff, K., Firan, C., Nejdl, W., Paiu, R.: Can All Tags Be Used for Search? In: Proceedings of CIKM '08, pp. 193–202 (2008)

[72] Bogen, P., Pogue, D., Poursardar, F., Shipman, F., Furuta, R.: WPv4: A Re-imagined Waldens Paths to Support Diverse User Communities. In: Proceedings of JCDL '11 (2011)

[73] Brewington, B., Cybenko, G.: Keeping Up With the Changing Web. Computer **33**(5), 52–58 (2000)

[74] Brin, S.: Near neighbor search in large metric spaces. In: Proceedings of VLDB '95, pp. 574–584 (1995)

[75] Brin, S., Davis, J., Garcia-Molina, H.: Copy detection mechanisms for digital documents. In: Proceedings of SIGMOD '95, pp. 398–409 (1995)

[76] Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems **30**(1-7), 107–117 (1998)

[77] Broder, A.Z.: On the Resemblance and Containment of Documents. In: Proceedings of SEQUENCES '97, pp. 21–29 (1997)

[78] Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G.: Syntactic Clustering of the Web. Computer Networks **29**(8-13), 1157–1166 (1997)

[79] Buckley, C.: Automatic Query Expansion Using SMART : TREC 3. In: Proceedings of TREC '94, pp. 69–80 (1994)

[80] Buckley, C., Salton, G., Allan, J.: The Effect of Adding Relevance Information in a Relevance Feedback Environment. In: Proceedings of SIGIR '94, pp. 292–300 (1994)

[81] Chakrabarti, D., Kumar, R., Punera, K.: Generating Succinct Titles for Web URLs. In: Proceeding of KDD '08, pp. 79–87 (2008)

[82] Chakrabarti, S., van den Berg, M., Dom, B.: Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. Computer Networks **31**(11-16), 1623–1640 (1999)

[83] Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of STOC '02, pp. 380–388 (2002)

[84] Chiang, W.T.M., Hagenbuchner, M., Tsoi, A.C.: The WT10G Dataset and the Evolution of the Web. In: Proceedings of WWW '05, pp. 938–939 (2005)

[85] Cho, J., Garcia-Molina, H.: The Evolution of the Web and Implications for an Incremental Crawler. In: Proceedings of VLDB '00, pp. 200–209 (2000)

[86] Cho, J., Garcia-Molina, H.: Estimating Frequency of Change. ACM Transactions on Internet Technology **3**, 256–290 (2003)

[87] Christensen, N.H.: Preserving the Bits of the Danish Internet. In: Proceedings of IWAW '05 (2005)

[88] Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A Comparison of String Distance Metrics for Name-Matching Tasks. In: Proceedings of IJCAI '03, pp. 73–78 (2003)

[89] Craswell, N., Hawking, D., Robertson, S.: Effective Site Finding Using Link Anchor Information. In: Proceedings of SIGIR '01, pp. 250–257 (2001)

[90] Croft, B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice, 1st edn. Addison-Wesley Publishing Company, USA (2009)

[91] Dai, N., Davison, B.D.: Mining Anchor Text Trends for Retrieval. In: Proceedings of ECIR '10, pp. 127–139 (2010)

[92] Davis, H.C.: Referential integrity of links in open hypermedia systems. In: Proceedings of HYPERTEXT '98, pp. 207–216 (1998)

[93] Davis, H.C.: Hypertext Link Integrity. ACM Computing Surveys **31** (1999). doi: http://doi.acm.org/10.1145/345966.346026

[94] Davison, B.D.: Topical Locality in the Web. In: Proceedings of SIGIR '00, pp. 272–279 (2000)

[95] Dean, J., Henzinger, M.R.: Finding Related Pages in the World Wide Web. Computer Networks **31**(11-16), 1467–1479 (1999). doi: http://dx.doi.org/10.1016/S1389-1286(99)00022-5

[96] Dellavalle, R.P., Hester, E.J., Heilig, L.F., Drake, A.L., Kuntzman, J.W., Graber, M., Schilling, L.M.: Information Science: Going, Going, Gone: Lost Internet References. Science **302**(5646), 787–788 (2003). doi: http://dx.doi.org/10.1126/science.1088234

[97] Dice, L.R.: Measures of the Amount of Ecologic Association Between Species. Ecology **26**(3), 297–302 (1945)

[98] Dou, Z., Song, R., Nie, J.Y., Wen, J.R.: Using Anchor Texts with Their Hyperlink Structure for Web Search. In: Proceedings of SIGIR '09, pp. 227–234 (2009)

[99] Douglis, F., Feldmann, A., Krishnamurthy, B., Mogul, J.C.: Rate of Change and other Metrics: a Live Study of the World Wide Web. In: USENIX Symposium on Internet Technologies and Systems (1997)

[100] Doyle, P.G., Snell, J.L.: Random Walks and Electrical Networks (Carus Mathematical Monographs). Mathematical Assn of America (1984)

[101] Eastlake III, D.E., Jones, P.E.: US Secure Hash Algorithm 1 (SHA1) RFC-3174 (2001)

[102] Edelman, B.: Domains Reregistered for Distribution of Unrelated Content − A Case Study of "Tina's Free Live Webcam". `http://cyber.law.harvard.edu/archived_content/people/edelman/renewals/`

[103] Eiron, N., McCurley, K.S.: Analysis of Anchor Text for Web Search. In: Proceedings of SIGIR '03, pp. 459–460 (2003)

[104] Fagin, R., Kumar, R., Sivakumar, D.: Comparing top k Lists. In: Proceedings of SODA '03, pp. 28–36 (2003)

[105] Fang, H.R., Murphy, K., Jin, Y., Kim, J.S., White, P.S.: Human Gene Name Normalization Using Text Matching With Automatically Extracted Synonym Dictionaries. In: Proceedings of BioNLP '06, pp. 41–48 (2006)

[106] Fetterly, D., Manasse, M., Najork, M.: On the evolution of clusters of near-duplicate web pages. In: Proceedings of LA-WEB '03, pp. 37–45 (2003)

[107] Fetterly, D., Manasse, M., Najork, M., Wiener, J.: A Large-Scale Study of the Evolution of Web Pages. In: Proceedings of WWW '03, pp. 669–678 (2003)

[108] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Hypertext Transfer Protocol – HTTP/1.1 RFC-2612 (1999). Updated by RFC 2817

[109] Fox, C.: Lexical Analysis and Stoplists. In: W.B. Frakes, B.R. Yates (eds.) Information Retrieval: Data Structures and Algorithms, pp. 102–130. Englewood Cliffs, NJ: Prentice Hall (1992)

[110] Frakes, W.B., Baeza-Yates, R.A. (eds.): Information Retrieval: Data Structures and Algorithms. Prentice-Hall (1992)

[111] Francisco-Revilla, L., Shipman, F., Furuta, R., Karadkar, U., Arora, A.: Managing Change on the Web. In: Proceedings of JCDL ’01, pp. 67–76 (2001)

[112] Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-Specific Keyphrase Extraction. In: Proceedings of IJCAI ’99, pp. 668–673 (1999)

[113] Franz, A., Brants, T.: All Our N-Gram are Belong to You. `http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html`

[114] Fujii, A., Itou, K., Akiba, T., Ishikawa, T.: Exploiting Anchor Text for the Navigational Web Retrieval at NTCIR-5. In: Proceedings of NTCIR-5 ’05, pp. 455–462 (2005)

[115] Fuxman, A., Tsaparas, P., Achan, K., Agrawal, R.: Using the Wisdom of the Crowds for Keyword Generation. In: Proceeding of WWW ’08, pp. 61–70 (2008)

[116] Gilpin, A.R.: Table for Conversion of Kendall’s Tau to Spearman’s Rho Within the Context of Measures of Magnitude of Effect for Meta-Analysis. Educational and Psychological Measurement **53**(1), 87–92 (1993). doi: 10.1177/0013164493053001007

[117] Harman, D.: Relevance Feedback Revisited. In: Proceedings of SIGIR ’92, pp. 1–10 (1992)

[118] Harmandas, V., Sanderson, M., Dunlop, M.D.: Image Retrieval by Hypertext Links. In: Proceedings of SIGIR ’97, pp. 296–303 (1997)

[119] Harrison, T.: Opal: In Vivo Based Preservation Framework for Locating Lost Web Pages. Master’s thesis, Old Dominion University (2005)

[120] Harrison, T.L., Nelson, M.L.: Just-in-Time Recovery of Missing Web Pages. In: Proceedings of HYPERTEXT ’06, pp. 145–156 (2006)

[121] Haslhofer, B., Popitsch, N.: DSNotify - Detecting and Fixing Broken Links in Linked Data Sets. In: Proceedings of DEXA ’09, pp. 89–93 (2009)

[122] Hayes, C., Avesani, P.: Using Tags and Clustering to Identify Topic-Relevant Blogs. In: Proceedings of ICWSM ’07, pp. 67–75 (2007)

[123] Hayes, C., Avesani, P., Veeramachaneni, S.: An Analysis of the Use of Tags in a Blog Recommender System. In: Proceedings of IJCAI ’07, pp. 2772–2777 (2007)

[124] Henzinger, M.: Finding near-duplicate web pages: a large-scale evaluation of algorithms. In: Proceedings of SIGIR ’06, pp. 284–291 (2006)

[125] Henzinger, M., Chang, B.W., Milch, B., Brin, S.: Query-free News Search. In: Proceedings of WWW ’03, pp. 1–10 (2003)

[126] Heymann, P., Koutrika, G., Garcia-Molina, H.: Can Social Bookmarking Improve Web Search? In: Proceedings of WSDM ’08, pp. 195–206 (2008)

[127] Hirsch, S.G.: How do Children Find Information on Different Types of Tasks? Children’s Use of the Science Library Catalog. Library Trends **45**(4), 725–745 (1997)

[128] Holtman, K., Mutz, A.: Transparent Content Negotiation in HTTP RFC-2295 (1998)

[129] Hulth, A.: Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In: Proceedings of EMNLP '03, pp. 216–223 (2003)

[130] Hyusein, B., Patel, A.: Web Document Indexing and Retrieval. In: Proceedings of CICLing'03, pp. 573–579 (2003)

[131] Jansen, B.J., Booth, D.L., Spink, A.: Determining the Informational, Navigational, and Transactional Intent of Web Queries. Information Processing and Management **44**(3), 1251–1266 (2008)

[132] Jansen, B.J., Spink, A.: How are we Searching the World Wide Web? A Comparison of Nine Search Engine Transaction Logs. Information Processing Management **42**(1), 248–263 (2006)

[133] Jansen, B.J., Spink, A., Saracevic, T.: Real Life, Real Users, and Real Needs: A Study and Analysis of User Queries on the Web. Information Processing and Management **36**(2), 207–227 (2000). doi: http://dx.doi.org/10.1016/S0306-4573(99)00056-4

[134] Jaro, M.A.: Advances in Record Linkage Methodology as Applied to the 1985 Census of Tampa Florida. Journal of the American Statistical Association **84**(406), 414–420 (1989)

[135] Jaro, M.A.: Probabilistic Linkage of Large Public Health Data Files. Statistics in Medicine **14**(5-7), 491–498 (1995)

[136] Järvelin, K., Kekäläinen, J.: IR Evaluation Methods for Retrieving Highly Relevant Documents. In: Proceedings of SIGIR '00, pp. 41–48 (2000)

[137] Järvelin, K., Kekäläinen, J.: Cumulated Gain-Based Evaluation of IR Techniques. ACM Transactions on Information Systems **20**(4), 422–446 (2002)

[138] Jason Morrison, P.: Tagging and Searching: Search Retrieval Effectiveness of Folksonomies on the World Wide Web. Information Processing and Management **44**(4), 1562–1579 (2008)

[139] Jatowt, A., Kawai, Y., Nakamura, S., Kidawara, Y., Tanaka, K.: A Browser for Browsing the Past Web. In: Proceedings of WWW '06, pp. 877–878 (2006)

[140] Ji, S., Li, G., Li, C., Feng, J.: Efficient Interactive Fuzzy Keyword Search. In: Proceedings of WWW '09, pp. 371–380 (2009)

[141] Jiang, S., Zilles, S., Holte, R.: Query Suggestion by Query Search: A New Approach to User Support in Web Search. In: Proceedings of WI-IAT '09, pp. 679–684 (2009)

[142] Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately Interpreting Clickthrough Data as Implicit Feedback. In: Proceedings of SIGIR '05, pp. 154–161 (2005)

[143] Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., Gay, G.: Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search. ACM Transactions on Information Systems **25**(2), 7 (2007). doi: http://doi.acm.org/10.1145/1229179.1229181

[144] Jones, K.S.: A Statistical Interpretation of Term Specificity and its Application in Retrieval. Journal of Documentation **28**, 11–21 (1972)

[145] Jones, K.S.: Index Term Weighting. Information Storage and Retrieval **9**(11), 619–633 (1973)

[146] Jung, C.G.: The Structure and Dynamics of the Psyche, 2 edn. Princeton University Press (1970)

[147] Kahle, B.: Preserving the Internet. Scientific American **276**, 82–83 (1997)

[148] Kan, M.Y.: Web Page Classification Without the Web Page. In: Proceedings of WWW '04, pp. 262–263 (2004)

[149] Kan, M.Y., Thi, H.O.N.: Fast Webpage Classification Using URL Features. In: Proceedings of CIKM '05, pp. 325–326 (2005)

[150] Kelleher, D., Luz, S.: Automatic Hypertext Keyphrase Detection. In: Proceedings of IJCAI '05, pp. 1608–1609 (2005)

[151] Keller, F., Lapata, M.: Using the Web to Obtain Frequencies for Unseen Bigrams. Computational Linguistics **29**(3), 459–484 (2003)

[152] Kendall, M.G.: A New Measure of Rank Correlation. Biometrika **30**(1-2), 81–93 (1938). doi: 10.1093/biomet/30.1-2.81

[153] Kendall, M.G. (ed.): Rank Correlation Methods. Griffin (1948)

[154] Khan, K., Locatis, C.: Searching Through Cyberspace: The Effects of Link Display and Link Density on Information Retrieval from Hypertext on the World Wide Web. Journal of the American Society of Information Science **49**(2), 176–182 (1998)

[155] Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing User Studies With Mechanical Turk. In: Proceeding of CHI '08, pp. 453–456 (2008)

[156] Klein, M., Emara, M., Nelson, M.L.: Synchronicity − Automatically Rediscover Missing Web Pages in Real Time. In: Proceedings of JCDL '11 (2011)

[157] Klein, M., Nelson, M.L.: A Comparison of Techniques for Estimating IDF Values to Generate Lexical Signatures for the Web. In: Proceeding of WIDM '08, pp. 39–46 (2008)

[158] Klein, M., Nelson, M.L.: Revisiting Lexical Signatures to (Re-)Discover Web Pages. In: Proceedings of ECDL '08, pp. 371–382 (2008)

[159] Klein, M., Nelson, M.L.: Correlation of Term Count and Document Frequency for Google N-Grams. In: Proceedings of ECIR '09, pp. 620–627 (2009)

[160] Klein, M., Nelson, M.L.: Inter-Search Engine Lexical Signature Performance. In: Proceedings of JCDL '09, pp. 413–414 (2009)

[161] Klein, M., Nelson, M.L.: Evaluating Methods to Rediscover Missing Web Pages from the Web Infrastructure. In: Proceedings of JCDL '10, pp. 59–68 (2010)

[162] Klein, M., Shipman, J., Nelson, M.L.: Is This a Good Title? In: Proceedings of Hypertext '10, pp. 3–12 (2010)

[163] Klein, M., Ware, J., Nelson, M.L.: Rediscovering Missing Web Pages Using Link Neighborhood Lexical Signatures. In: Proceedings of JCDL '11 (2011)

[164] Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. Journal of the ACM **46**(5), 604–632 (1999)

[165] Klöckner, K., Wirschum, N., Jameson, A.: Depth- and Breadth-First Processing of Search Result Lists. In: Proceedings of CHI '04, pp. 1539–1539 (2004)

[166] Koehler, W.C.: Web Page Change and Persistence - A Four-Year Longitudinal Study. Journal of the American Society for Information Science and Technology **53**(2), 162–171 (2002)

[167] Kolcz, A., Chowdhury, A., Alspector, J.: Improved Robustness of Signature-Based Near-Replica Detection via Lexicon Randomization. In: Proceedings of KDD '04, pp. 605–610 (2004)

[168] Kraft, R., Zien, J.: Mining Anchor Text for Query Refinement. In: Proceedings of WWW '04, pp. 666–674 (2004)

[169] Krause, B., Hotho, A., Stumme, G.: A Comparison of Social Bookmarking with Traditional Search. In: Proceedings of ECIR '08, pp. 101–113 (2008)

[170] Lagoze, C., Van de Sompel, H.: The Open Archives Initiative: building a low-barrier interoperability framework. In: Proceedings of JCDL '01, pp. 54–62 (2001)

[171] Lagoze, C., Van de Sompel, H., Nelson, M.L., Warner, S., Sanderson, R., Johnston, P.: A Web-Based Resource Model for eScience: Object Reuse & Exchange. Tech. rep. (2008)

[172] Lagoze, C., Van de Sompel, H., Nelson, M.L., Warner, S., Sanderson, R., Johnston, P.: Object Re-Use & Exchange: A Resource-Centric Approach. Tech. rep. (2008)

[173] Lampos, C., Eirinaki, M., Jevtuchova, D., Vazirgiannis, M.: Archiving the Greek Web. In: Proceedings of IWAW '04 (2004)

[174] Lawrence, S., Pennock, D.M., Flake, G.W., Krovetz, R., Coetzee, F.M., Glover, E., Nielsen, F.A., Kruger, A., Giles, C.L.: Persistence of Web References in Scientific Research. Computer **34**(2), 26–31 (2001)

[175] Lazonder, A.W., Biemans, H.J.A., Wopereis, I.G.J.H.: Differences Between Novice and Experienced Users in Searching Information on the World Wide Web. Journal of the American Society for Information Science **51**(6), 576–581 (2000)

[176] Leech, G., Grayson, L.P., Wilson, A.: Word Frequencies in Written and Spoken English: based on the British National Corpus. Longman, London (2001)

[177] Levenshtein, V.I.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady **10**(8), 707–710 (1966)

[178] Li, S.Z.: Markov Random Field Modeling in Computer Vision. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1995)

[179] Lim, L., Wang, M., Padmanabhan, S., Vitter, J.S., Agarwal, R.C.: Characterizing Web Document Change. In: Proceedings of WAIM '01, pp. 133–144 (2001)

[180] Ling, Y., Meng, X., Meng, W.: Automated extraction of hit numbers from search result pages. In: Proceedings of WAIM '06, pp. 73–84 (2006)

[181] Manku, G.S., Jain, A., Sarma, A.D.: Detecting Near-Duplicates for Web Crawling. In: Proceedings of WWW '07, pp. 141–150 (2007)

[182] Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)

[183] Marchionini, G.: Information Seeking in Electronic Environments. Cambridge University Press, New York, NY, USA (1995)

[184] Markwell, J., Brooks, D.W.: Broken links: The ephemeral nature of educational www hyperlinks. Journal of Science Education and Technology **11**, 105–108(4) (2002)

[185] Martin, J.D., Holte, R.: Searching for Content-Based Addresses on the World-Wide Web. In: Proceedings of DL '98, pp. 299–300 (1998)

[186] Martinez-Romo, J., Araujo, L.: Recommendation System for Automatic Recovery of Broken Web Links. In: Proceedings of IBERAMIA '08, pp. 302–311 (2008)

[187] Martinez-Romo, J., Araujo, L.: Retrieving Broken Web Links Using an Approach Based on Contextual Information. In: Proceedings of HT '09, pp. 351–352 (2009)

[188] Martinez-Romo, J., Araujo, L.: Analyzing Information Retrieval Methods to Recover Broken Web Links. In: Proceedings of ECIR '10, pp. 26–37 (2010)

[189] McCown, F.: Lazy Preservation: Reconstructing Websites from the Web Infrastructure. Ph.D. thesis, Old Dominion University (2007)

[190] McCown, F., Chan, S., Nelson, M.L., Bollen, J.: The Availability and Persistence of Web References in D-Lib Magazine. In: Proceedings of IWAW'05 (2005)

[191] McCown, F., Diawara, N., Nelson, M.L.: Factors Affecting Website Reconstruction from the Web Infrastructure. In: Proceedings of JCDL '07, pp. 39–48 (2007)

[192] McCown, F., Nelson, M.L.: Agreeing to Disagree: Search Engines and their Public Interfaces. In: Proceedings of JCDL '07, pp. 309–318 (2007)

[193] McCown, F., Nelson, M.L.: Characterization of Search Engine Caches. In: Proceedings of IS&T Archiving '07, pp. 48–52 (2007)

[194] McCown, F., Nelson, M.L.: Search Engines and Their Public Interfaces: Which APIs are the Most Synchronized? In: Proceedings of WWW '07, pp. 1197–1198 (2007)

[195] McCown, F., Nelson, M.L.: A Framework for Describing Web Repositories. In: Proceedings of JCDL '09, pp. 341–344 (2009)

[196] McCown, F., Smith, J.A., Nelson, M.L.: Lazy Preservation: Reconstructing Websites by Crawling the Crawlers. In: Proceedings of WIDM '06, pp. 67–74 (2006)

[197] McDonald, S., Stevenson, R.J.: Navigation in Hyperspace: An Evaluation of the Effects of Navigational Tools and Subject Matter Expertise on Browsing and Information Retrieval in Hypertext. Interacting with Computers **10**(2), 129–142 (1998)

[198] Metzler, D., Novak, J., Cui, H., Reddy, S.: Building Enriched Document Representations Using Aggregated Anchor Text. In: Proceedings of SIGIR '09, pp. 219–226 (2009)

[199] Mishne, G., de Rijke, M.: A Study of Blog Search. In: Proceedings of ECIR '06, pp. 289–301 (2006)

[200] Morishima, A., Nakamizo, A., Iida, T., Sugimoto, S., Kitagawa, H.: Pagechaser: A tool for the automatic correction of broken web links. In: Proceedings of ICDE '08, pp. 1486–1488 (2008)

[201] Morishima, A., Nakamizo, A., Iida, T., Sugimoto, S., Kitagawa, H.: Bringing Your Dead Links Back to Life: A Comprehensive Approach and Lessons Learned. In: Proceedings of HT '09, pp. 15–24 (2009)

[202] Morishima, A., Nakamizo, A., Iida, T., Sugimoto, S., Kitagawa, H.: Why Are Moved Web Pages Difficult to Find?: The WISH Approach. In: Proceedings of WWW '09, pp. 1117–1118 (2009)

[203] Myers, J.L., Well, A. (eds.): Research Design and Statistical Analysis. Lawrence Erlbaum (1995)

[204] Nakamizo, A., Iida, T., Morishima, A., Sugimoto, S., Kitagawa, H.: A tool to compute reliable web links and its applications. In: Proceedings of ICDEW '05, p. 1255 (2005)

[205] Nakov, P., Hearst, M.: Search Engine Statistics Beyond the N-Gram: Application to Noun Compound Bracketing. In: Proceedings of CONLL '05, pp. 17–24 (2005)

[206] Nakov, P., Hearst, M.: A Study of Using Search Engine Page Hits as a Proxy for n-gram Frequencies. In: Proceedings of RANLP '05 (2005)

[207] Negulescu, K.C.: Web Archiving @ The Internet Archive. `http://www.digitalpreservation.gov/news/events/ndiipp_meetings/ndiipp10/docs/July21/session09/NDIIPP072110FinalIA.ppt`

[208] Nelson, M.L., Allen, B.D.: Object Persistence and Availability in Digital Libraries. D-Lib Magazine **8**(1) (2002). doi: http://dx.doi.org/10.1045/january2002-nelson

[209] Nelson, M.L., McCown, F., Smith, J.A., Klein, M.: Using the Web Infrastructure to Preserve Web Pages. IJDL **6**(4), 327–349 (2007)

[210] Ntoulas, A., Cho, J., Olston, C.: What's New on the Web?: The Evolution of the Web from a Search Engine Perspective. In: Proceedings of WWW '04, pp. 1–12 (2004)

[211] Ntoulas, A., Najork, M., Manasse, M., Fetterly, D.: Detecting Spam Web Pages Through Content Analysis. In: Proceedings of WWW '06, pp. 83–92 (2006)

[212] Papineni, K.: Why Inverse Document Frequency? In: Proceedings of NAACL '01, pp. 1–8 (2001)

[213] Park, S.T., Pennock, D.M., Giles, C.L., Krovetz, R.: Analysis of Lexical Signatures for Finding Lost or Related Documents. In: Proceedings of SIGIR '02, pp. 11–18 (2002)

[214] Park, S.T., Pennock, D.M., Giles, C.L., Krovetz, R.: Analysis of Lexical Signatures for Improving Information Persistence on the World Wide Web. ACM Transactions on Information Systems **22**(4), 540–572 (2004). doi: http://doi.acm.org/10.1145/1028099.1028101

[215] Paskin, N.: Digital Object Identifiers. Information Services and Use **22**(2-3), 97–112 (2002)

[216] Pass, G., Chowdhury, A., Torgeson, C.: A Picture of Search. In: Proceedings of InfoScale '06 (2006)

[217] Patel, S.C., Drury, C.C., Shalin, V.L.: Effectiveness of Expert Semantic Knowledge as a Navigational Aid Within Hypertext. Behaviour and Information Technology **17**, 313–324 (1998)

[218] Phelps, T.A., Wilensky, R.: Robust Hyperlinks Cost Just Five Words Each. Tech. Rep. UCB//CSD-00-1091, University of California at Berkeley, Berkeley, CA, USA (2000)

[219] Popitsch, N.P., Haslhofer, B.: DSNotify: Handling Broken Links in the Web of Data. In: Proceedings of WWW '10, pp. 761–770 (2010)

[220] Porter, M.F.: An Algorithm for Suffix Stripping. Electronic Library and Information Systems **14**(3), 130–137 (1980)

[221] Qin, J., Zhou, Y., Chau, M.: Building Domain-Specific Web Collections for Scientific Digital Libraries: A Meta-Search Enhanced Focused Crawling Method. In: Proceedings of JCDL '04, pp. 135–141 (2004)

[222] Qiu, Y., Frei, H.P.: Concept Based Query Expansion. In: Proceedings of SIGIR '93, pp. 160–169 (1993)

[223] Rauber, A., Aschenbrenner, A., Witvoet, O.: Austrian Online Archive Processing: Analyzing Archives of the World Wide Web. In: Proceedings of ECDL '02, pp. 16–31 (2002)

[224] Rivest, R.: The MD5 Message-Digest Algorithm RFC-1321 (1992)

[225] Robertson, S.E., Walker, S.: Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In: Proceedings of SIGIR '94, pp. 232–241 (1994)

[226] Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at trec-3. In: Proceedings of TREC-3 '95, pp. 109–126 (1995)

[227] Robinson, D., Coar, K.: The Common Gateway Interface (CGI) Version 1.1 RFC-3875 (2004)

[228] Rothenberg, J.: Ensuring the Longevity of Digital Documents. Scientific American **272**(1), 42–47 (1995)

[229] Rothenberg, J.: Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation (1999). `http://www.clir.org/PUBS/abstract/pub77.html`

[230] Salton, G. (ed.): Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley Longman Publishing Co. (1988)

[231] Salton, G., Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management **24**(5), 513–523 (1988). doi: http://dx.doi.org/10.1016/0306-4573(88)90021-0

[232] Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM **18**(11), 613–620 (1975). doi: http://doi.acm.org/10.1145/361219.361220

[233] Sanderson, R., Albritton, B., Schwemmer, R., Van de Sompel, H.: SharedCanvas: A Collaborative Model for Medieval Manuscript Layout Dissemination. In: Proceedings of JCDL '11 (2011)

[234] Sanderson, R., Phillips, M., Van de Sompel, H.: Analyzing the Persistence of Referenced Web Resources with Memento. In: Proceedings of OR '11 (2011)

[235] Sanderson, R., Van de Sompel, H.: Interoperable Annotation: Perspectives from the Open Annotation Collaboration (2009). `http://www.cni.org/tfms/2009b.fall/Abstracts/PB-interoperable-sanderson.html`

[236] Sanderson, R., Van de Sompel, H.: Making Web Annotations Persistent Over Time. In: Proceedings of JCDL '10, pp. 1–10 (2010)

[237] Saraiva, P.C., Silva de Moura, E., Ziviani, N., Meira, W., Fonseca, R., Riberio-Neto, B.: Rank-Preserving Two-Level Caching for Scalable Search Engines. In: Proceedings of SIGIR '01, pp. 51–58 (2001)

[238] Shafer, K., Weibel, S., Jul, E., Fausey, J.: Persistent uniform resource locators `http://www.purl.org/`

[239] Singhal, A., Buckley, C., Mitra, M.: Pivoted Document Length Normalization. In: Proceedings of SIGIR '96, pp. 21–29 (1996)

[240] Smith, J.A.: Integrating Preservation Functions Into the Web Server. Ph.D. thesis, Old Dominion University (2008)

[241] Smith, J.A., Klein, M., Nelson, M.L.: Repository Replication Using NNTP and SMTP. In: Proceedings of ECDL '06, pp. 51–62 (2006)

[242] Soboroff, I.: Do TREC Web Collections Look Like the Web? SIGIR Forum **36**(2), 23–31 (2002). doi: http://doi.acm.org/10.1145/792550.792554

[243] Spinellis, D.: The decay and failures of web references. Communications of the ACM **46**(1), 71–77 (2003). doi: http://doi.acm.org/10.1145/602421.602422

[244] Spink, A., Wolfram, D., Jansen, M.B.J., Saracevic, T.: Searching the Web: the Public and Their Queries. Journal of the American Society for Information Science **52**(3), 226–234 (2001)

[245] Staddon, J., Golle, P., Zimny, B.: Web based inference detection. In: USENIX Security Symposium (2007)

[246] Sugiyama, K., Hatano, K., Yoshikawa, M., Uemura, S.: A Method of Improving Feature Vector for Web Pages Reflecting the Contents of Their Out-Linked Pages. Database and Expert Systems Applications **2453**, 839–856 (2002)

[247] Sugiyama, K., Hatano, K., Yoshikawa, M., Uemura, S.: Refinement of TF-IDF Schemes for Web Pages using their Hyperlinked Neighboring Pages. In: Proceedings of HYPERTEXT '03, pp. 198–207 (2003)

[248] Sun, A., Hu, M., Lim, E.P.: Searching Blogs and News: A Study on Popular Queries. In: Proceedings of SIGIR '08, pp. 729–730 (2008)

[249] Sun, S., Lannom, L., Boesch, B.: Handle System Overview RFC-3650 (2003)

[250] Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining, 1 edn. Addison Wesley (2005)

[251] Teevan, J.: The Re:Search Engine: Simultaneous Support for Finding and Re-Finding. In: Proceedings of UIST '07, pp. 23–32 (2007)

[252] Teevan, J., Adar, E., Jones, R., Potts, M.: History Repeats Itself: Repeat Queries in Yahoo's Logs. In: Proceedings of SIGIR '06, pp. 703–704 (2006)

[253] Teevan, J., Adar, E., Jones, R., Potts, M.A.S.: Information Re-Retrieval: Repeat Queries in Yahoo's Logs. In: Proceedings of SIGIR '07, pp. 151–158 (2007)

[254] Teevan, J., Ramage, D., Morris, M.R.: Twittersearch: A comparison of microblog search and web search. In: Proceedings of WSDM '11, pp. 35–44 (2011)

[255] Turney, P.D.: Learning Algorithms for Keyphrase Extraction. Information Retrieval **2**(4), 303–336 (2000)

[256] Turney, P.D.: Coherent Keyphrase Extraction via Web Mining pp. 434–442 (2003)

[257] Van de Sompel, H., Lagoze, C.: Notes from the interoperability front: A progress report on the Open Archives Initiative. In: Proceedings of ECDL '02, pp. 144–157 (2002)

[258] Van de Sompel, H., Nelson, M.L., Sanderson, R.: HTTP Framework for Time-Based Access to Resource States Memento (November 2010). `http://datatracker.ietf.org/doc/draft-vandesompel-memento/`

[259] Van de Sompel, H., Nelson, M.L., Sanderson, R., Balakireva, L., Ainsworth, S., Shankar, H.: Memento: Time Travel for the Web. Tech. Rep. arXiv:0911.1112 (2009)

[260] Van de Sompel, H., Sanderson, R., Nelson, M.L., Balakireva, L., Shankar, H., Ainsworth, S.: An HTTP-Based Versioning Mechanism for Linked Data. Tech. Rep. arXiv:1003.3661v1 (2010)

[261] Vassileva, J.: A Task-Centered Approach for User Modeling in a Hypermedia Office Documentation System. User Modeling and User-Adapted Interaction **6**(2), 185–223 (1996)

[262] Vaughan, L., Thelwall, M.: Search Engine Coverage Bias: Evidence and Possible Causes. Information Processing and Management **40**(4), 693–707 (2004)

[263] Wan, X., Yang, J.: Wordrank-based Lexical Signatures for Finding Lost or Related Web Pages. In: APWeb, pp. 843–849 (2006)

[264] Ware, J., Klein, M., Nelson, M.L.: Rediscovering Missing Web Pages using Link Neighborhood Lexical Signatures. Tech. rep., CS Department, Old Dominion University, Norfolk, Virginia, USA (2011)

[265] Weber, I., Jaimes, A.: Who Uses Web Search for What: And How. In: Proceedings of WSDM '11, pp. 15–24 (2011)

[266] Wilbur, W.J., Sirotkin, K.: The Automatic Identification of Stop Words. Journal of Information Science **18**(1), 45–55 (1992). doi: http://dx.doi.org/10.1177/016555159201800106

[267] Winkler, W.E.: The State of Record Linkage and Current Research Problems. Tech. rep., Statistical Research Division, U.S. Census Bureau (1999)

[268] Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: KEA: Practical Automatic Keyphrase Extraction. In: Proceedings of DL '99, pp. 254–255 (1999)

[269] Wu, Y.f.B., Li, Q., Bot, R.S., Chen, X.: Domain-Specific Keyphrase Extraction. In: Proceedings of CIKM '05, pp. 283–284 (2005)

[270] Yanbe, Y., Jatowt, A., Nakamura, S., Tanaka, K.: Can Social Bookmarking Enhance Search in the Web? In: Proceedings of JCDL '07, pp. 107–116 (2007)

[271] Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of ICML '97, pp. 412–420 (1997)

[272] Yi, X., Allan, J.: A Content Based Approach for Discovering Missing Anchor Text for Web Search. In: Proceeding of SIGIR '10, pp. 427–434 (2010)

[273] Yih, W.t., Goodman, J., Carvalho, V.R.: Finding Advertising Keywords on Web Pages. In: Proceedings of WWW '06, pp. 213–222 (2006)

[274] Zaragoza, H., Cambazoglu, B.B., Baeza-Yates, R.: Web Search Solved?: All Result Rankings the Same? In: Proceedings of CIKM '10, pp. 529–538 (2010)

[275] Zhu, X., Rosenfeld, R.: Improving Trigram Language Modeling with the World Wide Web. In: Proceedings of ICASSP '01, pp. 533–536 (2001)

APPENDIX A

LIST OF STOP TITLES

about
athletics
default plesk page
default page
home
home page
homepage
hometown has been shutdown people connection blog aim community network
index
index of
index htm
index html
index page
indexhtm
indexhtml
intro
introduction
main
main page
msn groups closure notice
moved
new page
start page
the club
the conservatory index
the page cannot be displayed
the page cannot be found
this site is under construction
this web site coming soon
transferring
untitled
untitled document
untitled page
web page under construction
webage unavailable
website disabled
website errors

```
website not yet available
website template
welcome
```

APPENDIX B

URIS IN THE BOOK OF THE DEAD

http://presidentofindia.nic.in/scripts/prlatest1.jsp?id=377
http://www.ukrainianorthodoxchurchusa.org/index.shtml
http://www.zinos.com/cool/zinos/scan/se=AR004044/sp=view_article/rs=yes/go.html
http://www.zinos.com/cool/zinos/scan/se=AR005218/sp=view_article/rs=yes/go.html
http://ias.berkeley.edu/africa/Courses/Lectures/Darfur-Links.htm
http://independencia.net/2004/compromiso_rubengob.html
http://jimeyer.org/congress/
http://lambornforcongress.org/cms/
http://legalaffairs.org/howappealing/
http://lettforliberty.blogspot.com/
http://library.law.smu.edu/miers/index.htm
http://lyndallamas29th.blogspot.com/
http://members.shaw.ca/erwin-w/klimbim/
http://myinfoserver.com/findkenny/findkenny.htm
http://notapundit.myblogsite.com/blog/Judiciary/
http://ostinato.stanford.edu/coping/
http://phil4congress.blogspot.com/
http://purportal.com/special/9-11/
http://robertdenison.campaignoffice.com/
http://sonomagreenparty.org/pamelizondo.html
http://sos.state.nv.us/nvelection/
http://stuttmanforcongress.blogspot.com/
http://taosvacationguide.com/arts/fall_arts.php
http://teacher.scholastic.com/newszone/specialreports/911/
http://thestar.com.my/sympathy/sympathymessages.asp
http://toddtiahrt.com/index.htm
http://ur.rutgers.edu/medrel/viewArticle.phtml?ArticleID=1682
http://verusratio.blogspot.com/
http://voicemag.net/opinion/lavins.shtml
http://waynewhitmer.blogspot.com/
http://web.mac.com/johndriscoll37/montana_first/we_can_win%21.html
http://webguy.blogspot.com/
http://webofsilence.blogspot.com/
http://wfmb.com/wtc_tribute.swf
http://wtc.filsa.net/WTC.html
http://ww1.prweb.com/byindustry.php?prcatid=62
http://www.540wfla.com/shannonburke.html

http://www.adn.com/life/v-pda/story/1735610p-1851714c.html

http://www.airdisaster.com/cgi_bin/view_details.cgi?date=09112001&airline=American+Airlines

http://www.airdisaster.com/cgi_bin/view_details.cgi?date=09112001&airline=United+Airlines

http://www.allianceforarts.org/nyc-arts/9-11/perf.htm

http://www.alphadeltaphi.org/new/default.asp?PID=37

http://www.alternet.org/issues/index.html?IssueAreaID=25

http://www.americansolutions.com/default.aspx

http://www.amsa.org/news/supreme.cfm

http://www.apbroadcast.com/AP+Broadcast/Television/Video+and+Graphics/APTN+9-11-02.htm

http://www.artsusa.org/

http://www.asaforgovernor.org/default.aspx

http://www.atr.org/national/issueareas/judiciary/

http://www.bipac.org/home.asp

http://www.bna.com/tm/insights_Roberts.htm

http://www.brownback.com/s

http://www.cbs.com/primetime/9_11/

http://www.cdc.gov/od/oc/media/9-11pk.htm

http://www.charliebrownforcongress.org/issues.htm

http://www.chron.com/content/chronicle/special/01/terror/victims/beamer.html

http://www.cidcm.umd.edu/inscr/mar/assessment.asp?groupId=62504

http://www.constitutionpartymi.net/cpmi/Patriot/Dashairya/

http://www.constitutionpartypa.com/hagberg/

http://www.cox2008.com/cox

http://www.creativetime.org/towers/

http://www.crs.org/our_work/where_we_work/overseas/africa/sudan/

http://www.cs.umb.edu/~rwhealan/jfk/forum_darfur.html

http://www.csicop.org/hoaxwatch/

http://www.ctgop.org/home.shtml

http://www.cyberastro.com/articles/article12.asp

http://www.dccomics.com/features/911/911.html

http://www.de.lp.org/election2004/morris.html

http://www.deidreali.com/davis/index.html

http://www.dougdodd4congress.com/

http://www.dukenews.duke.edu/911site/

http://www.edmarkey.org/base.php

http://www.elections.state.ny.us/portal/page?_pageid=35,1,35_8617&_dad=portal&_schema=PORTAL

http://www.electtrawinski.org/home.html

http://www.emilyslist.org/happening/insider-news/

http://www.enziforwyoming.com
http://www.ezcampaigns.com/joewilliams
http://www.famm.org/index2.htm
http://www.fb.org/fbn/html/supreme.html
http://www.flt93memorial.org/cfms/index.cfm
http://www.freep.com/index/oneyearlater.htm
http://www.georgedweber.com/
http://www.georgetown.edu/crossroads/asainfo.html
http://www.grassleyworks.com/grassley
http://www.gwumc.edu/sphhs/imhi/
http://www.hazlitt.org/united/
http://www.herbconawayforcongress.campaignoffice.com/
http://www.iapn.org/cand%20janine.htm
http://www.independentamerican.org/candidates/christopherhansen.php
http://www.independentamerican.org/candidates/darnellroberts.php
http://www.independentamerican.org/candidates/davidschumann.php
http://www.independentamerican.org/candidates/joshuahansen.php
http://www.indymedia.org/peace/
http://www.inspiretowermemorial.org/index2.htm
http://www.jackdavis.org/new/
http://www.jademoran.com/6000.html
http://www.jesuit.org/JCOSIM/advocacy/index.html
http://www.jimryun.com/
http://www.joe2004.com/site/PageServer
http://www.jonlarsonforcongress.com/index.html
http://www.latinainstitute.org/judicial.html
http://www.law.com/jsp/statearchive.jsp?type=Article&oldid=ZZZ7UHW36TC
http://www.law.umich.edu/library/news/topics/miers/miersindex.htm
http://www.law.umich.edu/library/news/topics/roberts/robertsindex.htm
http://www.law.yale.edu/outside/html/Public_Affairs/689/yls_article.htm
http://www.lawyerscommittee.org/2005website/home/home.html
http://www.library.umass.edu/subject/supcourt/
http://www.life.com/Life/lifebooks/911/
http://www.life.com/Life/lifebooks/faces/
http://www.linnabary.us/homepage.php
http://www.lpcandidate.com/celesteadams/
http://www.lpct.org/Rasch/
http://www.lptexas.org/2006/acosta/
http://www.lptexas.org/2006/ashby/
http://www.lptexas.org/2006/fjones/
http://www.lptexas.org/2006/flynn/
http://www.lptexas.org/2006/haas/

http://www.lptexas.org/2006/hawley/

http://www.lptexas.org/2006/helm/

http://www.lptexas.org/2006/jimthompson/

http://www.lptexas.org/2006/jperez/

http://www.lptexas.org/2006/mclauchlan/

http://www.lptexas.org/2006/messina/

http://www.lptexas.org/2006/mnelson/

http://www.lptexas.org/2006/moyes/

http://www.lptexas.org/2006/nulsen/

http://www.lptexas.org/2006/osborne/

http://www.lptexas.org/2006/parks/

http://www.lptexas.org/2006/perkison/

http://www.lptexas.org/2006/powell/

http://www.lptexas.org/2006/strickland/

http://www.lynnforliberty.blogspot.com/

http://www.maryland.gov/portal/server.pt?space=communitypage&cached=true&parentname=communitypage&parentid=4&in_hi_userid=1333&control=setcommunity&communityid=227&pageid=202

http://www.mcc.org/sudanconflict/

http://www.mckissack.com/philadelphia/default.htm

http://www.mcny.org/9112001.htm

http://www.medair.org/en_portal/medair_programmes/darfur/index.php

http://www.metransparent.com/english

http://www.metropolismag.com/html/wtc/

http://www.mymanmitch.com/aboutmitch.asp

http://www.mysanantonio.com/expressnews/

http://www.nbm.org/Exhibits/past/2002/New_World_Trade_Center.html

http://www.newsday.com/news/local/newyork/ny-911anniversary.htmlstory

http://www.nfib.com/cgi-bin/NFIB.dll/Public/SiteNavigation/home.jsp

http://www.nj.gov/lps/elections/electionshome.html

http://www.november2.org/home.html

http://www.nypost.com/09112002/09112002.htm

http://www.nysfop.org/WTCdisaster/Fund.html

http://www.oneill08.com/homepage

http://www.orgsites.com/fl/markcoutuforuscongress/

http://www.orlandosentinel.com/news/custom/911/

http://www.padems.com/index800.html

http://www.pagop.org/index.asp

http://www.pathcom.com/~kat/blogs/jkmainblog.html

http://www.pbs.org/spotlighton/

http://www.philly.com/mld/inquirer/news/special_packages/sept11/

http://www.politicsnationwide.com/member/default.asp?id=48 http://www.pollingreport.com/Court.htm

http://www.publicagenda.org/headlines/headlines_blog_previous.cfm

http://www.publicagenda.org/issues/frontdoor.cfm?issue_type=abortion

http://www.randy2006.com/default.htm

http://www.reagan.utexas.edu/resource/findaid/robertsj.htm

http://www.renewnyc.com/index.shtml

http://www.reuters.com/news.jhtml?type=specialcoverage

http://www.ridemocrats.org/index.asp

http://www.rodforillinois.com/

http://www.rossello.com/rossello.php

http://www.rushholt.com/index.jsp

http://www.rvapc.com/ht/hthome.aspx

http://www.sacbee.com/content/news/projects/attacks/

http://www.saeamerica.org/terrorattack/

http://www.savedarfur.org/go.php?q=home2.php

http://www.schneider-for-congress.com/2004/

http://www.scim.vuw.ac.nz/comms/staff/Folly.htm

http://www.sffog.org/marktribute.html

http://www.shrinkiowagov.org/governor.html

http://www.sos.state.nm.us/Election/ElectionInfo.htm

http://www.sos.state.oh.us/sos/elections/

http://www.sots.ct.gov/ElectionsServices/ElectionIndex.html

http://www.spiritualityhealth.com/newsh/items/blank/item_3297.html

http://www.staircase.org/mug.html

http://www.state.nj.us/lps/elections/electionshome.html

http://www.stltoday.com/stltoday/news/special/91101.nsf/front?openview&count=2000

http://www.stmarys-island-church.co.uk/thedarfurchallenge2005.htm

http://www.stonewalldemocrats.org/supreme/

http://www.sudan.gov.sd/english.htm

http://www.supremecourtwatch.org/

http://www.tcoinc.com/hooper/

http://www.tenlinks.com/NEWS/special/wtc/clifton/p1.htm

http://www.theaapc.org/index.asp

http://www.thearc.org/governmental-affairs.htm

http://www.timmurphyforcongress.com/about.htm

http://www.tnr.com/doc.mhtml?i=20020909&s=filler090902

http://www.tnr.com/scw.mhtml

http://www.tomdelay.com/home/

http://www.tomfeeney.com/cf/letter.html

http://www.unsudanig.org/emergencies/darfur/

http://www.usamemorial.org/sept11017.htm

http://www.usdoj.gov/ag/terrorismaftermath.html

http://www.usdoj.gov/olp/roberts.htm
http://www.usstudents.org/main.asp
http://www.usstudents.org/p.asp?webpage_id=107
http://www.violentserenity.net/blog/index.html
http://www.viviculture.org/weblog/index.html
http:
//www.vividence.com/public/news+and+events/press+releases/in+memory+of+jeremy.htm
http://www.voiceyourself.com/commentary.html
http://www.webcom.com/ncecd/
http://www.webcom.com/peaceact/
http://www.wetdreamspoetry.com/emptysky.html
http://www.wibc.com/911_2002/index.jhtml
http://www.williamluse.blogspot.com/
http://www.wisinfo.com/dailytribune/index.shtml
http://www.wohlmut.com/kevin/Ego/Friends/MarkBingham.htm
http://www.womenplayrugby.blogspot.com/
http://www.womenwarpeace.org/sudan/sudan.htm
http://www.workingfamiliesparty.org/fusion.html
http://www.world-of-wisdom.com/articles/articletwintowers.htm
http://www.worldtrade.com/Leonard_Pitts/body_leonard_pitts.htm
http://www.wsws.org/sections/category/news/us-2006van.shtml
http://www.wsws.org/sections/category/news/us-2006whit.shtml
http://www.wtcrelief.info/Charities/Information/pages/Home.jsp
http://www.wyay.com/pages/70767.asp
http://www.xentrik.net/stoneangel/blog/blog.html
http://www.yakima-herald.com/cgi-bin/liveique.acgi$rec=39778?home
http://www.yankiwi.com/yankiwi/
http://www.ycsi.net/users/reversespins/shoebomb.html
http://www.yorkjaycees.com:80/September_11_2001.htm
http://www.youwillneverbeforgotten.com/YWNBF/
http://www.zaldor.com/blog/
http://www.zweknu.org/blog/
http://www.zyworld.com/brancatelli/branc.htm
http://youth.haguepeace.org/hapyouth/
http://zentropolis.net/home.cfm

VITA

Martin Klein
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

EDUCATION

Ph.D.	Computer Science, Old Dominion University, 2011
Diploma	Applied Computer Science, University of Applied Sciences, Berlin, Germany, 2002

PROFESSIONAL EXPERIENCE

2010–2010	Instructor of Computer Science, Old Dominion University
2005–2011	Research Assistant, Old Dominion University
2002–2005	Research Assistant, University of Applied Sciences, Berlin, Germany
2001–2001	Software Engineer, Peito GmbH, Berlin, Germany
2000–2001	Software Engineer Intern, Yahoo!, Munich, Germany
1997–1999	System Administration Intern, German Heart Center, Berlin, Germany

PUBLICATIONS AND PRESENTATIONS

A complete list is available at `http://www.cs.odu.edu/~mklein/pubs.html`

PROFESSIONAL SOCIETIES

Association for Computing Machinery (ACM)

Typeset using LaTeX.