

IDENTIFYING THE EFFECTS OF UNEXPECTED CHANGE IN A DISTRIBUTED
COLLECTION OF WEB DOCUMENTS

A Dissertation

by

LUIS DAVID MENESES MACCHIAVELLO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Richard Furuta
Committee Members,	Frank Shipman
	James Caverlee
	Laura Mandell
Head of Department,	Dilma Da Silva

August 2016

Major Subject: Computer Science

Copyright 2016 Luis Meneses

ABSTRACT

It is not unusual for digital collections to degrade and suffer from problems associated with unexpected change. In previous analyses, I have found that categorizing the degree of change affecting a digital collection over time is a difficult task. More specifically, I found that categorizing this degree of change is not a binary problem where documents are either unchanged or they have changed so dramatically that they do not fit within the scope of the collection. It is, in part, a characterization of the intent of the change. In this dissertation, I present a study that compares change detection methods based on machine learning algorithms against the assessment made by human subjects in a user study. Consequently, this dissertation focuses on two research questions. First, how can we categorize the various degrees of change that documents can endure? This point becomes increasingly interesting if we take into account that the resources found in a digital library are often curated and maintained by experts with affiliations to professionally managed institutions. And second, how do the automatic detection methods fare against the human assessment of change in the ACM conference list?

The results of this dissertation are threefold. First, I provide a categorization framework that highlights the different instances of change that I found in an analysis of the Association for Computing Machinery conference list. Second, I focus on a set of procedures to classify the documents according to the characteristics of change that they

exhibit. Finally, I evaluate the classification procedures against the assessment of human subjects. Taking into account the results of the user evaluation and the inability of the test subjects to recognize some instances of change, the main conclusion that I derive from my dissertation is that managing the effects of unexpected change is a more serious problem than had previously been anticipated, thus requiring the immediate attention of collection managers and curators.

DEDICATION

To my mother and father, my family and Primo.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Richard Furuta, for all his help, input and support that led to the completion of this dissertation. I have learned so much from him; without his guidance and encouragement, I wouldn't have made it this far in the pursue of my PhD. More so, Dr. Furuta has been a role model for me when exercising his passion for knowledge while having the courage to try new things. Besides his obvious academic merits and skills, he always showed consideration and flexibility towards the needs of his students. I would like to thank him sincerely for all his support throughout the years of my PhD and my Master's degree.

I would also like to recognize the help and support from my committee. I would like to thank Dr. Frank Shipman for the discussions that helped clarify some of the issues that I faced while writing this dissertation. I am also grateful for the opportunity I was given to work on the Ensemble Project. The work in this project provided me with the stage I needed to apply the methodologies I developed. Special thanks go towards Dr. James Caverlee, for being always open to share his insights and experience; and to Dr. Laura Mandell, for supporting me to attend conferences and helping keep my research grounded in the humanities. I would also like to thank Dr. Enrique Mallen for giving me the opportunity to collaborate with him in the Online Picasso Project, which ultimately became the starting point in my academic journey.

I would like to thank all the graduate students from the Center for the Study of Digital Libraries. In particular, I would like to thank Carlos Monroy for mentoring me

not only with respect to academia, but also in views of life. I would like to thank Sampath Jayarathna for his knowledge and suggestions in designing the document classification algorithms.

I must also specially thank Dr. Tiffani Williams. Dr. Williams was my mentor when I was part a Graduate Teaching Fellow. She was always very objective and gave me her honest perspective on things, which was something I really needed during the time I was finishing writing this dissertation.

Finally, I would like to thank my family and friends. Their unconditional support is what made possible for me to complete this lengthy journey.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iv
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
LIST OF TABLES	xi
CHAPTER I INTRODUCTION	1
CHAPTER II PROBLEM	6
CHAPTER III BACKGROUND	10
3.1 Web Infrastructure	10
3.2 Search Engines	11
3.3 Information Retrieval Measures	12
3.4 Lexical Signatures	15
3.5 Distributed Collection Manager	15
3.6 Finding Missing Resources	17
3.7 Link Persistence	21
3.8 Identifying Soft 404 Errors	22
CHAPTER IV IDENTIFYING SOFT 404 ERRORS	24
4.1 Initial Assessment	29
4.2 Experiment Parameters	30
4.3 Identifying Soft 404 Error Pages	31
4.4 Discussion and Implications	35
CHAPTER V RESTORING INCOMPLETE COLLECTIONS	39
5.1 Experiment Setup	44
5.2 Results	46
5.3 Discussion and Implications	51

CHAPTER VI CATEGORIZING CHANGE IN AN INSTITUTIONALLY MANAGED REPOSITORY	53
6.1 A Distributed Collection within an Institutional Digital Library	53
6.2 Categorization of the Types of Change	56
6.3 Classification Features	62
6.4 Discussion and Implications.....	67
CHAPTER VII RESULTS.....	69
7.1 Classification Algorithms.....	73
7.2 Features Analysis	76
7.3 Classifying Documents using Principal Components	78
7.4 Discussion and Implications.....	80
CHAPTER VIII EVALUATION.....	82
8.1 Experiment Setup	82
8.2 Results	85
8.3 Discussion and Implications.....	87
CHAPTER IX CONCLUSIONS	90
REFERENCES.....	97

LIST OF FIGURES

	Page
Figure 1: Screenshot of http://www.jcdl2011.org . Accessed on 9/27/2014.....	4
Figure 2: Categorization of the documents in distributed collections.....	7
Figure 3: System diagram for the architecture of the DCM.....	17
Figure 4: Default 404 error page - http://www.jcdl.org/archived-conf-sites/jcdl2001/foo . Accessed on 6/1/2015.	25
Figure 5: Customized 404 error page - http://atariage.com/foo . Accessed on 6/1/2015..	25
Figure 6: Highly customized 404 error page - http://www.microsoft.com/en-ca/foo . Accessed on 6/1/2015.....	26
Figure 7: Soft 404 error page from http://badoo.com . Accessed on 4/3/2012.....	27
Figure 8: HTTP headers from a 404 error page. Taken from http://www.atariage.com/foo . Accessed on 6/2/2015.....	28
Figure 9: HTTP headers from a Soft 404 error page. Taken from http://www.symantec.com/errors/notfound.jsp . Accessed on 6/2/2015	28
Figure 10: Precision and recall measures per iteration for Soft 404 identification using complete HTML documents.....	33
Figure 11: Precision and recall measures per iteration for Soft 404 identification using the extracted title metadata.	34
Figure 12: Average precision and recall values for Soft 404 identification using complete HTML documents and title metadata.....	34
Figure 13: Example of a “Soft 404” error viewed through the Walden’s Paths user interface. This missing document makes the collection semantically incomplete.....	41
Figure 14: Average cosine similarity between a missing document and the other valid documents within the path.	47
Figure 15: Average resemblance grouped by the number of nodes in each analyzed path.	48

Figure 16: Average cosine similarity grouped by the number of nodes in each analyzed path.	49
Figure 17: Average quadratic mean between the original and modified similarity matrices grouped by the number of nodes on each path.....	50
Figure 18: Distribution of the pages that were retrieved with a 200 (OK) HTTP response code.....	56
Figure 19: Screenshot of the ASPLOS 98 site showing an example of a “directory listing page”. Accessed at http://arch.cs.ucdavis.edu/ASPLOS98/ on 9/27/2014.	58
Figure 20: Screenshot of the DGO 2010 site showing an example of a “domain for sale page”. Accessed at http://www.dgo2010.org on 9/27/2014.	59
Figure 21: Screenshot of the IDC 2004 site showing an example of a “deceiving page”. Accessed at http://www.idc2004.org on 9/27/2014.	60
Figure 22: Screenshot of a “deceiving page” where the layout is related to the original content, but the links are unrelated. Accessed at http://www.cikm.org on 9/27/2014.....	61
Figure 23: Distribution of the incorrect pages.	62
Figure 24: Accumulated page lifespan histogram.....	70
Figure 25: Binary classification of the “clearly correct” category with the “incorrect” and “unsure” categories combined to “not-correct”.	71
Figure 26: Classification of the “not-correct” category combining the “incorrect” and “unsure” categories.....	72
Figure 27: Screenshot of a sample questionnaire regarding a document in the user study.....	85

LIST OF TABLES

	Page
Table 1: Frequency distribution of the retrieved pages according to their HTTP response codes.	55
Table 2: Binary classification using link and content based features.	74
Table 3: Classification of only the “incorrect” categories.	75
Table 4: Link-based and content-based features performance comparison.	75
Table 5: Classification of only the “incorrect” categories by removing the “pages in a different language” category.....	75
Table 6: Classification of only the “incorrect” categories by removing the “pages in a different language” and “unsure” categories.	75
Table 7: Principal components for the features in the binary classification.	77
Table 8: Principal components for the features in the classification of error pages.	78
Table 9: Binary classification using the features from the principal components analysis.	79
Table 10: Error page classification using principal components.	79
Table 11: Error page classification for the “deceiving” and “kind of correct” categories.	80
Table 12: Gender, age distribution and educational level of the population.....	83
Table 13: User responses for the document classification in the user study by categories. Shaded: Correct – Not shaded: Incorrect.....	86
Table 14: Reasons for the documents that were correctly identified.	86
Table 15: Reasons for the documents that were incorrectly identified.	87
Table 16: Comparison between the classification made by human subjects and the classification algorithms.	87

CHAPTER I

INTRODUCTION

Bush's Memex and its associative trails offered a vision of how digital collections could take form [1]. *As We May Think* has become an iconic essay that introduced many concepts that were ahead of its time such as hypertext systems and associative trails. Because of its novelty, it is understandable that this essay failed to foresee some of the challenges that digital collections will bring with time.

Curating a digital collection is not an easy task. Selecting, organizing and presenting the documents in a collection are laborious tasks that require significant resources from a curator. Moreover, and contrary to popular belief, a curator's efforts do not cease once the documents have been ingested into a particular collection: a curator must also look after the documents to ensure that the collection remains consistent over time.

To make matters worse, there is a specific type of digital collection where looking after the consistency in its documents has a more crucial role. These collections are known as *distributed*, which means the administrative control of information related to a topic may be spread across other digital collections maintained by multiple scholars in multiple institutions. This administrative decentralization leads to changes that are unexpected by the maintainer of a collection, or in this case, a "meta-resource"— a resource created by tying together the existing resources.

These unexpected changes can be caused by different factors or circumstances. Changes can occur because of deliberate actions on part of the collector – for example, reorganization of the structure of the collection, switching to a different content management system, or changing jobs and institutions. Changes might also be due to unexpected events – earthquakes, power outages, disk failures, – or may be due to other uncontrollable factors – death, seizure of computers by law enforcement, or termination of the services from an Internet Service Provider [2].

Over time, great strides have been made to characterize and manage change within collections of documents. Klein and Nelson argued that digital documents do not disappear from the Web, but leave artifacts that can be used to reconstruct them [3]. Bar-Yosseff et al. carried out experiments to measure the decay of the Web [4]. SalahEldeen determined that nearly 11% of shared resources will be lost one year after being published and that this decay will continue at a 0.02% rate per day [5]. Nevertheless, and despite these previous efforts, managing and characterizing change in a collection is inherently difficult. I will elaborate on the reasons behind this difficulty with three points.

First, documents from the Web are not static resources and a certain degree of change is expected from them [6]. For example – and taking into account the specific infrastructure of Walden's Paths [7] where decentralized collections are stored and represented as traversable paths containing multiple nodes and documents – Web resources suffer from changes in content, layout, presentation and location. However, as members of a larger assembly, these documents are expected to either change little over

time or mutate harmoniously and accordingly with the other documents in order to preserve the semantic meaning and systematic order of the collection.

Second, distributed collections that are hosted in institutional repositories operate under the assumption that they are more resilient and able to withstand change. Because of their focus on long-term storage, these repositories have different attributes and operate under different principles when compared to the Web as a whole. By definition, a digital repository must include procedures and tools for curating, organizing, storing, and retrieving the documents and media contained in the collection. Surprisingly, I have found that these features – which are often found in digital repositories emphasizing in long-term storage – do not foster the use of procedures to actively curate the metadata and preserve the referenced documents to make them more impervious to change. For example, the ACM Digital Library has 15 unique links referencing the different sites in the JCDL conference series and 8 of them report errors or point to the wrong content. Figure 1 shows a screenshot of <http://www.jcdl2011.org> – which now displays information about diets and weight loss.

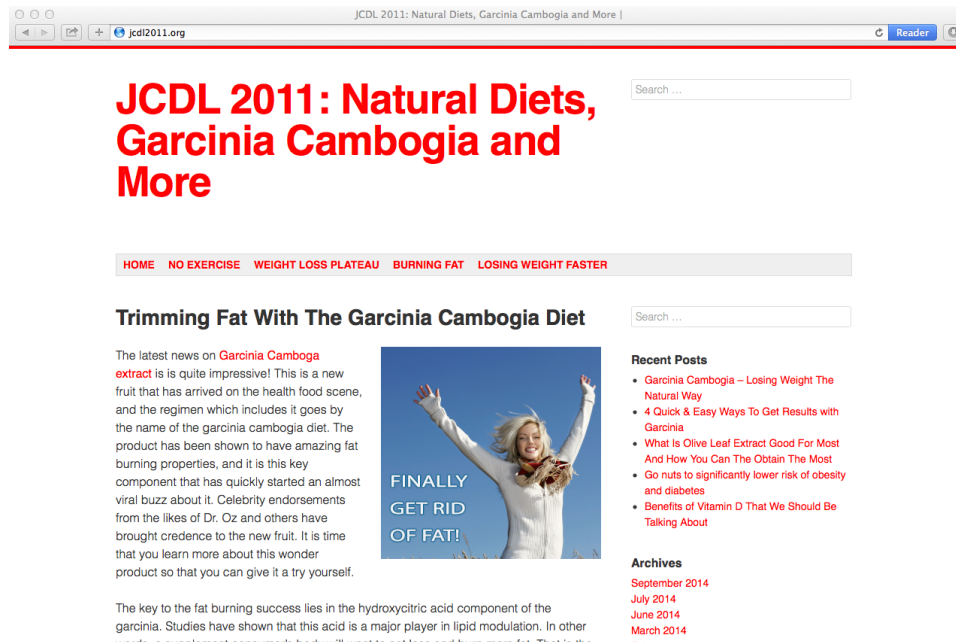


Figure 1: Screenshot of <http://www.jcdl2011.org>. Accessed on 9/27/2014.

Finally, categorizing the degree of change affecting the documents referenced in a distributed digital collection over time is a complex task. More specifically, I found that categorizing this degree of change is not a binary problem where documents are either unchanged or they have changed so dramatically that they do not fit within the scope of the collection. Previous work on this topic has relied on methods based on response codes, monitoring the fluctuation in file sizes and analyzing the documents' content. However, previous approaches do not take into account the fluid nature of the Web [8] and that some degree of change is normal and expected in documents over time.

Therefore, and because of its complexity, a single solution that can be applied to mitigate and manage the effects of unexpected change in a distributed collection does not exist. Nonetheless, previous work has shown that some measures and procedures are more successful than others when applied to different types of resources. Thus,

mitigating the effects of unexpected change can be viewed as a multiple step process where first, the characteristics and the types of resources in the collections are identified; and second, the appropriate measures and procedures are applied.

Therefore, the purpose of the investigation that I will describe in this dissertation is to focus the attention of curators on potentially problematic resources. In this dissertation I will describe a procedure that can be used to detect documents that have been affected by unexpected change in distributed digital collections. I will also address several of the implications that this procedure introduces. More specifically, I will address two research questions in this dissertation. First, which attributes and methods can be used to effectively detect unexpected changes in a distributed collection? This point becomes increasingly interesting when taking into account that the resources found in distributed digital repositories are often curated and maintained by experts with affiliations to professional institutions. And second, how do these detection methods fare against the assessment of change made by human subjects? I will address these research questions in the following chapters of this dissertation.

CHAPTER II

PROBLEM

At the present time, the preservation and curation of digital collections falls under the responsibility of archival institutions, which translates into archives and libraries investing a considerable amount of effort into maintaining the consistency of their collections. Nowadays, three preservation approaches are commonly used: refreshing, migration and emulation. The first two are somewhat similar: refreshing deals with replicating files in different system, whereas migration is concerned with moving documents to a newer and more current environment. On the other hand, emulation is concerned with replicating the functionality of obsolete hardware and systems while utilizing modern equipment [9, 10].

Refreshing, migration and emulation are approaches that follow a “just in case” philosophy – preserving data just in case it is needed in the future. On the other hand, the World Wide Web [11] provides its own preservation methods. “Just in time” preservation manifests itself in the Web when users create, copy and move content [12]. Because of its passive nature and characteristics, which stem from being a by-product of search engines and web archives, “just in time” preservation is disorganized and lacks any centralized quality control. Nevertheless, this passive preservation approach allows digital documents in the Web to be preserved in some form because of the information habits and the ubiquitous nature of users of the Web.

Assigning different degrees of urgency to mitigate the impact of unexpected changes can be done by determining the parts of a collection that require immediate attention from a curator. However, determining which documents in a collection require immediate attention is a complex task, and this complexity stems from the different characteristics of the change that can affect the different resources. For instance, the documents in distributed collections can be categorized into four tiers or *meta* groups which are shown in figure 2. In turn, each of these meta groups reveals a certain degree of divergence from the original state of each document and its metadata when they were originally ingested into in a collection.

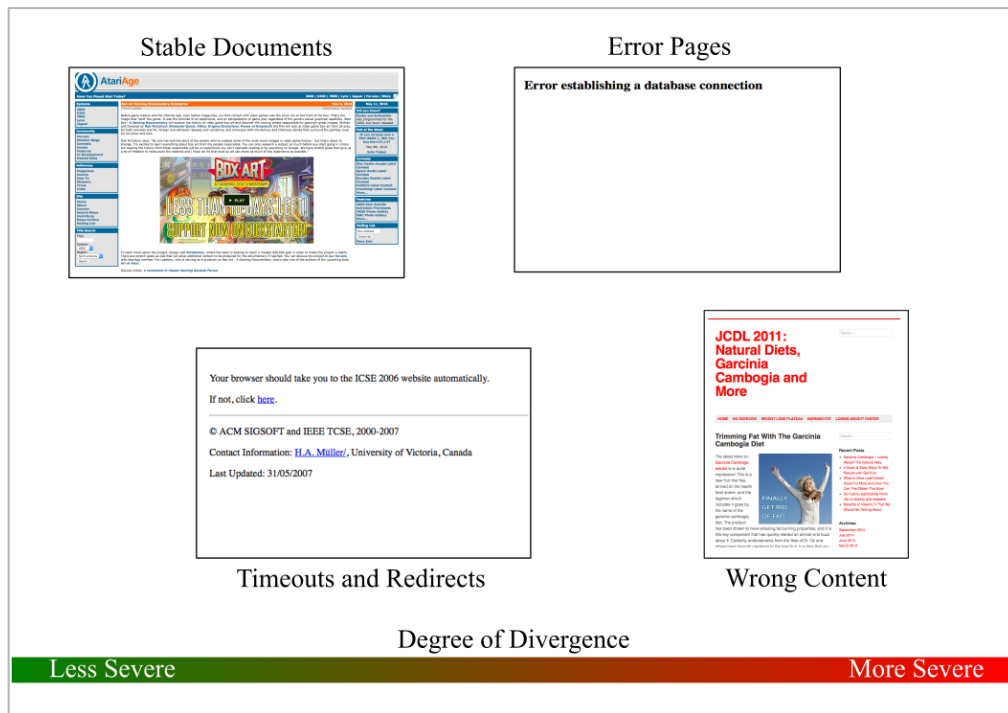


Figure 2: Categorization of the documents in distributed collections.

The first tier deals with stable documents. This tier constitutes an ideal scenario in many cases, as the documents and metadata reflect the original state when they were ingested into the collection. However, a document that remains unchanged for a significant period of time can indirectly suggest that it has been abandoned and becoming a problematic resource.

The second, third and fourth tiers contain instances when documents suggest the existence of errors; and some documents indicate errors more explicitly than others. The second tier deals with server timeouts and failed redirects, while the third tier deals with error pages. These cases require special attention as they can be caused by uncontrollable factors and circumstances – such as misconfiguration of a Web server, hard disk failures and power outages – or by negligence from a Webmaster by letting the domain name registration lapse. In some cases, error pages can be easily identified with the first digit of the status code from the server’s HTTP response. However, some Web servers do not report errors explicitly by masking them with a different status code, thus introducing the notion of a *Soft 404 error* page. Soft 404 error pages are difficult to identify using automated methods.

Finally, the fourth tier deals with cases when a document has been affected by change to some degree and has diverged from its initial state. In this tier the contents of a document haven’t changed triggering explicit errors from their Web server, but the nature of their change affects the semantic integrity of the collection as a whole. Examples of documents in this tier include: abandoned pages and sites that change their

topic, hyperlinks that point to the wrong content, and sites that have been taken over to manipulate search engine results.

The research problem of detecting the effects of unexpected changes in a digital document collection has been previously addressed [6, 13]. However, my dissertation has a very defined scope – which is focused on distributed collections. Additionally, the procedure that I will describe in the following chapters has some key differences that set it apart from previous efforts. For example, Bogen [6] focused on change in the Web as part of the natural life cycle of a document and determined the difference between what a maintainer expects a page to do from what it actually does using Kalman filters; whereas Francisco-Revilla et al. [13] focused on monitoring the structural changes and presentation characteristics of a document. Despite these key differences, I use these previous studies and their conclusions as inspiration and as a partial foundation for my work.

This dissertation is organized as follows: chapter III provides an overview of previous work in related areas. Chapter IV deals with the background needed for identifying unexpected change in the form of Soft 404 errors. Chapter V describes the importance of methods for dealing with change in documents by restoring the semantic integrity of collections that have been affected by unexpected change. Chapter VI describes a categorization system for documents that have been affected by unexpected change in their content. Chapter VII describes the results I obtained; while chapter VIII provides an overview of the evaluation methods that I used. Finally, chapter IX is dedicated towards discussion, conclusions and ideas for future work.

CHAPTER III

BACKGROUND

This chapter is divided into three sections. First, I will present several concepts that are used throughout this dissertation. I will elaborate on the details that form the underlying infrastructure of the Web and how it is tied to my motivation for studying unexpected change in the distributed digital collections. Second, I will describe the methods that are used to evaluate the performance of information retrieval algorithms in the Web. I will use these methods in the following chapters of this dissertation. Understanding them will ultimately help the reader get a better understanding of my work. Finally, I will describe the previous work that lays the foundation for this dissertation. The previous work that I will describe focuses in three areas: finding missing resources, link persistence and identifying Soft 404 error pages.

3.1 Web Infrastructure

Part of the underlying infrastructure of the Web that is relevant to my dissertation is made up of search engines (e.g., Google [14], Yahoo! [15], Bing [16]), non profit archives (the Internet Archive [17] and the European Archive [18]), and large-scale academic digital preservation projects (such as CiteSeer [19] and the National Science Digital Library [20]). The underlying infrastructure of the web is discussed in detail by Jatowt et al. in [21].

McCown et al. investigated the factors that contribute towards the reconstruction of websites [22]. Their work indicated that the PageRank [23] and the age of the missing page are the two factors that can influence the outcome of the reconstruction. Additionally, McCown and Nelson formally defined *flat* and *deep* repositories [24]. According to their framework, a repository is flat if it only stores the last version of a document. On the other hand, deep repositories store multiple versions of a given document – in which different versions of a resource are distinguished by their timestamp.

3.2 Search Engines

A Web search engine is computer software designed to search information on the Internet. Search results, which consist of a mixture of web pages, images and surrogates for other types of documents, are generally presented as numbered sets and results ranked higher indicate a greater relevance towards the search terms. Search engines gather and aggregate pages into their search indices using an automated web crawler.

All the major search engines offer APIs that are open to the public to access their services – sometimes for a fee. These APIs allow users and researchers to write programming scripts that can retrieve results circumventing the need to access a traditional Web interface. One of the main advantages of accessing search engines through their API is that results can be retrieved systematically and at a faster rate. Users can also specify the range of the desired results and their granularity: for example retrieving results from a specific website and that sit above or below a ranking threshold.

Search engines and their APIs are very important in the scheme of research related to the Web, thus making them the subjects of performance evaluations. Previous studies have found discrepancies between the results obtained from Web user interfaces and APIs [25]. More specifically, the results obtained from querying directly the APIs seemed to point to older Web indices. At the time that this study took place (2007), the search results that were retrieved from Microsoft Network (MSN – renamed to Bing) had the greatest overall similarity between its API and Web interfaces when compared to other Google and Yahoo! [25]. Additionally, the time it takes for top search results is not standardized across search engines. Top results for popular queries can remain in place for up to 18 months in Google and Yahoo! [25]. However, refreshing results took 2 to 3 months in MSN.

3.3 Information Retrieval Measures

Term frequency (known as TF) expresses the number of occurrences of a particular term in a given document and thus represents the importance of a term. On the other hand, inverse document frequency (known as IDF) expresses the number of documents that contain a given term in a document corpus [26]. Therefore, IDF indicates the rareness of a term. Consequently, a TF-IDF score provides a measure of the importance of a term within a specific document and a corpus. Equations 1 and 2 show the formulas for calculating TF and IDF respectively.

$$TF(t, d) = \frac{|t \text{ appears in } d|}{|Terms \text{ in } d|} \quad (1)$$

$$IDF(t, D) = \log \left(\frac{|D|}{|Documents\ in\ D\ that\ contain\ t|} \right) \quad (2)$$

Precision and *recall* are ubiquitous measures to quantify the retrieval performance of a system [27, 28]. Precision is a measure of the fraction of retrieved documents that are relevant to a query, whereas recall is a measure of the fraction of the documents that are relevant to the query that are successfully retrieved. This distinction between the two measures implies that there is a trade-off between them. Systems with high precision often exhibit low recall; on the other hand, it is possible to obtain a high recall by simply returning all documents on any query. Equations 3 and 4 show that precision is a measure of how good the system is at rejecting documents that are not relevant, and recall measures the performance of a system when returning all the relevant documents.

$$Precision = \frac{|Relevant\ documents \cap Retrieved\ documents|}{|Retrieved\ documents|} \quad (3)$$

$$Recall = \frac{|Relevant\ documents \cap Retrieved\ documents|}{|Relevant\ documents|} \quad (4)$$

F-measure is a measure that helps reveal the accuracy of a system. More specifically, F-measure is the harmonic mean between precision and recall values [28, 29]. One of the advantages of using F-measure to measure the performance of a system is that it conveys the performance of a system in a single value. However, it has the

drawback of hiding some of the nuances that can be expressed using precision and recall.

Equation 5 shows how the traditional F-measure is calculated.

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (5)$$

Cosine similarity is one of the most widely used similarity measures. Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them in N-dimensional space [30]. In general, a vector represents each document where the value of each dimension corresponds to the number of times that a given term appears in the document. Given the representations for documents A and B, the similarity of these two documents is defined as:

$$sim(A, B) = \frac{\vec{v}(A) \cdot \vec{v}(B)}{\|\vec{v}(A)\| \|\vec{v}(B)\|} \quad (6)$$

where the numerator of equation 4 is the dot product of the vector representations of the documents and the denominator is the product of their Euclidean lengths.

Shingling is a technique for identifying and detecting near-duplicates of web pages [31, 32]. Given a sequence of terms in a document D, the N-shingles of D is a set of all the consecutive contiguous subsequences of N terms that can be extracted from D. Therefore, for a given shingle size, equation 7 shows the degree to which two documents A and B resemble each other as a measure of their resemblance.

$$resemblance(A, B) = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|} \quad (7)$$

3.4 Lexical Signatures

A lexical signature of a document is a set of terms that expresses (to a certain degree) the subject and “aboutness” of a document. In other words, the lexical signature of a document is a representation of the most significant terms in the textual content. Lexical signatures are generated following an arrangement where the terms of the documents are ranked in decreasing order according to their TF-IDF weights. The top N terms from this arrangement are the resulting lexical signature of the document. Stop words, which occur frequently in a language but do not contribute towards the semantic meaning of a document [33, 34], are filtered before the TF-IDF calculation. Additionally, removing the stop words in a document has the advantage of reducing the complexity of computing the lexical signatures.

3.5 Distributed Collection Manager

The original version of the Distributed Collection Manager (DCM) was created as a successor to the Walden’s Paths Path Manager [35]. The first implementation of the DCM was created to observe the fluidity of web pages and collections while taking into account that the web as a whole was also changing. Although it was useful in detecting instances of abnormal change, the first implementation of the DCM had the shortcoming that a single method of detection could only be used. The current implementation of the DCM, which I will refer to as version 2, addresses this shortcoming by employing a system architecture where pluggable components can be used for testing different detection approaches.

Like its predecessor, the current version of the DCM can be broadly defined as an ecosystem of computational tools that share a common repository and database. More specifically, the current DCM is composed of four layers that communicate between each other using JSON [36]. The first layer is the user interface, which is the point of interaction between the users and the system. The user interface layer also includes functionality for the management of user accounts and their authentication. The second layer is the document harvester. The document harvester is a threaded service that wakes up periodically and retrieves pages at a given interval of time. To avoid errors and instances associated with retrieving pages at the same time periodicity, this interval is set to 20 hours – which also means that some documents will be retrieved twice in a given day. The third layer is the storage layer, which stores the documents in flat file system and logs all the interactions between the layers a MySQL database. Additionally, the storage layer is capable of storing multiple versions of the retrieved documents by creating a random hash of the retrieved URL, which is then used to store each resource. Finally, the fourth layer consists of the components and procedures used to detect unexpected change. Two of these components – that are used to identify Soft 404-error pages and restore incomplete collections – will be described with more detail in chapters IV and V. Figure 3 shows a system diagram for the architecture of the DCM.

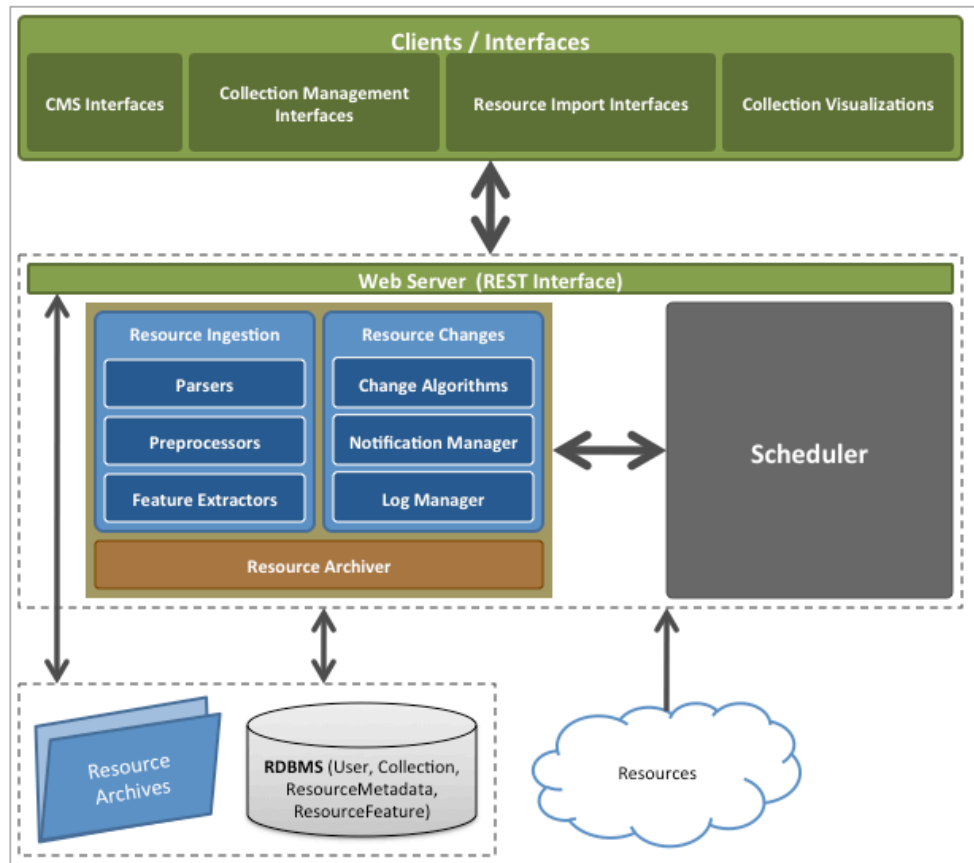


Figure 3: System diagram for the architecture of the DCM.

3.6 Finding Missing Resources

Previous work on finding missing resources is based around the premise that documents and information are not lost but simply misplaced [37] as a consequence of the lack of integrity in the Web [38, 39]. Reconstruction tools can be used to recover a website when it is lost and backups are not available [40]. Other studies have also focused on analyzing content and link-based methods to rediscover missing pages [41], finding the longevity of documents in the Web [42] and also in distributed collections [43, 44].

Douglis et al. studied the rate of change in a corpus of Web pages [45]. Their research found that HTML documents changed at a faster rate when compared to more static resources – such as images or similar resources. Their research also showed that the most frequently accessed resources undergo frequent modifications. Given that embedded resources are archived at different intervals from the rest of the linked resources, the presentation of archived Web document might appear coherent. Previous studies have shown that only 1 out of 5 documents reconstructed from archives are temporally coherent and complete – meaning that their presentation matches how they looked when they were ingested into the archive [46]. Archival crawlers can also produce results that differ from what the user expects, especially when dealing with interactive features of HTML and Javascript [47].

Characterizing the amount of change on the Web has been studied previously. Cho and Garcia-Molina studied how sites change depending on their domain suffix [48]. They found that it took approximately 11 days for 50% of the .com domains in their sample to change. However, .gov sites were updated less frequently – taking 4 months for 50% of the sites in this domain to exhibit some form of change. As a conclusion of their study, Cho and Garcia-Molina also stated change can be expected in a Web page every 4 months approximately. In a later study, Cho and Garcia-Molina introduced a model to estimate the frequency of change in Web documents [49]. In terms of size and scale, Fetterly et al. conducted one of the largest studies of focused on Web page change. They crawled 150 million pages once a week for 11 weeks and carried out a study that compared the changes across pages while focusing on features such as page size [50].

When comparing sizes in bytes, Fetterly et al. found that larger pages change more often than smaller ones.

Ntoulas, Cho, and Olston studied how Web pages can change using weekly snapshots of 150 websites that were collected over a year [51]. They found that most pages did not change according to traditional measures of similarity (most notably bag-of-words). They also found that how frequently the documents changed was not a good predictor for their degree of change, but the degree of change was a good predictor of future degrees of change. Additionally, Adar et al. found that the amount of change in a Web document is dependent of the domain and structure of the page [52]. However, they found that 65% of the pages in their corpus exhibited some form of change after a period of five weeks. Olston and Pandey crawled 10,000 random Web pages and 10,000 pages sampled from the Open Directory every two days for several months [53]. Their analyses measured both change frequency and information longevity. As a result of this study, new crawl policies that are sensitive to information longevity were developed.

Phelps and Wilensky pioneered the use of lexical signatures to locate missing content in the Web [54]. They claimed that if a Web request returned a 404 error, querying a search engine with five terms describing the document could retrieve the missing content. However, it is known that search engines are dynamic entities and their results can change over time [55]. Park et al. used Phelps and Wilensky's previous research to perform an evaluation of nine lexical signature generators that incorporated term frequency measures [56]. Additionally, Klein and Nelson have extracted lexical signatures from titles and backlinks to find missing Web resources [3].

Finding replacements complements the problem of finding missing resources. Furthermore, finding replacements shares some similarities with identifying near-duplicate documents. Near-duplicate documents are identical in terms of content, but have differences in their formatting, minor corrections, advertisements, logos and timestamps. Finding near-duplicate documents is difficult because single web resources usually convey many semantic components. Despite the inherent difficulty associated with this problem, the Shingling [31] and Simhash [57] algorithms are considered the state of the art in near-duplicate document detection. The similarity measure used in the Shingling algorithm was previously described in chapter III.

Another approach used to identify near-duplicates is based on creating a digital fingerprint for a document. A digital fingerprint is a stream of bits that uniquely identifies the original contents of a document. Early proponents of this approach include Manber [58]; along with Shivakumar and Garcia-Molina [59]. Along the same lines, Brin, Davis and Garcia-Molina also proposed a system that detected documents that overlapped in “significant ways” [60].

Forman et al. identified near-similar technical support documents [61]. Interestingly, they chose to rely only on the contents of the documents because of missing and corrupted metadata. Their approach was defined by the document corpus, as support documents can contain illustrations and diagrams and cannot be broken into semantic sections. Instead they chose to break the documents in a consistent way that was not dictated by semantics and then detect collisions between near-duplicates.

Although detecting near-duplicates shares some similarities with my research, my methods have a different scope. Near-duplicates are documents that share the same content but have different presentation characteristics. The purpose of my research is finding documents that can serve as replacements for missing items to restore the integrity and continuity of a path. Moreover, finding near-duplicates of missing documents would require a cached or archived copy of the lost document – which for practical purposes I am assuming that I don't have – and a very specific crawl of the web that is also beyond the scope of my approach.

More specifically, my previous research aimed to enhance and complement the methods used by Dalal et al. to find appropriate replacements for missing resources from the web that belonged to a collection in the Walden's Paths Project [62]. Their approach was based on a two-step process. First, metadata was extracted when the path was created thus preserving the author's intent and vision. Second, the extracted metadata was used to find pages when they cannot be retrieved. In the specific case of collections such as Walden's Paths, each node in a path is destined to make a contribution towards the overall concept and the continuity in the narration. Therefore, finding replacements becomes a critical factor to maintain the integrity of the collections and preserve their semantic meaning.

3.7 Link Persistence

On the other hand, previous work on link persistence has focused on characterizing the availability of resources over time. Nelson and Allen measured the

persistence and availability of documents in a digital library [63]. Based on the premise that objects placed in a digital library should persist longer than an average Web Page, they analyzed 50 random objects from twenty digital repositories and found that 3% of these objects were no longer available. Koehler found that specialized document collections – such as legal, educational and some scientific citations – tend to stabilize over time [64]. However, citations in some domains have higher rates of failure [65]. McCown et al. also explored other factors that might cause a resource to fail by examining its age, path depth, top-level domain and file extension [66]. Similar studies have also focused on distributed science education resources, where 16.5% of the URLs in the collection have ceased to function or had their content changed over a period of 14 months [67]. Obviously, maintaining the contents of documents in the Web has been identified as an urgent problem. Knowledge-based approaches have been employed to properly verify the contents of Web sites using semantic markup to formulate rules and constraints that must hold according to the information contained in a document [68]. Furthermore, my research attempts to extend previous work on this area as a whole by examining a distributed collection as part of an institutional digital library. Additionally, I will describe the types of issues found in the collection, and examine the potential to automatically identify these issues when they surface.

3.8 Identifying Soft 404 Errors

Some websites are known to mask a "not found" error by returning a standard web page with a "200 OK" response code. This type of response is known as a Soft 404

error page. Soft 404s present a problem for automated detection methods because they hinder the nature and purpose of a document. As is expected, the problem of identifying Soft 404 pages has been attempted before. Bar-Yossef et al. created an algorithm that analyzed the behavior of web servers to predict the occurrence of Soft 404 error pages [4]. Their approach was based on creating a request for a document that does not exist on server, thus triggering a 404 error. As webservers usually share similar rules for each hosted site, Soft 404s can be predicted by extending the response patterns to a larger set of documents within the same host.

My approach (which I will describe in chapter IV) differs from previous efforts in its scope and application: I use the lexical signatures of Web documents to identify Soft 404 errors. Additionally, previous research has relied heavily on HTTP status codes (mainly 404s) and server responses to identify error pages. However, these status codes do not always represent the nature and content of the retrieved document. Consequently, my work presents an approach to identifying errors not reported by status codes.

CHAPTER IV

IDENTIFYING SOFT 404 ERRORS¹

I base my methods and procedures for collection management on the results of two studies, which are described in chapters IV and V. The first study explores methods for identifying patterns of abnormal change in Web documents in the form of Soft 404 errors, while the second examines methods for restoring collections where its documents have been altered due to unexpected changes. These studies expanded my understanding of unexpected change in the Web and provided the foundation for the methods for categorizing change.

A 404 or “Not Found” error in the Web hints that the requested content is not available in the given server. These errors are very common on the Web and have become part of the user browsing experience. Because of this and as an effort to make the Web more tractable, 404 error pages have changed over time. Figures 4, 5 and 6 illustrate three common examples of 404 error pages: Figure 4 shows a default error page that states that the requested URL and its corresponding Web page could not be found in the file structure of the server. Next, figure 5 is more customized and polished in an effort to appear more pleasant than a standard error page (It shows the logo of the site and the text is centered and highly formatted). Finally, figure 6 displays a highly

¹ Part of this chapter is reprinted with permission from "Identifying “Soft 404” Error Pages: Analyzing the Lexical Signatures of Documents in Distributed Collections" by L. Meneses, R. Furuta, and F. M. Shipman, 2012, in *Proceedings of Theory and Practice of Digital Libraries 2012*. Copyright 2012 by Springer Berlin Heidelberg.

customized error page, where the user is provided suggestions based upon what he was probably looking for and links to resume browsing the site (if he chooses to do so).

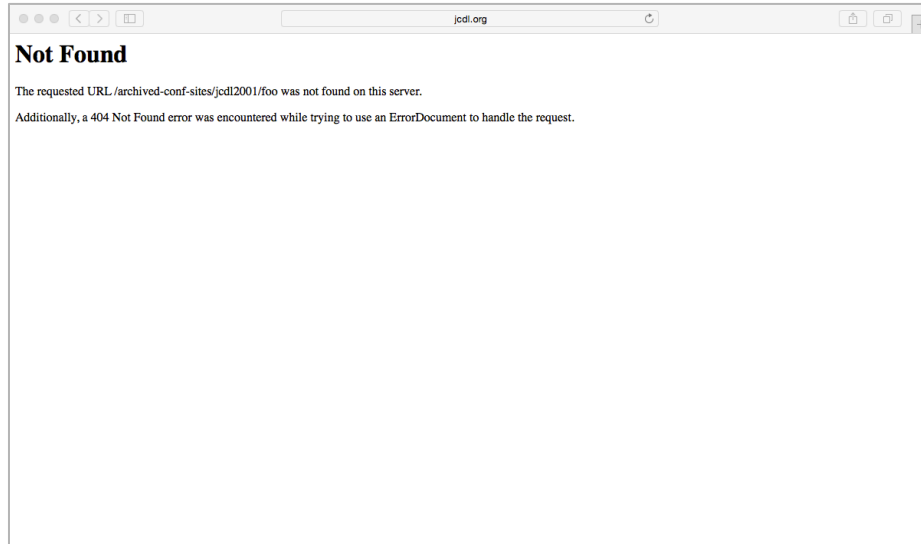


Figure 4: Default 404 error page - <http://www.jcdl.org/archived-conf-sites/jcdl2001/foo>. Accessed on 6/1/2015.

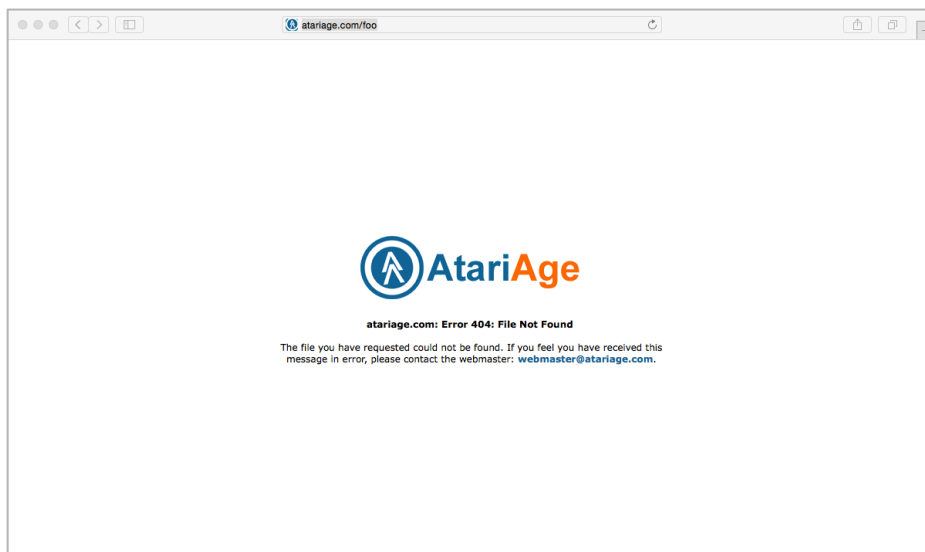


Figure 5: Customized 404 error page - <http://atariage.com/foo>. Accessed on 6/1/2015.

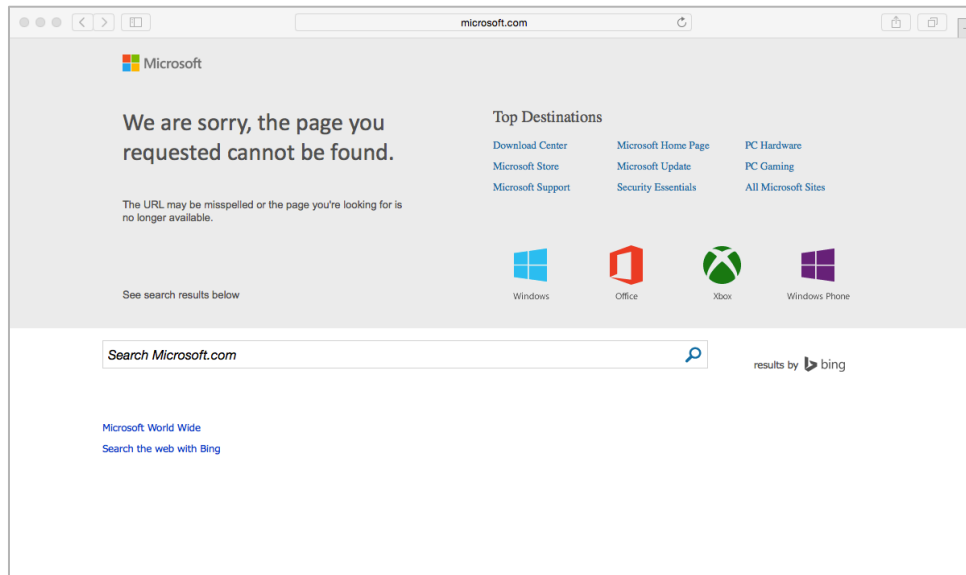


Figure 6: Highly customized 404 error page - <http://www.microsoft.com/en-ca/foo>. Accessed on 6/1/2015.

The three examples that I have shown are easily identifiable by analyzing their response headers and their HTTP response codes: in particular because they all return a 404 HTTP response code. On the other hand, there are more sophisticated and problematic cases in which a Web server reports change in a human-readable, but not machine-readable format. These sophisticated and problematic cases can occur intentionally (for example, attempting to provide a “friendly” or “soft” way to allow the reader to continue by redirecting failed pages to the site’s index page), without knowledge of the original site (for example, the landing page provided by some ISPs when an address is given that refers to a host that is no longer available), or deceptively (for example, the registration of lapsed domain names and uploading a different Web site as an attempt to sell the domain name back to its original holder). Collectively, these error pages are called “friendly” or “soft” 404s as they mask the return code of 404

normally returned when there is a failure to access a Web resource. Soft 404s also hinder the task of monitoring the changes in the structure and content of a Web document. Figure 7 shows an example of a Soft 404 taken from <http://badoo.com>, while figure 8 and 9 show the differences in the HTTP headers for a 404 and Soft 404 error page.

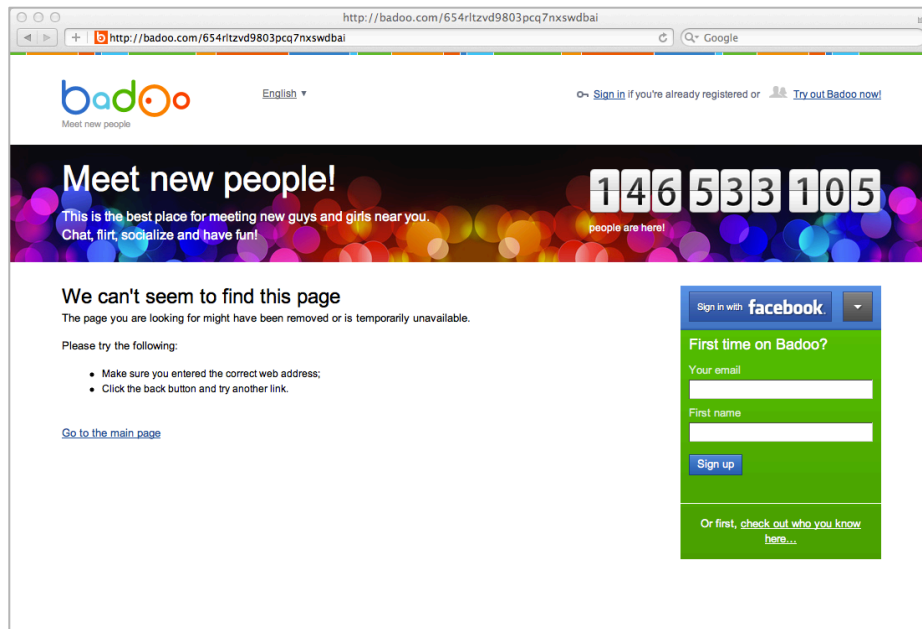
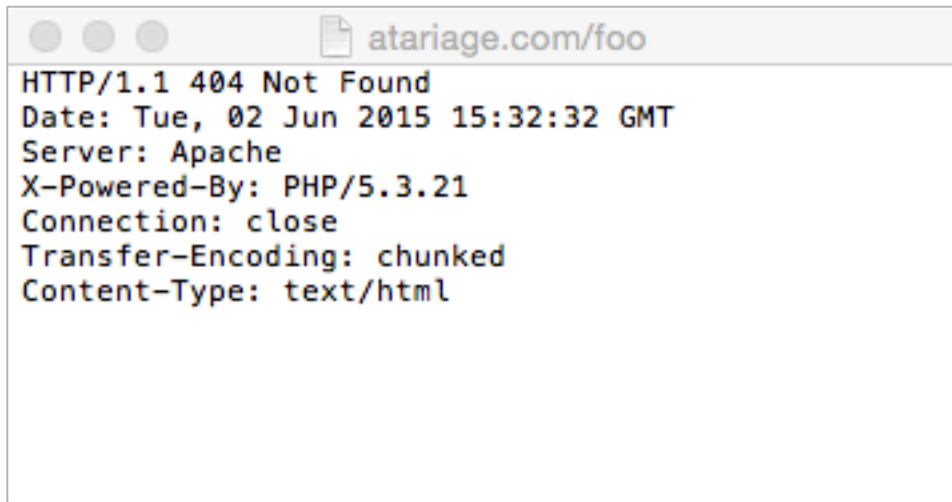


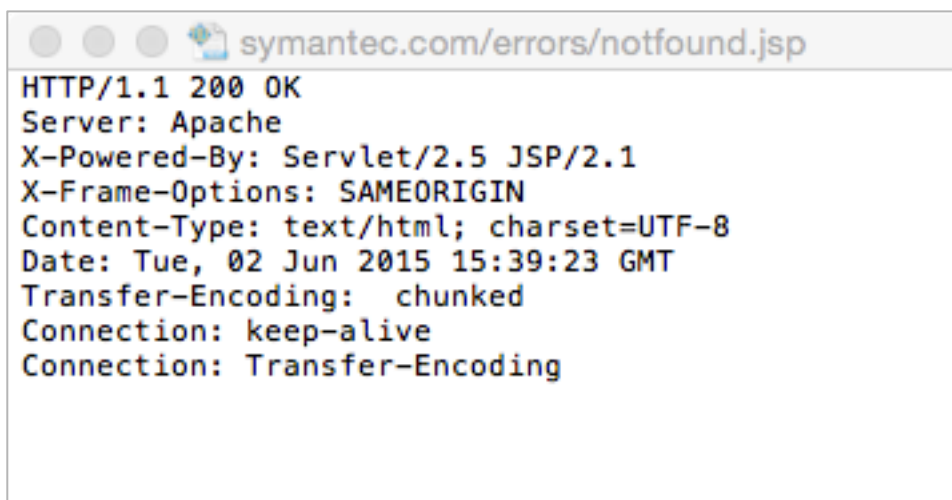
Figure 7: Soft 404 error page from <http://badoo.com>. Accessed on 4/3/2012.

As a general rule, Soft 404s errors return pages that the user is not expecting - making the difference between the expected and the returned document very significant. Consequently, search engines have identified Soft 404 errors as undesirable results and consider them not useful to the user. Google has professed a farewell to 404s [69] and announced that their crawling mechanisms are able to detect Soft 404 and notify network administrators about their occurrence [70]. As desirable as this procedure might seem, their results have been inaccurate and known to include server side errors [71].



```
atariage.com/foo
HTTP/1.1 404 Not Found
Date: Tue, 02 Jun 2015 15:32:32 GMT
Server: Apache
X-Powered-By: PHP/5.3.21
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html
```

Figure 8: HTTP headers from a 404 error page. Taken from <http://www.atariage.com/foo>. Accessed on 6/2/2015



```
symantec.com/errors/notfound.jsp
HTTP/1.1 200 OK
Server: Apache
X-Powered-By: Servlet/2.5 JSP/2.1
X-Frame-Options: SAMEORIGIN
Content-Type: text/html; charset=UTF-8
Date: Tue, 02 Jun 2015 15:39:23 GMT
Transfer-Encoding: chunked
Connection: keep-alive
Connection: Transfer-Encoding
```

Figure 9: HTTP headers from a Soft 404 error page. Taken from <http://www.symantec.com/errors/notfound.jsp>. Accessed on 6/2/2015

Thus, detecting Soft 404 pages is important to maintain semantic integrity among the documents in a distributed collection. More specifically, Soft 404 pages hinder the

task of monitoring the changes in content of a Web resource by masking its purpose behind an HTTP response code that does not correspond with its contents.

4.1 Initial Assessment

I started the first study by analyzing the commonness of Soft 404 errors in two collections of documents: resources that return an OK HTTP response (with a 200 code) and those returning Soft 404 pages. For this purpose, I obtained a dataset for pages with OK responses by analyzing a random subset of 166,103 websites taken from the Open Directory Project database (<https://www.dmoz.org>). Out of this sample, 147685 URLs returned OK responses when queried, which accounts for 88.91% of the sample. Only 206 returned a 404 error, which accounts for 0.12% of the sample. However, 404 errors in the web are very common. For instance, 475 out of 1000 sites in the sample were already “dead” (returning a soft or hard 404) when Bar-Yossef et al. ran their first experiment to measure the decay in the Web [4]. Thus, at this point I hypothesized that the low number of 404 errors in the sample served as an indicator of the prevalence of Soft 404 errors in the Web.

Then I tested the validity and correctness of the sample for the OK responses, which consisted of 147685 URLs. For this purpose, I forced a 404 error by appending a random sequence of 25 characters to each URL (i.e.: <http://www.youtube.com/8q07a24ildpy3sbjmwngl0vs>) and analyzed the response codes from the servers. 5,017 requests still returned an OK response, which accounts for the 3.41% of the subsample. 131,529 URLs returned a 404 error code, which accounts for

89.35% of the subsample. After conducting this cross-reference for validation purposes, I computed that Soft 404 errors could be found when accessing 3.41% of the documents in the web.

4.2 Experiment Parameters

Following this initial assessment, I used a different sample for the next phase in my experiment, which included the development and testing of a classification system. I used a snapshot of Alexa's daily 1,000,000 Top Sites taken on 3/22/2012. This dataset is freely available from Alexa.com and it is provided in CSV format. I used the sites listed in Alexa's classification because of the increased certainty that the list would contain valid and up to date Web addresses.

Coincidentally, my study has some degree of similarity in the parameters used by Bar-Yossef et al [4]. I will summarize the parameters used in four points. First, I attempted to fetch the "absolute URL" for each given website. It is a common practice nowadays for websites to use redirects depending on specific content, location, language or other factors. For example: nokia.com actually redirects to <http://www.nokia.com/us-en/>. For this study I only allowed up to 10 redirects for each URL.

Second, to produce a Soft 404 response from an actual working URL, I forced each Web server to return an error page by concatenating a random combination of 25 letters (a to z) and numbers (0 to 9) to the requested entry. The probability of a page actually existing in the server to fulfill the request is trivial: $N/(36^R)$ where N is the number of documents in the actual file path and $R=25$. To ensure that the server would

return a Soft 404, the random sequence of characters was appended to an absolute URL and file path. This ensured that the retrieved document would be a Soft 404 instead of a “hard” 404 error page.

Third, the content of the Web pages was retrieved using Python’s HTTP protocol client with a timeout value of 10 seconds. The HTML cleanup, key term extraction and analysis was carried out with Python’s Natural Language Toolkit (NLTK) [72] and BeautifulSoup libraries [73].

Finally, using the analysis tools from the NLTK package, I found that the average Soft 404-error page in the resulting collection contains 173 words; 24 of which are stop words. As expected, the lexical signatures of the Soft 404s provided indications of error responses: page, not, found, 404, requested, sorry, error. In the case of Soft 404 error pages, these error terms synthesize the subject and purpose of the documents. With these characteristics, I created methods to identify Soft 404 pages in collections of documents from the Web.

4.3 Identifying Soft 404 Error Pages

For this experiment, I applied the characteristics extracted from Soft 404 pages towards building a text classifier to analyze the contents of Web documents. I used a Naïve Bayes classifier that looked for specific lexical signatures in the content of the pages. I ran the text classifier against a dataset of 1000 random URLs from Alexa’s daily 1,000,000 Top Sites snapshot for 10 iterations. Additionally, I used the classifier against two corpuses of data: complete HTML documents and metadata extracted from the title

tags. Each run was completely independent from each other and the data for the experiment was extracted at run time.

In the case of the HTML documents, I took into account the commonness of finding the lexical signatures and the overall term length of the documents. In the case of the title metadata, I only used the lexical signatures because the number of terms was trivial. On the other hand, I concentrated on the titles embedded in the contents of Web documents based on the premise that titles synthesize the contents of the documents [74].

To evaluate my approach, the classifier used different sets of Web documents for training and testing. The training set was constituted with 100 documents: 50 Soft 404 pages and 50 normal (not soft 404) responses. The documents in the training set were selected randomly from the top-sites corpus obtained from Alexa.com. Additionally, I implemented features for the classifier to minimize false positives: pages with OK responses that were predicted as 404 errors. For these specific cases, I ran the Naïve-Bayes classifier for a second time using a different Web address constructed randomly with the same URL to compare the document titles and their corresponding predictions. Consequently, this second pass of the classifier increased the precision of the predictions of the classifier by approximately 10%.

For the first document corpus, dealing with full HTML documents, the text classifier was able to predict on average Soft 404s with a precision of 95% and a recall of 87%. For the second corpus, which focused on the title metadata, the classifier was able to predict on average Soft 404 pages with a precision of 99% and a recall of 92%.

Using the titles as surrogates for the documents allowed the text classifier to make more reliable and accurate predictions. Figures 10 and 11 illustrate the individual precision and recall values obtained identifying Soft 404s on each of the ten iterations. Figure 12 shows the average values for Soft 404 identification using complete HTML documents and the title metadata.

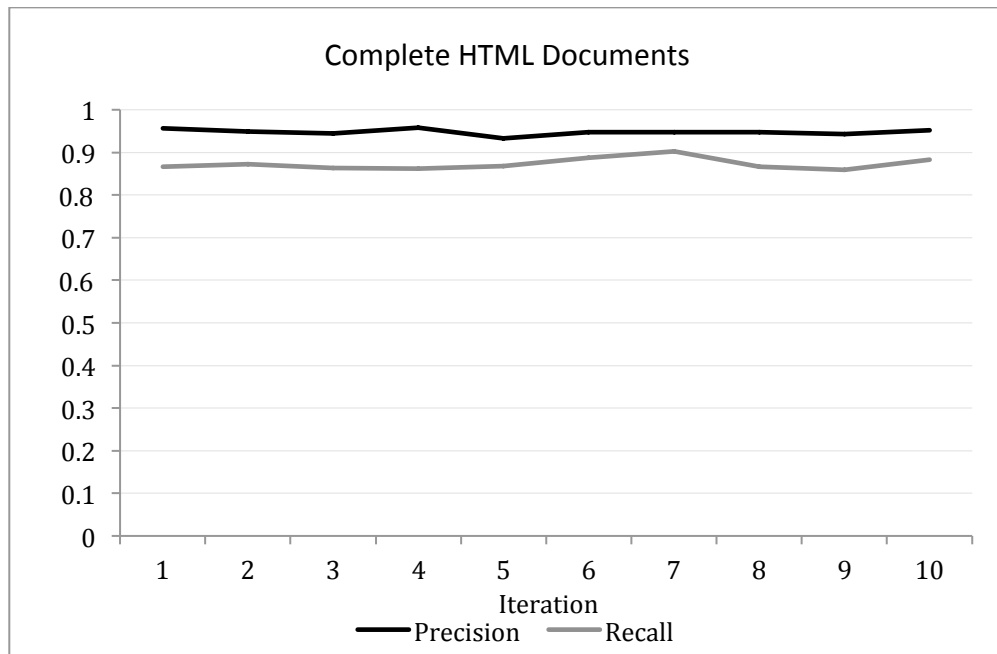


Figure 10: Precision and recall measures per iteration for Soft 404 identification using complete HTML documents.

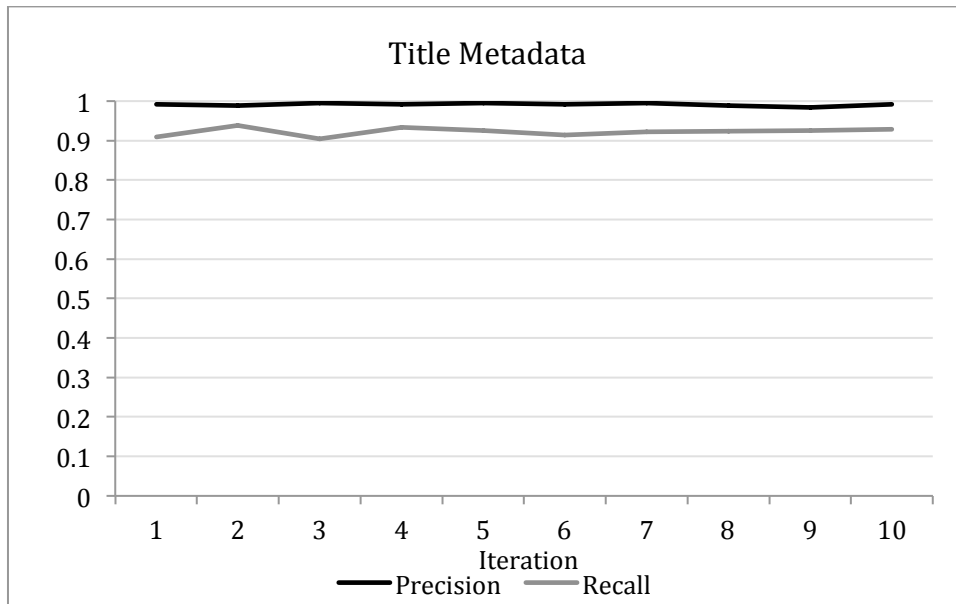


Figure 11: Precision and recall measures per iteration for Soft 404 identification using the extracted title metadata.

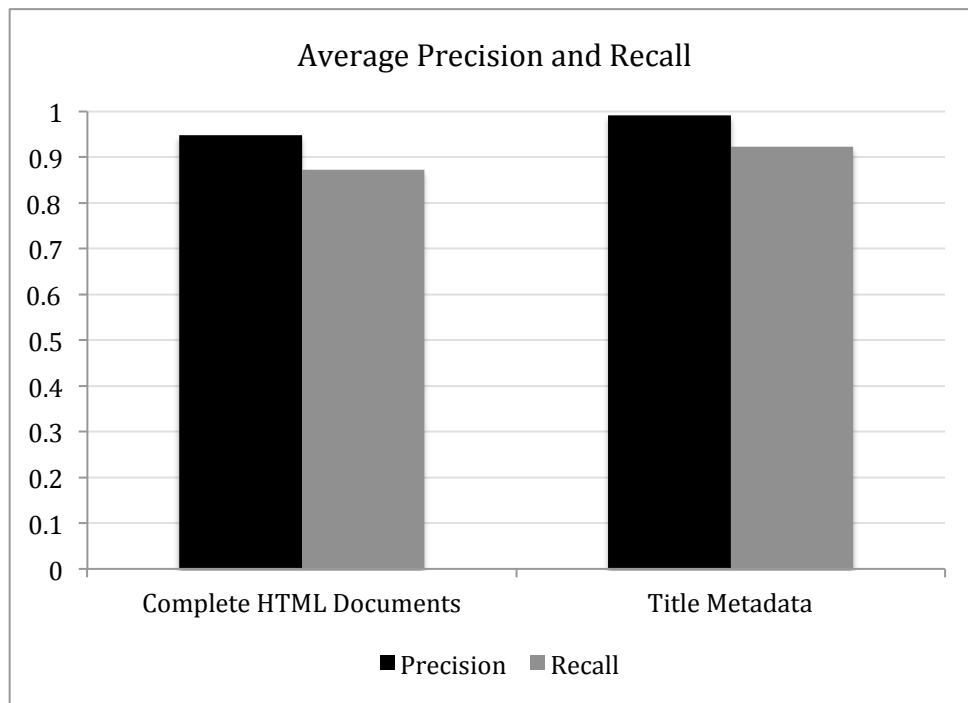


Figure 12: Average precision and recall values for Soft 404 identification using complete HTML documents and title metadata.

4.4 Discussion and Implications

Identifying Soft 404 responses is not trivial. My initial methods, which precede the results outlined in this dissertation chapter, were based on an approach that allowed a text classifier to extract the most important features. This approach dealt with generating a large term concordance where weights and probabilities were assigned to each occurrence. This approach returned varied and inconsistent measures of precision and recall. There are two reasons for these results. First, the variability in the sample: Web pages vary greatly in content and layout across the Web. Second, the combined probabilities of occurrence from the terms in a Web page introduced noise into the classifier. More so, a similar approach would work better in a specialized domain where a concordance built from the textual content of the documents has a smaller number of key terms.

A central obstacle for this research was obtaining valid and reliable datasets. Nowadays, Web pages change constantly and their lifespan is limited. As expected, text classifiers are only as good as the training data. If the data fed into the training set of the classifier contains errors, false positives or false negatives, the classifier will not be accurate in its predictions. The continuous change in content and availability of documents in the Web created difficulties when I attempted to evaluate alternate approaches for the extraction of patterns and features. Thus, expanding the corpus of Soft 404s allows identifying additional unique features of Soft 404 pages and will improve the overall accuracy of the classifier.

The classifier was able to achieve a high level of precision with complete HTML documents because of the combination of two features: the analysis of lexical signatures and the overall term count of the documents. When each feature is isolated, the classifier achieved lower levels of precision when detecting Soft 404 pages. However, I found the lexical signatures to be a better classifying feature than the overall term count. Factoring only the lexical signatures returned a precision value of 82% with the corpus of HTML documents. I obtained higher precision values when analyzing the lexical signatures from the title metadata.

Despite obtaining good levels of precision and recall, analyzing large numbers of terms in the complete HTML document corpus introduced noise into the classifier. Consequently, these fluctuations had the potential to make the classifier produce erroneous results. For the second approach, which synthesized the titles of documents, the text classifier was less prone to noise by analyzing only document surrogates. The boost in performance was a consequence of the synthesized lexical signatures in the extracted data.

However, I found that the classifying procedure by itself favors 404 predictions. This perceived bias towards the predictions increased the number of false positives. In these specific cases, I addressed the problem by running the classifier again which ultimately increased the precision measure. Alternatively, I could have implemented additional classification features into the algorithm, which in turn could have further increased the complexity of the experiments and analysis. Additionally, by performing a

second pass on individual documents with the text classifier, I addressed the problematic case where two different documents from the same site could share the same title.

I also found that page redirects in sites caused inaccurate predictions as false negatives. This issue was prevalent with sites whose content was different for different countries or audiences. For example, a request to “http://skype.com” originated within the United States will redirect the user to “http://www.skype.com/intl/en-us/home”. I ultimately addressed this issue by fetching and “absolute” URL and path. However, the number of redirects caused inaccurate predictions and ultimately affected the overall recall value of the algorithm in the early stages of the study.

Although somewhat similar, the two procedures I have described can have different application areas. Corpora of Web documents can benefit from using lexical signatures and overall term count, which yields a more thorough analysis but has a slightly lower precision and recall. Scenarios where only the title metadata is available can expect higher precision and recall values under the assumption that the title metadata contextualizes the contents from Web documents. Specific cases where higher precision and recall values are crucial should use the analysis of lexical signatures synthesized from the title metadata.

Given that identifying Soft 404s was previously attempted in 2004 by Bar-Yossef et al [4], two questions still need to be answered. First, How do the methodologies compare? And second, which one is better? The two methodologies have great differences. The method presented by Bar-Yossef et al. relied on server response codes to “learn” how websites react to different requests. On the other hand, my approach

analyzes the contents of Web documents and relies on lexical signatures. I purposely used similar parameters and procedures in the experiments to facilitate comparisons and analysis. The similarities include: number of iterations (10), sample size per iteration (1000 URLs), retrieval timeout (10 seconds). Also, the algorithms for fetching the relevant data from each URL share some principles. For example: how to determine the absolute URL, how to circumvent redirects, and how to force a Soft 404 error.

Ultimately, the two methods (mine and Bar-Yossef's) have different applications. Decentralized collections where documents are stored and cached can benefit from the approach using lexical signatures, while environments with network bandwidth limitations can utilize the approach from Bar-Yossef et al. that relies on response server codes.

Answering which methodology is better is complicated because of two reasons. First, Bar-Yossef et al. do not provide precision and recall measures; and second, there are differences in the samples that were used in the experiments. Additionally, it must be taken into account their method for identifying Soft 404s was not intended as an end result as they were ultimately attempting to measure the decay in the Web. However, my approach presents some advantages given that it analyzes the lexical signatures of documents retrieved from the Web.

CHAPTER V

RESTORING INCOMPLETE COLLECTIONS²

Despite the previous efforts that I have described in the background chapter of this dissertation, managing missing resources in a digital collection can still be problematic. This is due mostly to the dynamic nature of the Web, which allows documents and Web pages to change and disappear without notice. Taking into account the infrastructure of Walden's Paths as an example, decentralized collections can be stored as traversable paths containing multiple nodes and documents. Adopting this architecture metaphor does not spare decentralized collection and its documents from changing unexpectedly. The degree of change that these collections endure is not homogeneous: in some cases documents might exhibit Soft 404 errors, while in others change is manifested in variations in content and in presentation. As I have discussed in previous chapters, changes in content can be minimal or have a huge impact in the semantic meaning in the collection. Thus, when documents change unexpectedly I can foresee two possible scenarios occurring when resources cannot be found. First, we have access to a copy of the missing document or to its lexical signatures – which are defined as a set of key identifying terms. Previous research by Dalal et al. has already addressed this case [62], which makes finding the missing resource a trivial scenario. The second

² Part of this chapter is reprinted with permission from " Restoring Semantically Incomplete Document Collections Using Lexical Signatures" by L. Meneses, H. Barthwal, S. Singh, R. Furuta, and F. M. Shipman, 2013, in *Proceedings of Theory and Practice of Digital Libraries 2013*. Copyright 2013 by Springer Berlin Heidelberg.

scenario is more interesting to me at this point: What happens if we don't have any metadata associated to the missing resource? A missing document is of grave importance, because each resource contributes to the overall meaning of the collection and to the continuity of the narration. Therefore, a missing document can potentially interrupt the flow of the collection and make it semantically incomplete. Figure 13 shows an example of a "Soft 404" error, where a server masks a "404: not found" HTTP response with a standard page and a different response code, which in turn causes a collection in Walden's Paths to become semantically incomplete. Given that the URL of the resource is part of the metadata describing the node, gathering the documents that point to the missing resource could be proposed to solve this problem [3]. However, this solution would not render adequate results when taking into account that some of the URLs point to outdated resources that are no longer included in search indexes or harvested by preservation services such as the Internet Archive.

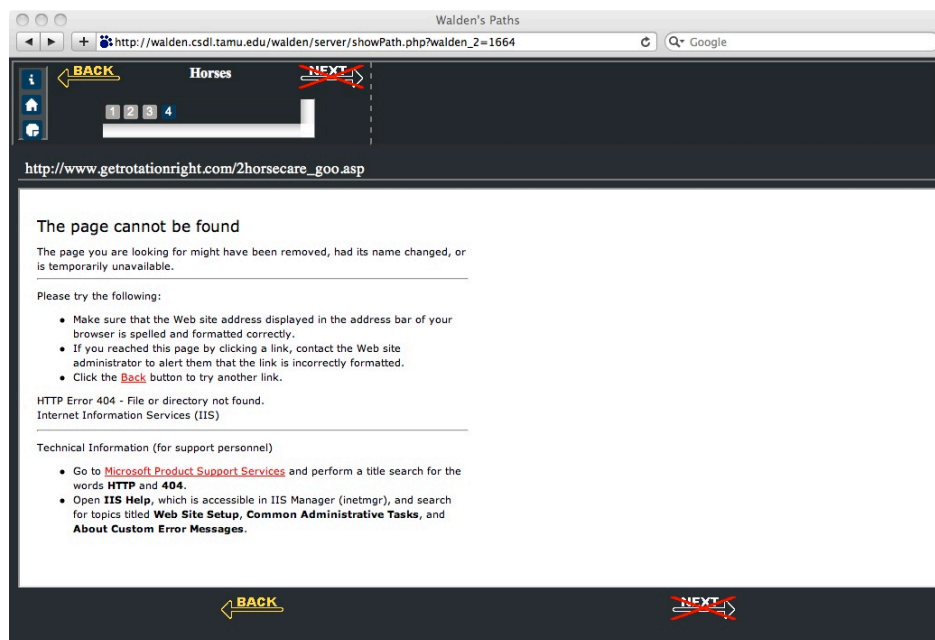


Figure 13: Example of a “Soft 404” error viewed through the Walden’s Paths user interface. This missing document makes the collection semantically incomplete.

As a solution to this problem, similar documents for the missing resources can be found in the Web. These documents are known as “duplicates” (or “near-duplicates”). Although the detection of duplicates has focused on techniques to optimize Web crawling, finding these duplicate documents can be used to preserve distributed collections. For example, the procedure for calculating the contiguous subsequences of tokens in a document introduced by Broder et al. [31, 32] can be used to cluster documents according to their contents to provide suitable replacements for missing or unstable documents. Charikar introduced a hashing function that hashes changes relative to changes in a given input set [57]. As a consequence, entire Web pages or a subset of pages can be compared to their hash values. Later on, Henzinger compared the two

techniques introduced by Broder and Charikar and found that they perform equally well in identifying near duplicates from different sites [75].

More importantly, research has been carried out that directly supports the notion that a lost page can often be replaced (or similarly restored) by another page which can be found in the Web. Fetterly et al. created clusters of near duplicates Web pages and measured their similarity using shingles and other procedures [76]. They found that 28% of their documents were duplicates and 22% were identical. Additionally, Baeza-Yates et al. analyzed genealogical trees from Web pages and found that a portion of the web is created based on existing content [37].

To address the problem of restoring incomplete collections, I used the lexical signatures of the other “valid” documents (i.e. that are not missing) within the collection to find a suitable replacement for the absent resource. In the best-case scenario, I was able to find the actual resource that is missing. However, my main focus was on finding similar resources that can act as surrogates (or placeholders), contribute towards the continuity of the narration and restore the semantic completeness of the collection.

Thus, restoring incomplete collections deals with analyzing the original user’s intent, cognition and the Web search patterns that were utilized when creating a collection. Jansen et al. conducted a study that analyzed search engine query patterns [77]. As a result, they found that 35% of the queries were unique; whereas 22% of the queries exhibited slight variations and 43 were duplicates from the original set. Jansen and Spink compare nine studies that provided insights about the patterns found in Web searches from 1997 to 2002 [78]. As a result of this study, the researchers found that the

length in the query strings was decreasing and that nearly half of the queries were tied to people, places and things.

Studies have also examined the demographics, session statistics and query types. Weber and Jaimes carried out an analysis of the logs from Yahoo! Search to determine who searches for what and how they do it [79]. Baeza-Yates et al. conducted a study that utilized supervised and unsupervised learning techniques to identify the user's intent in a search engine query [80]. On the other hand, Zaragoza et al. investigated if the problem of web searching was ultimately solved and if the rankings across search engines were the same [81]. As a result of their inquiry, they found that Google, Yahoo! and Bing exhibit good performance with commonly-issued queries and for navigational queries. However, differences in performance surfaced with non-navigational queries. Additionally, Pass et al. found in their study that queries have an average length of 3.5 terms, 28% of queries are a reformulation of a previous one and that between 12% and 28% of queries include a local aspect (i.e.: Texas Aggies) [82].

Finally, Adar et al. explore the relationship between the number of changes in the content of page and the visitation patterns of its users [83]. The conclusions from their study support the known belief that pages that change frequently are visited more often when compared to pages that exhibit a more static behavior. However, the amount of change that sites exhibit across the web varies – which led the researchers to conclude that the action of users returning to a given site is correlated to the information that is changing in a particular document. Additionally, sites that exhibited some degree of change within shorter periods of time were revisited more frequently as well.

5.1 Experiment Setup

At its backend, the Walden's Paths project uses a MySQL database. This database stores the metadata for each path and its nodes. Title, abstract and language describe each path; while metadata for the nodes includes the resource URL and general notes. Given that users of Walden's Paths created all the paths in the database, it is safe to assume that the nodes share semantic ideas and follow a similar cognitive pattern. For this experiment, a replacement was considered suitable if it shared similarities with the missing resource and it conveyed the semantic meaning of the rest of the path. I initially considered paths that fulfilled three requisites: First, they must contain at least one node. Second, the nodes must have links to web resources; and third, the paths must be in English. Thus, the initial sample from the collection consisted of 948 paths that fit these criteria.

The documents from the resource URLs for each node were retrieved using a similar procedure to the one I used in chapter IV. The procedure consisted of seven steps. First, I attempted to fetch the "absolute URL" for each given URL allowing 10 redirects per URL entry: if it exceeds 10 redirects the URL entry was discarded and not used in the study. Second, the web resources were downloaded using Python's HTTP protocol client with a timeout value of 10 seconds. Third, and because of the diversity of documents in the web, I decided to focus only on HTML files. Thus, the retrieved documents were then identified as HTML files using Python-magic and non-HTML files were discarded. Fourth, I checked the HTML documents for "Soft 404" errors using the procedure based on lexical signatures that I will described in chapter IV and in [84].

Fifth, I removed the HTML elements using Python's Beautiful Soup libraries. Sixth, I used functions and procedures found in Python's Natural Language Toolkit (NLTK) to eliminate stop words, remove punctuation using string translation functions, convert to lowercase and stem each term using a Lancaster stemmer. Finally, the resulting lexical signatures were stored as individual documents in a folder structure, where each folder represented a different path.

To carry out this experiment, I assumed that the paths were "incomplete" meaning that one of the resource URLs from the nodes could not be retrieved. I did this by pretending that the document referenced in the metadata from one of the nodes was "missing" and could not be retrieved. The node that referenced the missing resource was randomly selected from each path. This allowed me to compare the original missing document and its replacement.

The replacements for the missing resources in the nodes were found using a search engine. I used the Microsoft Bing Search API because it allowed 5000 queries per month for free. Also, Bing is based on MSN Search and its document index was known to be updated more frequently when compared to other search engines [25]. I gathered the five most significant lexical signatures – which describe and characterize each incomplete path – which I used to create a query string taking into account Phelps and Wilensky's previous work [54]. Then I retrieved the resources from the URLs of the five top results. This retrieval was done at run time and followed the seven-step procedure that was used for the resource URLs in the path nodes.

5.2 Results

For this analysis, I restricted the analysis using the paths in the sample that contained at least three nodes with retrievable HTML documents. This condition was enforced to provide contextual information to the retrieval algorithms for my experiment. However, it further reduced the sample to a working set of 447 paths. Paths with three and four nodes were the most common cases. I hypothesized that a better notion of context, provided by a larger number of nodes and documents, would allow my processes to obtain better replacements for missing resources.

I validated the assumption that the nodes within a path share semantic ideas and follow a stream of thought. For this purpose, I calculated the cosine similarity between the resource in the incomplete node and the rest of the documents in the path. I found that the vectors formed with the documents within a same path are not identical, but they do share some similarities. Figure 14 shows the average cosine similarity between a missing document and the other valid documents within the path. I also calculated the link similarity between the missing and the valid documents, but this measure did not provide any statistical evidence of similarity.

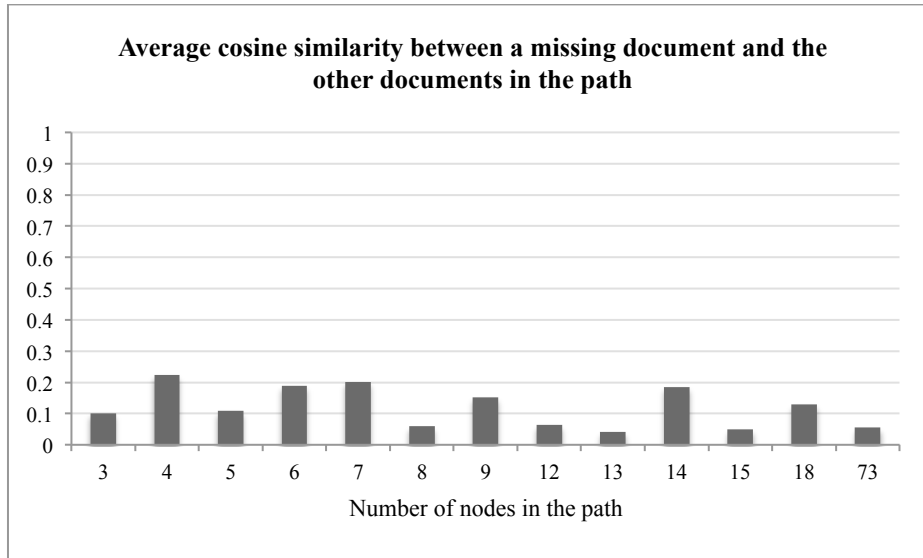


Figure 14: Average cosine similarity between a missing document and the other valid documents within the path.

I selected the replacements using two procedures. In the first, I used the top-ranked result from the search engine API. For convenience, I will refer to this method as “top-ranked”. For the second procedure, which I will refer to as the “top-similar” method, I evaluated if the most similar documents from the search results could be used as suitable placeholders. For this method, I calculated the cosine similarity between the documents in the top five search results and selected the two that had the shortest distance between them. I then chose the topmost-ranked result from the resulting pair. I evaluated my algorithms using the resemblance measure, which was defined in the Shingling algorithm [31], and the cosine similarity between the original document and its surrogate. The cosine similarity was calculated using NumPy.

The average resemblance and cosine similarity were very close for both methods. For resemblance, the average values were 0.735 for “top-ranked” and 0.738 for the “top-similar” method. With resemblance, optimal cases are closer to one. Likewise, the

average values for calculated the cosine similarity were 0.148 for “top-ranked” and 0.166 for the “top-similar” method. For cosine similarity, optimal values are closer to zero. Given the magnitude of these values, using resemblance as a similarity measure was more accurate when expressing the relationships and common ground between documents. However, the results vary greatly when considering that the number of documents in each path that were used for the calculations. Figure 15 and 16 show the distribution of the average resemblance and cosine similarity when grouped by the number of nodes in each path. Thus, the analysis of the data represented in these figures reveals that they do not truly represent the relationship between the number of documents used to extract the lexical signatures and the replacements obtained and that further analysis was needed.

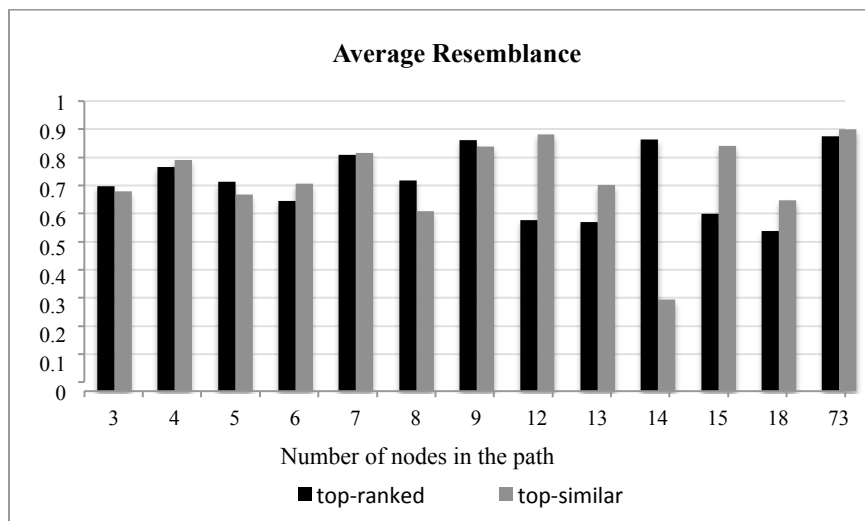


Figure 15: Average resemblance grouped by the number of nodes in each analyzed path.

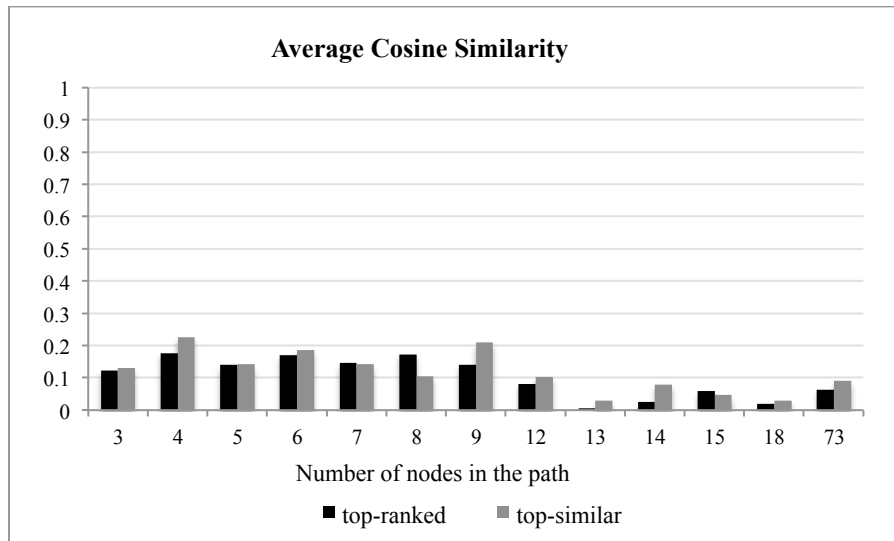


Figure 16: Average cosine similarity grouped by the number of nodes in each analyzed path.

Next, I calculated the quadratic mean between the similarity matrix of the original documents in the path and the similarity matrix of the original documents with a replacement resource in place. I calculated the quadratic mean between the similarity matrices to determine how the surrogates would fit among the original documents and their capability to restore the semantic meaning of the path. I performed this calculation for the “top-ranked” and the “top-similar” methods. Taking into account that these are two similarity matrices, the best-case scenario is found with values closer to zero. Figure 17 shows the Average Quadratic Mean between the original and modified similarity matrices grouped by the number of nodes on each analyzed path.

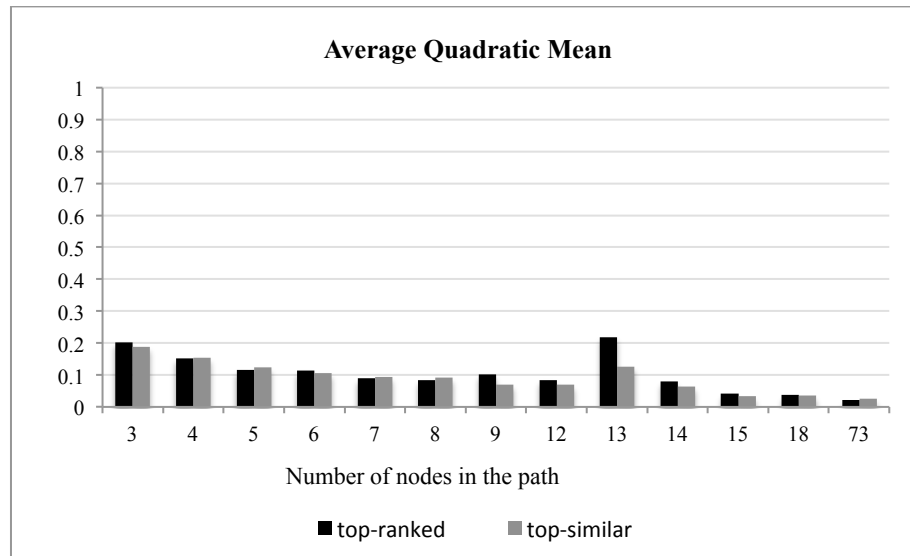


Figure 17: Average quadratic mean between the original and modified similarity matrices grouped by the number of nodes on each path.

Using the quadratic mean between the two matrices, I observed that there is correlation between the number of nodes in the path and the quality of the document surrogates. If a path has more nodes, the extraction algorithm has more information available and it is able to obtain more significant lexical signatures, thus retrieving better document surrogates in the end. Figure 16 shows that there is a tendency leaning towards zero as more nodes are present in a path collection. However, figure 16 also shows an irregular spike in the group of paths that contained 13 nodes. In this case, the algorithm was unable to find any suitable replacements, which caused the quadratic measure to augment drastically. Looking closely at the source documents, the lack of a main theme in these paths caused the algorithm to be unable to extract relevant lexical signatures from these documents.

5.3 Discussion and Implications

Finding documents that can semantically restore a document collection is a complicated task. Thus, I faced three obstacles for this task. First, the variability in the documents served as Web pages makes content fluctuate greatly across the Web. Second, the combined probabilities of occurrence from the terms in a Web page can have a tendency to introduce false positives. Although I minimized this tendency by stemming and streamlining the contents of the documents, my approach would work better in a specialized domain. And third, capturing user intent can be problematic when dealing with a diverse corpus of documents.

I relied on resemblance and cosine similarity to quantify the relationship between the original document and its possible replacement. Likewise, I also aimed to quantify the relationship between the surrogate and the rest of the documents in the path. Calculating the determinant of the similarity matrices was an obvious choice, but it was not a good option. I looked at the determinant as a measure that represents the volume formed by the vectors in n dimensions, where n is the number of nodes in the path. However, I found cases where the determinant had a value of zero. This was caused because the algorithm retrieved a document with the same lexical signatures as one already present in the path; thus causing rows and columns in the matrix to be equal. Therefore, I used the quadratic mean instead between the original similarity matrix and the one derived from the modified document set.

An important result from my work was the realization that traditional similarity measures were not adequate to quantify the relationships between documents in the

collections and their replacements. I can point out two cases. First, the link similarity between the assumed missing document and the other valid resources in the collection did not provide any statistical evidence of similarity. This was caused by the high variability in the documents' structural markup and their associated metadata. And second, my analyses showed that cosine distance as a similarity measure made it difficult to fully express the associations and connections between a missing document and its replacement. More so, this difficulty stems from the fact that a missing document in a narrative-based collection not only depends on the resource that it is replacing, but also depends on its neighboring documents. On the other hand, using resemblance as a similarity measure gave us a better idea of the relationships between documents because it is based on the combined probability of occurrence of n-grams.

I must also address if using a different search engine could have changed the outcome of my study. I believe a different search engine could have probably shifted the results on the paths with fewer nodes. However, the resemblance and cosine similarity measures would not have varied significantly for more populated paths.

CHAPTER VI

CATEGORIZING CHANGE IN AN INSTITUTIONALLY MANAGED REPOSITORY³

Categorizing the degree of change affecting the documents in a digital collection over time is a complex task. As I have stated previously in the several chapters of this dissertation, categorizing this degree of change is not a binary problem where documents are either unchanged or they have changed so dramatically that they do not fit within the scope of the collection. Therefore, there is growing need for a taxonomy that will allow curators and scholars to classify documents according to the amount of change that they exhibit. In this chapter, I will describe a categorization system for documents that have been affected by unexpected changes in content. Furthermore, this categorization system lays the groundwork and foundation for detecting unexpected changes in content within the boundaries of a digital collection.

6.1 A Distributed Collection within an Institutional Digital Library

I harvested the metadata for the conference proceedings found in the Association for Computing Machinery Digital Library, which I used as a document corpus. The ACM Digital Library stores and maintains the “Full text of every article ever published

³ Part of this chapter is reprinted with permission from "Analyzing the Perceptions of Change in a Distributed Collection of Web Documents" by L. Meneses, S. Jayarathna, R. Furuta, and F. M. Shipman, 2016, in *Proceedings of the 27th ACM Conference on Hypertext and Social Media - Hypertext 2016*. Copyright 2016 by ACM.

by ACM and bibliographic citations from major publishers in computing”, which includes the links to the actual conference sites as distributed resources hosted externally and therefore more prone to be affected by unexpected change.

The ACM maintains a list of the conference proceedings (<http://dl.acm.org/proceedings.cfm>), which I retrieved on 9/27/2014 and used as a starting point. Then I followed each hyperlink to a metadata page that displayed basic information for each corresponding conference and workshop, which in turn allowed me to extract the external URLs. As a result of this procedure, I extracted 6086 URLs – out of which 2001 were unique. The reason for the large number of duplicates in the group of URLs is that the ACM Digital Library displays the related and upcoming conferences in each of the series.

The Web pages corresponding to the unique external conference URLs were retrieved using three steps. First, I attempted to fetch the “absolute URL” for each given URL. My methods allowed 10 redirects per URL entry; if it exceeded 10 redirects the URL entry was discarded and not used in the study. Second, the Web resources were downloaded using Python [85] and its HTTP protocol client with a timeout value of 10 seconds. Finally, I stored the resulting HTML file as an individual document in a folder structure, where each individual filename consisted of a string of random length (between 15 and 36 characters) that I generated by concatenating a random combination of 25 letters (a to z) and numbers (0 to 9). The probability of generating a filename that already exists is minimal: $N/(36^R)$ where N is the number of documents in the actual file path and R=15 in the worst case. Additionally, I also stored the metadata associated with

each retrieved page in an XML file. This metadata included the anchor text, URL requested, URL retrieved, HTTP headers and response code. Table 1 shows the frequency distribution of the retrieved pages according to their HTTP response codes. Approximately 75% of the page requests resulted in a response code indicating success (200), which means that no problems were found when trying to fulfill the request. The remaining pages were mostly divided among page not found (404) responses and timeouts.

Table 1: Frequency distribution of the retrieved pages according to their HTTP response codes.

Response Code	Count	%
200	1492	74.56%
300	2	0.10%
404	276	13.79%
403	20	1.00%
401	1	0.05%
410	3	0.15%
504	1	0.05%
503	2	0.10%
500	4	0.20%
Timeout	200	10.00%

Then I proceeded to inspect and categorize the 1492 pages that were retrieved with a 200 HTTP response code. I categorized these pages into three categories by evaluating the relationship between the anchor text and the corresponding retrieved page. As a result of this categorization, I found that 917 pages with contents that were “clearly correct” and 531 were incorrect. Additionally, I was unable to evaluate 44 pages because their contents didn’t provide enough information to make an accurate assessment. These pages could have been placed into the “incorrect” category, but I

decided to use an additional category to make the experiment as transparent as possible.

Figure 18 shows this classification.

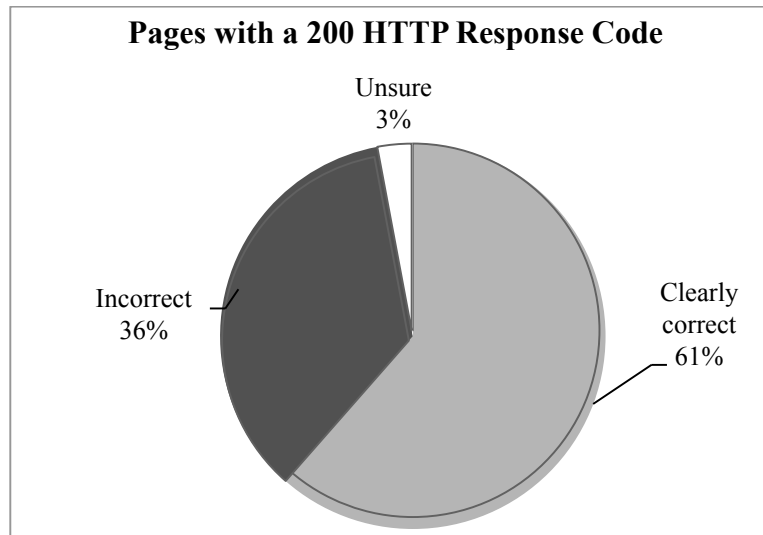


Figure 18: Distribution of the pages that were retrieved with a 200 (OK) HTTP response code.

6.2 Categorization of the Types of Change

The 531 pages that were reported by the HTTP server as being correctly retrieved but were clearly not the original contents were then analyzed in an effort to understand how conference sites degrade over time. The coding scheme evolved through examination of the particular collection rather than using a pre-defined classification scheme.

In the end, nine categories were used to classify the “incorrect” pages, which I list in (approximate) order of severity. These nine groups provide insight regarding the different stages that conference pages go through until they are ultimately abandoned:

1. **Kind of correct:** (197 entries) Pages that contain related content, but they do not fully match the semantic concept encapsulated in the anchor text. When taking into account conference proceedings, these pages often link to a different year in the conference series. For example: Anchor text “Conference X 2006” references the Conference X 2009 site.
2. **University/institution pages:** (36 entries) This case that surfaces when a site has been taken down, but the server configuration redirects the user to its parent institution. In cases dealing with conference sites, servers would usually redirect the user to the website of the University that hosted the conference or to a related professional organization.
3. **Directory listings pages:** (18 entries) Pages displaying a listing of files or a “Hello World” page. Probably caused by an error in the server configuration. Figure 19 shows a screenshot corresponding to the ASPLOS 98 conference site that displays an example of this case. Here the original content looks to still be available but the new web server does not recognize homepage.html as a default page to view for this URL.
4. **Blank pages:** (141 entries) pages that returned no content.
5. **Failed redirects:** (30 entries)
6. **Error pages:** (17 entries) Pages that specifically state that an error has occurred.

7. **Pages in a different language:** (32 entries) Pages that didn't match the language found in the anchor text. Most of these pages were in a language different than English.

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
Parent Directory		-	
asplos8.htm	07-Feb-2003 13:59	9.0K	
asplos8.ps	07-Feb-2003 13:59	154K	
camera.html	07-Feb-2003 13:59	591	
camera.txt	07-Feb-2003 13:59	10K	
cfp.html	07-Feb-2003 13:59	4.9K	
cfp.ps	07-Feb-2003 13:59	66K	
copyright.html	07-Feb-2003 13:59	3.2K	
form.txt	07-Feb-2003 13:59	486	
guidelines.html	07-Feb-2003 13:59	2.3K	
homepage.html	07-Feb-2003 13:59	3.7K	
logo.gif	07-Feb-2003 13:59	442	
missionfront.GIF	07-Feb-2003 13:59	24K	
policy.html	07-Feb-2003 13:59	3.2K	
reg.txt	07-Feb-2003 13:59	1.6K	
sj.jpg	07-Feb-2003 13:59	18K	
sj.uu	07-Feb-2003 13:59	25K	
specs.html	07-Feb-2003 13:59	2.2K	
student.txt	07-Feb-2003 13:59	2.3K	
submissions.html	07-Feb-2003 13:59	829	
submissions.txt	07-Feb-2003 13:59	1.0K	
tutorial.html	07-Feb-2003 13:59	8.7K	

Figure 19: Screenshot of the ASPLOS 98 site showing an example of a “directory listing page”. Accessed at <http://arch.cs.ucdavis.edu/ASPLOS98/> on 9/27/2014.

8. **Domain for sale pages:** (17 entries) Pages that indicated that the domain name registration has lapsed and it is being sold by a registrar, or taken over

by a third party in order to profit from the sale. Figure 20 shows a screenshot corresponding to the DGO 2010 conference site that displays an example of this case.

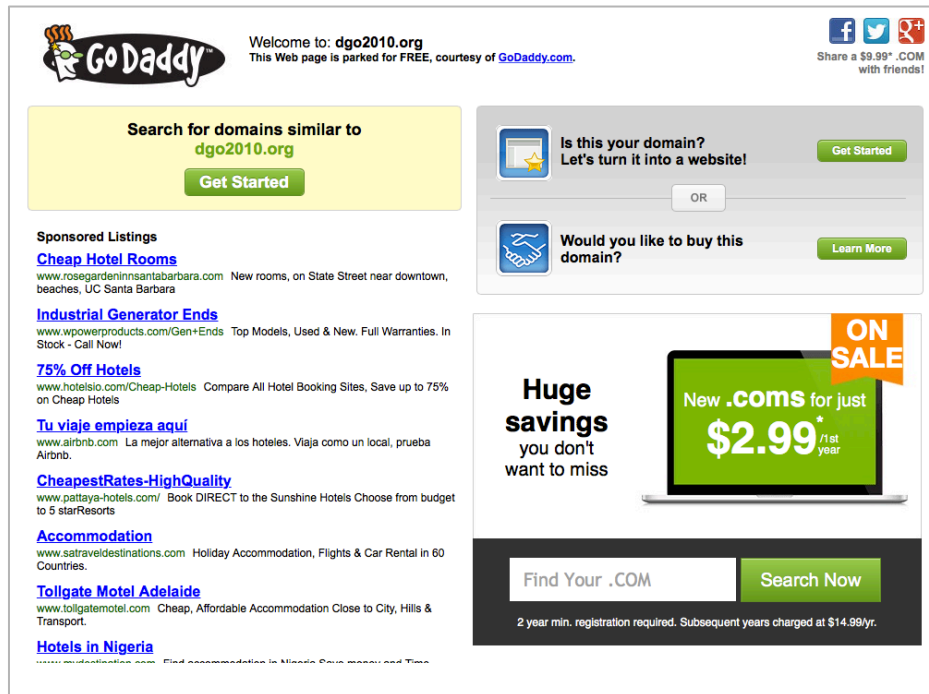


Figure 20: Screenshot of the DGO 2010 site showing an example of a “domain for sale page”. Accessed at <http://www.dgo2010.org> on 9/27/2014.

- 9. Deceiving pages:** (43 entries) Pages that have been taken over by a third party. The content displayed in these pages is totally unrelated to the original purpose of the site. I believe that these pages were not created to deceive users, but as an attempt to manipulate the PageRank algorithm [23]. Figure 21 shows a screenshot corresponding to the IDC 2004 site that displays an example of this case. On the other hand, figure 22 shows an example of a

subcategory among these entries, where the content or presentation of the page is related to the original content but the links are unrelated.



Figure 21: Screenshot of the IDC 2004 site showing an example of a “deceiving page”. Accessed at <http://www.idc2004.org> on 9/27/2014.

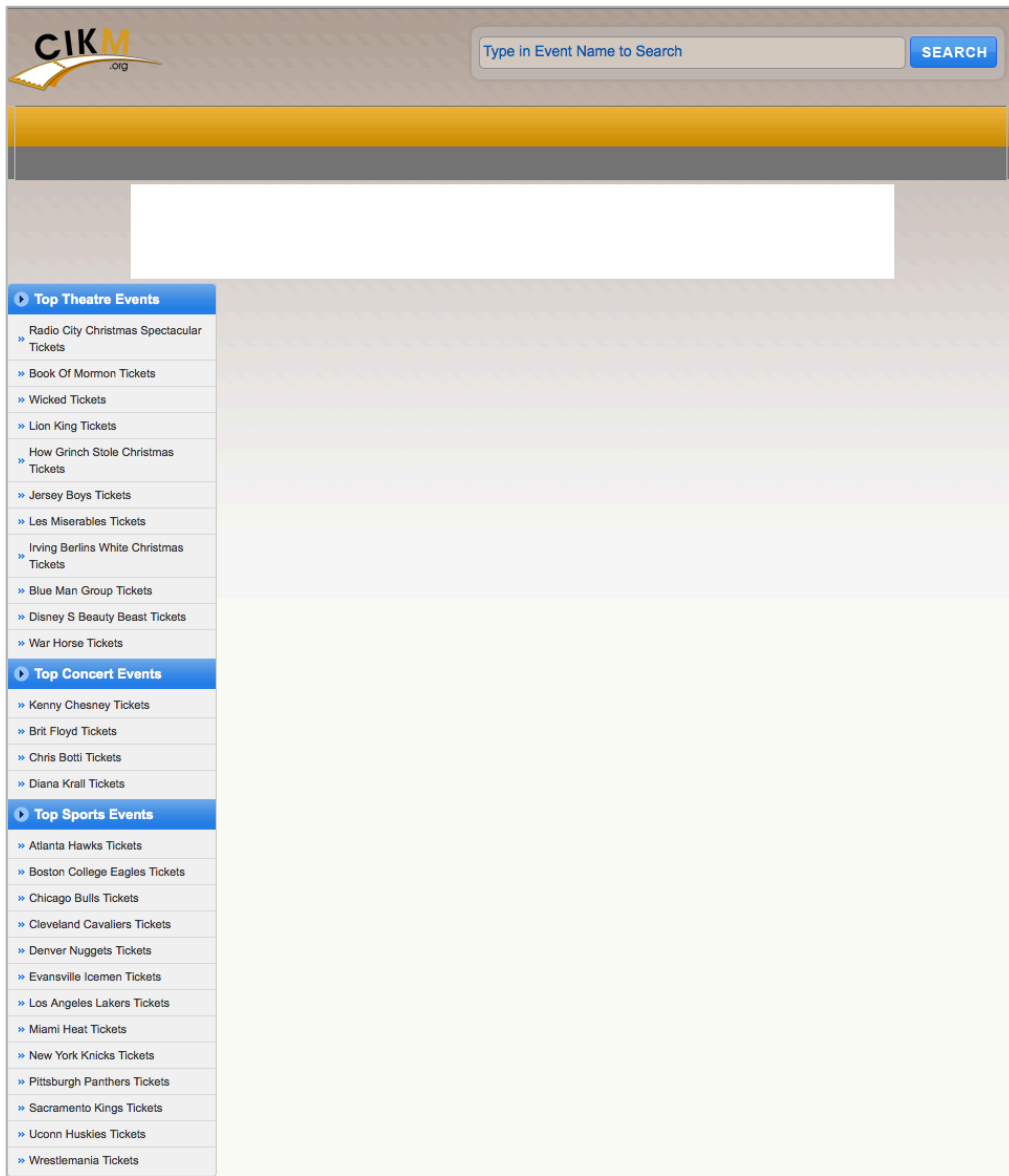


Figure 22: Screenshot of a “deceiving page” where the layout is related to the original content, but the links are unrelated. Accessed at <http://www.cikm.org> on 9/27/2014.

Figure 23 shows the overall distribution of the incorrect pages. Many of these links still lead to information related to the original purpose but clearly not to the originally intended materials. There are a number of categories that result when no

content is available depending on how the servers are configured – blank pages, failed redirects, some directory listings, error pages, and university/institutional pages. The remaining pages are perhaps the most problematic, when the web address has been taken over and is either for sale or being used for other purposes.

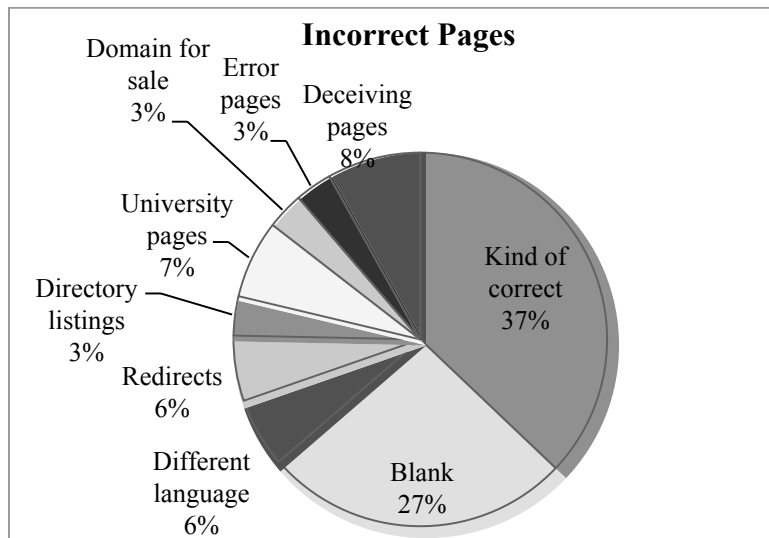


Figure 23: Distribution of the incorrect pages.

Developing techniques to categorize these different issues with web resources will enable the development of tools that focus collection manager attention on the more egregious problems.

6.3 Classification Features

To develop classifiers for the types of issues identified in chapter V, I explored the potential features to help discriminate between the categories observed. In particular, I considered features computed based on the contents and links returned by the initial

request (the *base node*) and the contents and links returned by traversing the links in the base node. This analysis did not examine the potential of the broader network topology to facilitate the classification, as I wanted to limit the number of HTTP requests required for classification to the number of links in the original resource.

6.3.1 Link-based Features

Taking into account that the links on a Web page can be a major factor for determining the relevance of a site within a set of results from a search engine, I decided to use twelve link-based features. These features are divided into topology features, content-type features, anchor-text features and child-node features. Most of the link-based features were computed for the base-node and are based on the number of out-links in a page. In addition, I calculated some of the features for child-nodes that are the valid out-links in these base-nodes. Additionally, I did not differentiate between links to embedded content (e.g. the content of frames, divs, etc.) and navigational links in order to keep the traversal logic as simple as possible and the number of requests required low.

Topology: The local topology of the base node may provide evidence about whether the resource is being used to convince readers to go to another web location or is being used to inflate the ranking of other resources in search results. There are five topology features included.

1. *Out degree:* The number of links in the base node.

2. *Reciprocal link degree*: The number of links from child nodes that point back to the base node.
3. *Edge-reciprocity*: The ratio of reciprocal links to outbound links.
4. *Host links*: The number of links in the base node that point to the same (host) server.
5. *External links*: The number of links in base node that point to other (non-host) servers.

Link-type: The type of content, or the lack of content, at the end of links may also be evidence of the goals of a resource. Thus, there are four features that provide information about the type of content (but not what content) is found at the end of links in the base node.

6. *Broken links*: The number of broken links in the base node.
7. *Multimedia links*: The number of links to sound, image, and video content.
8. *Import links*: The number of links to CSS, text/plain and text/richtext content.
9. *Working non-multimedia links*: The number of outbound links minus the number of broken and multimedia links.

Anchor text: When a page links to another, the anchor text shows the relevant information of the target page or summarizes this information in a way to persuade a user to visit this link. While text analysis could be used to provide insight into this intent, here I include a single, easy to compute feature.

10. *No anchor text links*: The number of links without any anchor text.

Child-node related measures: The types of links found in the child nodes (those at the end of the links in the base node) may also characterize a resources purpose. I included two such features.

11. *Sum of child out-links*: Sum of the number of outbound links across all children.

12. *Sum of child import links*: Sum of the number of links to CSS, text/plain text and text/richtext across all children.

6.3.2 Content-based Features

The content of resources is generally highly indicative of their purpose. Eight quantitative measures related to the content of the resources were included as features for developing classifiers.

Six of these involved quantifying the similarity in topics for two sets of terms.

Image features: The number of images embedded in a resource can be indicative of its type or role.

13. *Base images*: The number of embedded images in the base node.

14. *Child images*: Sum of the number of embedded images in child nodes.

Text-content features: The text associated with resources is the most obvious feature for determining the topics. While a deep understanding of the domain of conference web sites could have been used to develop a domain-oriented expectation model (e.g. a discussion of content submission, organizing committees, schedule, keynotes, travel, and hotels), instead I focused on generating quantitative measures based on the text content that could be potentially valuable across domains.

In particular, I used topic modeling to examine the similarity among the metadata and the contents of the base node and the metadata and the contents of the child nodes.

To do this the textual content of the base node is collected into a *base-content text*. I combined the text from the “description”, “keywords” and “title” metadata for the base node into a single *base-header text*. Similarly, the content of the child nodes are aggregated into a *child-content text* and description, keywords, and titles of the child nodes are aggregated into a *child-header text*.

I used Latent Dirichlet Allocation (LDA) to model the content of the four sets of text [86] and I used KL-divergence similarity to measure and compare them pairwise [87]. All six pairs of the text are examined and are the remaining six features, which are:

15. *Base-content text and base-header text similarity.*
16. *Base-content text and child-content text similarity.*
17. *Base-content text and child-header text similarity.*
18. *Base-header text and child-content text similarity.*
19. *Base-header text and child-header text similarity.*
20. *Child-content text and child-header text similarity.*

6.4 Discussion and Implications

I found it particularly interesting (and alarming) that 36% of the resources referenced in the ACM Digital Library were actually incorrect. These alarming numbers hint that not all is lost: curators in the ACM are focusing their efforts in maintaining the articles and research papers that constitute the main items in the collection. However, some of these efforts are unnecessary as most digital repositories nowadays already emphasize and provide methods for the long-term preservation of documents. In perspective, their preservation efforts fall short as they have neglected to curate the descriptive metadata of the items in the collection. In the specific case of the ACM I am advocating for implementing an additional level of curation to deal with the descriptive metadata of the documents in the collection.

In this case, there are two possible approaches that could ensure the correctness of the descriptive metadata. First, create and ingest a snapshot of the conference websites — which also brings forth a set of issues that are beyond the scope of this study. Or second, periodically look into the correctness of the metadata and distributed resources using manual or automated methods. Both approaches are theoretically correct; their implementation and use is very dependent on the type and scope of the collection. In this dissertation, I am focusing on the later by describing methods to ensure the correctness of the external resources.

Additionally, the classification that I am describing in this chapter is not definitive. I do not discard that additional types of documents might surface when analyzing a different collection. However, the main purpose of the taxonomy and

characterization that I presented in this chapter is to illustrate the complexities of change in the Web and that documents in distributed collections can change in ways that are unexpected by collection managers.

CHAPTER VII

RESULTS⁴

Analysis of the rate of conference web site decay shows this collection is consistent with that found in previous work. Although previous work has focused on calculating the age of resources on the Web [88], we estimated the amount of time a conference site has been valid or invalid by using the date found in the anchor text found in the ACM Digital Library and the last modified date from the server response. For example, the JCDL 2011 conference site with an anchor text of “JCDL '11” (from Figure 1) and a last modified date of “01 Oct 2014” was valid for at most 3 years. We ignored the pages that did not report a last modified date, so we ended up with 1267 conference pages. Consequently, we found that the half-life of the corpus – the amount of time that it takes for half of the collection to disappear (in this case 633 pages) – is approximately 5 years, which aligns with previous work that reported the link decay of Information Science journals [65]. We also found that the decay of the collection stabilizes over time, as it is shown in Figure 24.

⁴ Part of this chapter is reprinted with permission from "Analyzing the Perceptions of Change in a Distributed Collection of Web Documents" by L. Meneses, S. Jayarathna, R. Furuta, and F. M. Shipman, 2016, in *Proceedings of the 27th ACM Conference on Hypertext and Social Media - Hypertext 2016*. Copyright 2016 by ACM.

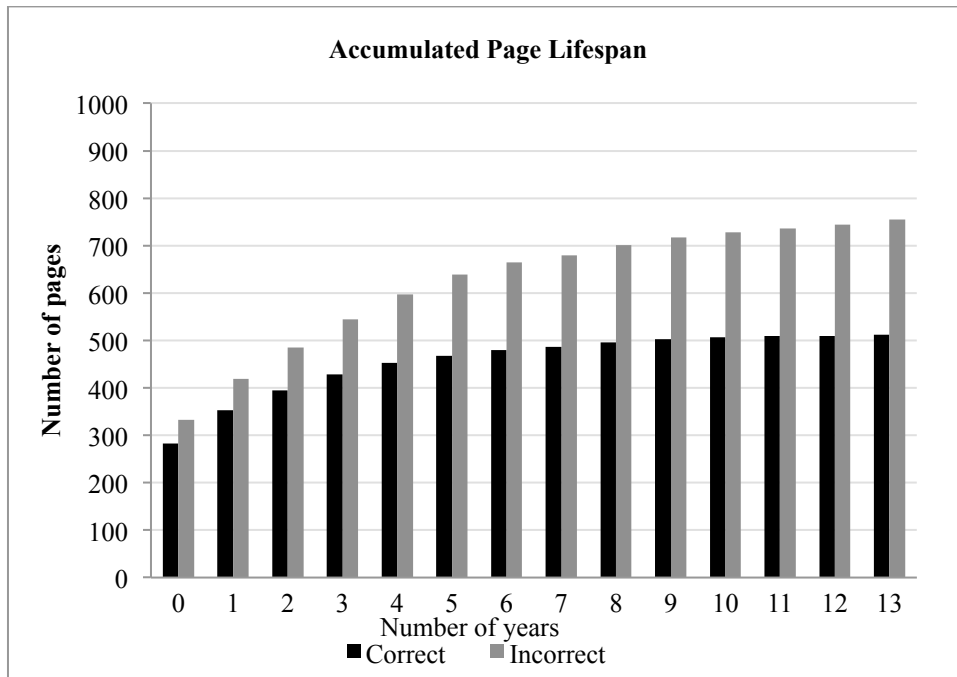


Figure 24: Accumulated page lifespan histogram.

The problem of classifying resources in the ACM conference website dataset can be viewed as a two-phase process. In the first phase, resources can be classified as either *correct* or *not-correct*. In the second phase, resources identified as incorrect are further classified into the different incorrect categories. I explored the success of different classifiers in both phases.

To improve the reliability of my results, each evaluation of the learning schemas was performed by a stratified ten-fold cross-validation [89]. For each evaluation, the dataset is divided into ten equal folds and is trained ten times. Each fold is evaluated with a classifier that was trained with the other nine folds.

The testing corpus consisted of a subset of the overall corpus, removing the resources that did not return a resource (e.g. blank pages, failed redirects, error pages,

and directory listings). I also removed the domain for sale pages as these represent a category that can be identified based on the resulting content that is independent of the topic of the original collection. I added the 44 resources in the unsure category that were not included in the categorization scheme described in chapter VI. The resulting categories and their sizes are shown in figures 25 and 26.

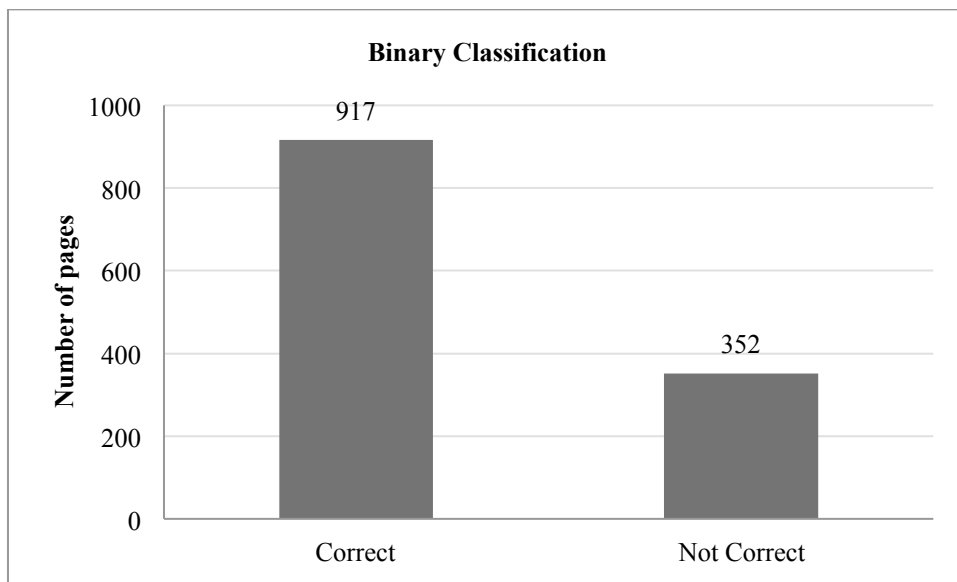


Figure 25: Binary classification of the “clearly correct” category with the “incorrect” and “unsure” categories combined to "not-correct".

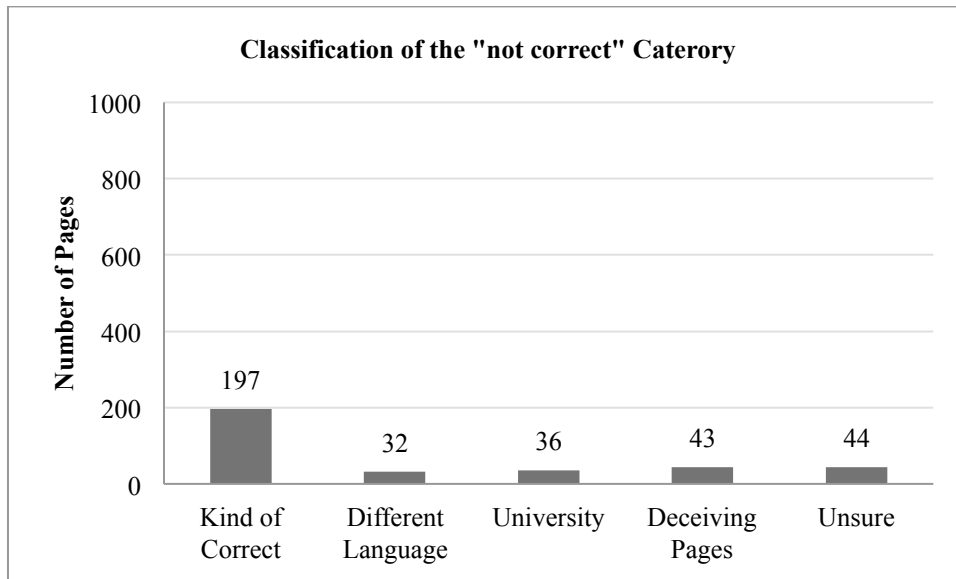


Figure 26: Classification of the "not-correct" category combining the "incorrect" and "unsure" categories.

The kind of severe imbalance in a dataset shown in figures 25 and 26 will lead to poor classification results without any data rebalancing [90, 91]. Under sampling of the majority category is preferred compared to over sampling of minority categories because over sampling leads to over fitting [90]. However, under sampling has the drawback of under fitting for the majority category (in this case, the correct category) due to possible loss of valuable information. This is not a serious problem in this case as the priority is to identify the pages in the incorrect categories. To train the classifiers, random sampling was used to select a number of data instances of the majority class to balance the dataset.

I choose precision, recall and F-measure as the evaluation measures for this work. Prior studies [92, 93] have shown that these measures are independent of category distributions provided that precision and recall are measured at the same time.

7.1 Classification Algorithms

I performed the binary and category classification with 71 algorithms that are implemented in the Weka toolkit [94]. This section will report on the best classification results based on the F-measures from the following classifiers: K*, Decorate, Random Committee, Rotation Forest, Bagging, Boosting (e.g., LogitBoost) and decisions trees (e.g., Random Forest). The algorithmic details of these classifiers are beyond the scope of this dissertation and interested readers are referred to [94, 95].

To finalize the topic model, I explored how varying the number of topics in the LDA model affected classifier performance. As part of these experiments, I varied the number of topics K between 5 and 25. The F-measure of seven classifiers when given the 20 features from chapter VI to classify the incorrect and unsure categories varied between 0.54 and 0.64. After training and testing the category classification data and performing this evaluation, I found that the classifiers reached their optimal performance using 5 topics ($K=5$). Therefore, I used 5 topics for the remainder of these experiments involving the training and testing datasets and the analysis of the classifier results.

The results of the experiments for the "clearly correct", "incorrect" and "unsure" categories as a binary classification problem, and the performance metrics for the 7 most effective classifiers from the evaluation are presented in Table 2. The majority of the classifiers consistently perform with a precision value of 0.62; Random Forest was the best performer for the binary classification. Decorate and Random Committee both exhibit a slightly higher F-measure for "correct" category, but Random Forest offers a substantially better F-measure for the "not correct" category.

To investigate the performance of the classifiers in the "not correct" category, I divided the category classification using the same set of classifiers that were used in the binary classification problem. As Table 3 shows, Random Forest, Rotation Forest and Decorate all perform in the vicinity of a precision value of 0.65.

Table 4 illustrates the comparison of category classification using only Link-based and Content-based features. This result shows that Content-based features (Random Forest 0.48) are not as efficient on their own when compared to Link-based features (Random Forrest 0.613). On the other hand, this result suggests that combining the Content-based features with the Link-based features improves classification performance.

Two categories of pages in the "incorrect/unsure" dataset, the "unsure" and "different language" groups, are the result of the inability to classify the contents returned for a resource as being either correct or incorrect. As I discussed in chapter VI, I originally grouped these two categories into the same "not correct" group — making it possible that some of these pages might contain valid or "correct" pages. The results displayed in table 5 and table 6 show the performance of the classifiers when just the "different language" group and both groups are removed from the training/testing corpus.

Table 2: Binary classification using link and content based features.

	Accuracy	MAE	TP		FP		Precision		Recall		F-Measure	
			C	N	C	N	C	N	C	N	C	N
Decorate	63.62%	0.478	0.64	0.63	0.36	0.35	0.63	0.63	0.64	0.63	0.63	0.63
RandomCommitte	61.74%	0.494	0.67	0.56	0.43	0.32	0.60	0.63	0.67	0.56	0.63	0.59
RotationForest	62.75%	0.455	0.63	0.62	0.37	0.36	0.62	0.62	0.63	0.62	0.62	0.62
RandomForest	64.78%	0.295	0.66	0.63	0.36	0.33	0.64	0.65	0.66	0.63	0.63	0.64
K*	63.19%	0.368	0.62	0.63	0.36	0.37	0.63	0.63	0.62	0.63	0.63	0.63
Bagging	63.04%	0.438	0.64	0.62	0.38	0.35	0.62	0.63	0.64	0.62	0.63	0.62
LogitBoost	61.30%	0.452	0.62	0.59	0.40	0.37	0.61	0.61	0.62	0.59	0.61	0.60

Table 3: Classification of only the “incorrect” categories.

	Accuracy	MAE	TP	FP	Precision	Recall	F-Measure
Decorate	66.38%	0.1904	0.664	0.262	0.633	0.664	0.632
RandomCommittee	62.03%	0.1825	0.62	0.271	0.576	0.62	0.59
RotationForest	67.25%	0.1851	0.672	0.281	0.641	0.672	0.637
RandomForest	67.25%	0.1916	0.672	0.299	0.662	0.672	0.624
K*	61.45%	0.1555	0.614	0.275	0.583	0.614	0.589
Bagging	62.61%	0.2093	0.626	0.357	0.575	0.626	0.562
LogitBoost	63.19%	0.2008	0.632	0.302	0.58	0.632	0.589

Table 4: Link-based and content-based features performance comparison.

	F-Measure	
	Link-based	Content-based
Decorate	0.600	0.471
RandomCommittee	0.604	0.471
RotationForest	0.577	0.476
RandomForest	0.613	0.48
K*	0.603	0.511
Bagging	0.566	0.457
LogitBoost	0.583	0.475

Table 5: Classification of only the “incorrect” categories by removing the “pages in a different language” category.

	Accuracy	MAE	TP	FP	Precision	Recall	F-Measure
Decorate	70.70%	0.2116	0.707	0.33	0.684	0.707	0.677
RandomCommittee	70.70%	0.1946	0.707	0.335	0.685	0.707	0.68
RotationForest	71.02%	0.2144	0.71	0.347	0.668	0.71	0.668
RandomForest	72.29%	0.2167	0.723	0.35	0.714	0.723	0.69
K*	64.97%	0.1756	0.65	0.334	0.637	0.65	0.632
Bagging	68.79%	0.2338	0.688	0.395	0.664	0.688	0.636
LogitBoost	67.83%	0.2213	0.678	0.383	0.639	0.678	0.637

Table 6: Classification of only the “incorrect” categories by removing the “pages in a different language” and “unsure” categories.

	Accuracy	MAE	TP	FP	Precision	Recall	F-Measure
Decorate	83.33%	0.1832	0.833	0.27	0.824	0.833	0.824
RandomCommittee	78.15%	0.1849	0.781	0.329	0.759	0.781	0.765
RotationForest	82.59%	0.1850	0.826	0.321	0.82	0.826	0.813
RandomForest	80.74%	0.1916	0.807	0.341	0.798	0.807	0.793
K*	78.52%	0.1497	0.785	0.344	0.767	0.785	0.767
Bagging	78.15%	0.2153	0.781	0.42	0.778	0.781	0.758
LogitBoost	80.00%	0.1872	0.8	0.367	0.79	0.8	0.784

7.2 Features Analysis

I used Principal Component Analysis (PCA) to analyze the internal structure of the documents and the features that were used to classify them. PCA is a multivariate technique that uses an orthogonal transformation to convert a set of statistically correlated variables into a set of uncorrelated variables, which are defined as the principal components. As a result of the transformation applied to the data, the first principal component becomes a representation of the maximum allowed variance in the sample while fulfilling the requirement of being orthogonal to all the other components. In other words, principal components are ordered according to the amount of variance in the sample that they represent. Thus, each subsequent principal component will contain a lesser amount of variance than its predecessor.

The importance of PCA as a multivariate technique is that it can reveal the internal structure of the data by analyzing its variance. Thus, PCA provides a representation for the correlation between different variables, which can be viewed as a representation of the data from its most informative viewpoint.

In both cases concerning the binary and error pages classification there is a clear separation in the features forming the eigenvectors. The first eigenvector is mostly constituted by link-based features (35-40% of the variance in the data), whereas features extracted from the text found in the pages constitute the second eigenvector (14% of the variance). Tables 7 and 8 show the results from the principal components analysis for the binary and error page classification respectively. However, the rest of the eigenvectors are constituted by a combination of link and text-based features and each vector accounts

for less than 10% of the total variance in the sample. Why is this analysis important? In this case, summarizing the existing classification attributes can create more representative features. Additionally, there is the added benefit of using a classification algorithm with a reduced number of features – which can eliminate potential noise and confusion during the classification phase – bringing the potential to produce better results.

Table 7: Principal components for the features in the binary classification.

Eigenvalue	Proportion	Cumulative	Variables
7.00154	0.3685	0.3685	0.362Internal_links+0.355valid_links+0.352out_degree+0.335Empty_anchor_links+0.326External_links
2.67827	0.14096	0.50946	0.463baselinkheader_outlinksheader_similarity+0.46baselink_outlinks_similarity+0.441baselink_outlinksheader_similarity+0.436baselinkheader_outlink_similarity0.179Imports_of_outlinks
2.01409	0.106	0.61547	0.343Number_of_images_of_outlinks+0.31Number_of_images-0.306Broken_links-0.299External_links+0.274Imports
1.22712	0.06459	0.68005	0.553outlinks_outlinksheader_similarity+0.357Number_of_images-0.322baselinkheader_outlink_similarity-0.283Imports-0.28baselink_outlinks_similarity
1.14241	0.06013	0.74018	-0.599baselink_baselinkheader_similarity-0.412MIME_links+0.349Number_of_images-0.328baselink_outlinksheader_similarity+0.282Number_of_images_of_outlinks
1.03933	0.0547	0.79488	0.592Imports-0.5in_degree-0.35Total_Number_of_outlinks+0.332Number_of_images-0.236baselink_baselinkheader_similarity
0.88156	0.0464	0.84128	-0.605baselink_baselinkheader_similarity+0.362baselinkheader_outlinksheader_similarity+0.354MIME_links-0.301baselink_outlinks_similarity+0.292outlinks_outlinksheader_similarity
0.83466	0.04393	0.88521	-0.753MIME_links+0.408Imports-0.314Number_of_images+0.265Imports_of_outlinks-0.152Empty_anchor_links
0.66665	0.03509	0.9203	-0.608outlinks_outlinksheader_similarity+0.556baselinkheader_outlinksheader_similarity+0.307Number_of_images+0.252baselink_outlinksheader_similarity-0.24baselinkheader_outlink_similarity
0.4305	0.02266	0.94295	0.657Number_of_images_of_outlinks-0.454Number_of_images-0.301Imports_of_outlinks+0.246Empty_anchor_links-0.226Total_Number_of_outlinks
0.31305	0.01648	0.95943	0.573baselink_outlinksheader_similarity-0.502baselinkheader_outlinksheader_similarity-0.329baselinkheader_outlink_similarity+0.317baselink_outlinks_similarity-0.232baselink_baselinkheader_similarity

Table 8: Principal components for the features in the classification of error pages.

Eigenvalue	Proportion	Cumulative	Variables
7.47609	0.39348	0.39348	0.343valid_links+0.332Internal_links+ 0.328out_degree+0.325Total Number of outlinks+0.319in degree...
2.66283	0.14015	0.53363	0.454baselink_outlinks_similarity+ 0.451baselinkheader_outlinksheader_similarity+ 0.442baselink_outlinksheader_similarity+ 0.425baselinkheader_outlink_similarity-0.262External_links...
1.65673	0.0872	0.62082	-0.492Broken_links-0.404Imports-0.373External_links- 0.26Imports_of_outlinks- 0.242baselinkheader_outlinksheader_similarity...
1.22067	0.06425	0.68507	0.527MIME_links+0.523Imports-0.368Broken_links- 0.347External_links+0.262Imports_of_outlinks...
1.03507	0.05448	0.73955	-0.51outlinks_outlinksheader_similarity-0.417MIME_links- 0.37baselink_outlinksheader_similarity- 0.363baselink_baselinkheader_similarity+ 0.326baselinkheader_outlink_similarity...
1.00719	0.05301	0.79256	0.546baselink_baselinkheader_similarity- 0.399Broken_links+0.384External_links+ 0.361baselinkheader_outlink_similarity+ 0.264baselink_outlinks_similarity...
0.87123	0.04585	0.83841	-0.569MIME_links-0.395Broken_links+ 0.379outlinks_outlinksheader_similarity+ 0.289Imports_of_outlinks-0.261baselink_outlinks_similarity...
0.73608	0.03874	0.87715	0.64 baselink_baselinkheader_similarity-0.409External_links- 0.318outlinks_outlinksheader_similarity+ 0.293Broken_links-0.292MIME_links...
0.51853	0.02729	0.90444	0.624Imports-0.462Imports_of_outlinks- 0.305Total_Number_of_outlinks+ 0.273Empty_anchor_links-0.25MIME_links...
0.48176	0.02536	0.9298	-0.736baselinkheader_outlinksheader_similarity+ 0.39 outlinks_outlinksheader_similarity+ 0.343baselink_outlinks_similarity+0.253Broken_links- 0.143baselink_baselinkheader_similarity...
0.34457	0.01814	0.94793	0.477baselinkheader_outlink_similarity+0.422Number_of_images- 0.418baselink_outlinksheader_similarity- 0.401baselink_outlinks_similarity+ 0.252outlinks_outlinksheader_similarity...
0.27295	0.01437	0.9623	0.706Number_of_images-0.383baselinkheader_outlink_similarity+ 0.284baselink_outlinks_similarity+ 0.274Imports_of_outlinks-0.222Broken_links...

7.3 Classifying Documents using Principal Components

Next, I used the PCA to create new features in an attempt to improve the classification results. Table 9 shows the classification results using the eigenvectors representing 95% of the variance as features. Random Forest still provides the best classification results although its performance is marginally improved after using the results from the PCA.

Table 9: Binary classification using the features from the principal components analysis.

	Accuracy	MAE	TP		FP		Precision		Recall		F-Measure	
			C	N	C	N	C	N	C	N	C	N
Decorate	62.79%	0.483	0.63	0.62	0.37	0.36	0.62	0.62	0.63	0.62	0.63	0.62
RandomCommittee	67.27%	0.386	0.72	0.62	0.38	0.27	0.65	0.69	0.72	0.62	0.68	0.65
RotationForest	67.81%	0.442	0.65	0.69	0.30	0.34	0.68	0.67	0.65	0.69	0.67	0.68
RandomForest	69.89%	0.445	0.69	0.70	0.29	0.30	0.70	0.69	0.69	0.70	0.69	0.7
K*	66.50%	0.344	0.6	0.73	0.27	0.4	0.69	0.64	0.6	0.73	0.64	0.68
Bagging	66.56%	0.420	0.67	0.65	0.34	0.32	0.66	0.66	0.67	0.65	0.66	0.66
LogitBoost	60.90%	0.456	0.65	0.56	0.43	0.35	0.60	0.61	0.65	0.56	0.62	0.59

On other hand, the classification of the error pages did not improve using the features from the PCA. As it is shown in table 10, the performance of the classifier was diminished when using the eigenvectors representing 95% of the variance as features.

Table 10: Error page classification using principal components.

	Accuracy	MAE	TP	FP	Precision	Recall	F-Measure
Decorate	76.66%	0.2398	0.767	0.382	0.748	0.767	0.75
RandomCommittee	79.52%	0.1849	0.759	0.399	0.74	0.759	0.742
RotationForest	79.62%	0.203	0.796	0.409	0.792	0.796	0.773
RandomForest	79.25%	0.209	0.793	0.418	0.789	0.793	0.768
K*	76.66%	0.159	0.767	0.39	0.746	0.767	0.747
Bagging	76.66%	0.223	0.767	0.456	0.745	0.767	0.73
LogitBoost	75.93%	0.2151	0.759	0.416	0.74	0.759	0.74

However, I was able to improve the classification of the error pages by viewing it as a binary problem. I did this by removing from the classification the pages in the unsure category, thus focusing on classifying the pages in the “deceiving” and “kind of correct categories”. As before, I viewed the error page classification as a binary problem. I also discarded some categories from the classification: blank pages, redirects, directory listings, error pages and university/institutional pages can be detected to some extent using previous work on identifying Soft 404 error pages [4, 84]. Table 11 shows the results obtained by evaluating the performance of the classifiers, where Random Committee and Random Forest obtained the best results.

Table 11: Error page classification for the “deceiving” and “kind of correct” categories.

	Accuracy	MAE	TP		FP		Precision		Recall		F-Measure	
			D	K	D	K	D	K	D	K	D	K
Decorate	90.79%	0.116	0.92	0.89	0.10	0.07	0.89	0.91	0.92	0.89	0.90	0.90
RandomCommittee	94.47%	0.092	0.95	0.93	0.06	0.04	0.93	0.95	0.95	0.93	0.94	0.94
RotationForest	92.89%	0.109	0.92	0.93	0.06	0.07	0.93	0.92	0.92	0.93	0.92	0.93
RandomForest	94.74%	0.110	0.94	0.94	0.05	0.05	0.94	0.94	0.94	0.94	0.94	0.94
K*	90.53%	0.064	0.85	0.95	0.04	0.14	0.95	0.86	0.85	0.95	0.89	0.91
Bagging	89.21%	0.136	0.92	0.85	0.14	0.07	0.86	0.92	0.92	0.85	0.89	0.88
LogitBoost	87.63%	0.123	0.87	0.88	0.12	0.12	0.87	0.87	0.87	0.88	0.87	0.87

7.4 Discussion and Implications

It is hard to determine the features that are more significant when classifying the documents into different categories. As I have shown previously in this chapter, the best results were achieved when using a combination of features. In this specific study that dealt with identifying the “correctness” of the Web documents in a distributed collection, individual features were not strong indicators on their own to correctly classify the documents. I believe this was caused by the diversity and heterogeneous nature of the data in the collection. In the end, the classification process in this study was far from an ideal case, which would have been to rely on a limited number of features to accurately perform the classification of the documents in the collection.

The use of PCA also presented an interesting scenario. In an optimistic case, the first PCA eigenvector would ideally represent 90% of the data. However, the first eigenvector provided a representation of 40% of the documents in the collection using mostly link-based features. Using this analysis as a baseline, it can be stated from a statistics-based standpoint that the link-based features were more relevant in the classification. The analysis that I presented in this chapter showed that link-based features were not strong indicators to classify the documents on their own. After I

completed the analysis for this study, I was inclined towards studying the classification results obtained from combining link-based features with features extracted from the layout and presentation of Web documents. However, analyzing the presentation of Web documents falls beyond the scope of my dissertation and constitutes a possible direction for future work.

CHAPTER VIII

EVALUATION⁵

One of the main difficulties of the type of research that I am conducting in this dissertation is finding a way to compare the results I obtained using machine learning algorithms against the assessment made by humans. A dataset created specifically for this purpose does not exist. In order to evaluate my findings, I carried out a user study where human subjects were asked to analyze the contents of web documents and assess if these documents belonged to a specific category.

8.1 Experiment Setup

The user study was administered through the Web and used Python and Django as its backend. The user study consisted of two sections. The first section of the study prompted the participants for their demographic data, which included questions regarding their gender, age and educational level. The second section was the main part of the study: participants were asked to go over fifty documents from the ACM corpus that I described in Chapter VII. The fifty documents were randomly selected for each session and consisted of 25 correct and 25 incorrect documents. The data gathered from each participant was then stored in an XML file. The identity of each participant

⁵ Part of this chapter is reprinted with permission from "Analyzing the Perceptions of Change in a Distributed Collection of Web Documents" by L. Meneses, S. Jayarathna, R. Furuta, and F. M. Shipman, 2016, in *Proceedings of the 27th ACM Conference on Hypertext and Social Media - Hypertext 2016*. Copyright 2016 by ACM.

remained anonymous, which means that the XML files didn't contain any sensitive data or any information that could possibly reveal the identity of the study participants. In the end, I collected data from 62 participants – mostly upper-level Computer Science undergraduates – and assessed the validity of 2875 pages. Table 12 shows information about the population regarding their gender, age distribution and educational level.

Table 12: Gender, age distribution and educational level of the population.

Gender		Age Distribution		Level of Education	
Male	53	18 - 24	54	High School Graduate	4
Female	9	25 - 29	6	Some College	55
		30 - 34	1	College Graduate	2
		35 - 39	1	Postgraduate Degree	1

In the second part of the study, participants were presented with a questionnaire regarding each random Web document in order to assess their degree of degradation. More specifically, participants were shown a screenshot of the Web document along with its corresponding anchor text – which was extracted from the ACM Digital Library – and a brief questionnaire that consisted of two questions. The questions were answered using a Likert scale and selecting between multiple choices. The two questions and their possible answers were:

1. Does this Web page show signs of change or degradation?

5- Very Much

4- Somewhat

3- Undecided

2- Not Really

1- Not At All

2. What is the reason for your answer?

1- The content of the page


2- Layout and presentation

3- Images

4- Language of the text

5- Error codes

Figure 26 shows a screenshot of a questionnaire used to assess the degradation of a random document in the second part of the user study

Progress:  - 14 out of 50.

Given that the anchor text for this page is:
SLIP'01
 Does this Web page show signs of change or degradation?

ECE 481/581: ASIC DESIGN

TEACHING ASSISTANT: Ranajoy Nandi, rrandi@cecs.pdx.edu OFFICE HOURS: Mondays 2pm-4pm, VLSI Design Lab, FAB

Projects

- [Project-1](#)
- [Project-2](#)
- [Project-3](#)
- [Project-4](#)
- [Project-4 files](#)

Tools

- [Mentor_Graphics_ModelSim](#)
- [Leonardo_Spectrum_Tutorial](#)

Answer:

Very Much
 Somewhat
 Undecided
 Not Really
 Not At All

What is the reason for your answer?:

The content of the page
 Layout and presentation
 Images
 Language of the text
 Error codes

Figure 27: Screenshot of a sample questionnaire regarding a document in the user study.

8.2 Results

Interestingly, the participants of the study did not have difficulties identifying documents in the correct category. However, this was not the case in all the other categories. Table 13 shows a summary of the user responses for the document classification. Documents in the correct category were correctly identified in 71% of the cases, but users were not able to correctly identify the documents that were incorrect.

Surprisingly, documents in the “error page” category were incorrectly identified in 98% of the cases. Similar scenarios occurred with documents where it was evident that the documents are incorrect, such as in the “domain for sale” and “hello world” categories. The identification did marginally improve in less evident cases, for example in the “deceiving”, “kind of correct” and “not correct” categories. Tables 14 and 15 detail the reasons behind the user classification of the documents. The language of the text and explicit error codes did not influence the user responses. On the other hand, layout, presentation and content of the documents were significant factors in the classification. I also analyzed the amount of time users took to classify documents, but I could not find any statistical evidence that would shed insights to support their choices and decisions.

Table 13: User responses for the document classification in the user study by categories. Shaded: Correct – Not shaded: Incorrect.

	Correct	Error	Deceiving	Hello World	Kind of Correct	Not Correct	Domain for Sale	University
Very Much	456	54	99	54	242	56	59	59
Somewhat	567	5	32	7	299	80	5	37
Undecided	74	0	10	7	45	17	3	10
Not Really	178	0	21	0	75	13	5	14
Not At All	163	1	29	4	63	13	6	13
Total	1438	60	191	72	724	179	78	133
Correctly Identified	0.71	0.02	0.31	0.15	0.25	0.24	0.18	0.28
Incorrectly Identified	0.29	0.98	0.69	0.85	0.75	0.76	0.82	0.72

Table 14: Reasons for the documents that were correctly identified.

Correctly Identified	Correct	Error	Deceiving	Hello World	Kind of Correct	Not Correct	Domain for Sale	University
Content of the page	134	1	18	8	69	14	4	10
Layout & presentation	296	0	37	1	85	19	8	23
Images	568	0	1	0	25	9	0	3
Language of the text	10	0	4	2	3	1	0	1
Error codes	15	0	0	0	1	0	2	0

Table 15: Reasons for the documents that were incorrectly identified.

Incorrectly Identified	Correct	Error	Deceiving	Hello World	Kind of Correct	Not Correct	Domain for Sale	University
Content of the page	162	16	53	28	181	24	46	35
Layout & presentation	149	11	33	7	163	43	4	29
Images	88	3	21	3	175	64	2	29
Language of the text	12	0	17	0	18	3	1	2
Error codes	4	29	7	23	4	2	11	1

The question that remains to be answered is: do the classification algorithms perform better than the human subjects? The answer is that the algorithms in the end were far more consistent. Human subjects were able to identify pages in the correct category slightly better than the algorithms, but failed when asked to identify pages in the incorrect categories. On the other hand, the machine learning algorithms outperformed human subjects when identifying incorrect. Table 16 shows a comparison between the classification by humans and the algorithms. The algorithms outperformed humans with a 2 to 1 ratio (F-measures of 0.64 and 0.32), which for me is a clear indication that managing the effects of unexpected changes in digital collection is more serious problem than I had originally anticipated.

Table 16: Comparison between the classification made by human subjects and the classification algorithms.

Classification	Human Subjects		Algorithms	
	Correct	Incorrect	Correct	Incorrect
Precision	0.71	0.24	0.70	0.69
Recall	0.48	0.46	0.69	0.67
F-measure	0.58	0.32	0.69	0.70

8.3 Discussion and Implications

The decision of delivering the survey over the web was done intentionally. Allowing the participants to complete the survey at a time and location of their choosing was convenient for them and allowed me to obtain a larger number of test subjects. The

alternative would have been to conduct on-site sessions followed by a structured interview, but this alternative would have required a significant investment of resources – which was ultimately unnecessary. The obvious advantage using the alternative evaluation would have resulted in a greater amount of data collected through video recording and the later analysis of all the aspects of the interaction with the system. However, this larger amount of data would have brought along a tradeoff, which would have translated into a greater amount of time to interpret and analyze the results. In the end I made the conscious decision to invest more time designing and fine-tuning the framework for the user study in order to achieve a more streamlined analysis stage against the alternative of spending extended periods of time and resources interpreting the results.

The survey was initially conceived in a way that the subjects would classify 100 documents (50 correct and 50 incorrect). However, upon carrying a preliminary run of the experiment I discovered that classifying 100 documents in one session was overwhelming. Consequently, at that point the test subjects would either start giving erroneous answers or abandon the study altogether. I had to make a decision: decrease the number of documents to classify obtaining more truthful results vs. gaining a greater degree of statistical significance in the study because of the larger number of documents, but with erroneous and biased results. I obviously chose the former alternative, which resulted in the users analyzing 50 documents (25 correct and 25 incorrect) in each session. This decision proved to be the right one, as all the users that agreed to

participate in the user study ultimately completed their assessment of Web pages and provided the evidence that I needed to carry out this evaluation.

CHAPTER IX

CONCLUSIONS

The analysis of the conference website links within the ACM Digital Library shows that institutional archives are not immune to the challenges of distributed collection management. Knowing when changes to a resource require human attention is not a simple problem. This assessment was necessary, especially since the problem areas must be identified in order to apply techniques to recover or replace a broken resource.

As I have stated before, some of my findings align with previous work. Upon my initial assessment, 404 HTTP errors were more prevalent in the corpus. However, upon further inspection I found that 36% of the pages that were allegedly correct (according to their HTTP response codes) were actually incorrect. This is a clear indication that the correctness of a web page is relative and that there is a growing need for methods to categorize and locate likely problematic resources that might require the attention of collection managers.

As the categorization of changes to the collection shows, determining the degree of change affecting a digital collection over time is a difficult task. A web resource may gradually degrade from being correct to one that is still of some use by providing access to related information or information about the institution to contact for more information. Changes in web servers, directory structures, etc. may cause requests to still result in a successful 200 code from the server yet provide no information to the

requestor. A number of these categories were purposely left out of the evaluation of the classification algorithms as these cases can be handled by previous work. More specifically, detecting “blank pages”, “failed redirects”, “directory listings”, “domain for sale” and “error pages” are handled with previous work on identifying Soft 404 error pages [4, 84].

Along the same lines, being able to distinguish between the “kind of correct” and “deceiving” pages is important to collection managers. A contribution of my research is detecting when documents change unexpectedly and fall into more problematic categories such as “kind of correct” and “deceiving pages”.

This last point lead me to investigate the purpose of the documents in the “deceiving pages” category. Although the pages in this category are very diverse in content and presentation, they do share two characteristics. First, the number of links that point to other pages within the site is much greater than the number of out-links. On average, pages in the “deceiving” category had 66 links, which is more than twice the average in the “correct” and “kind of correct” categories (20 and 27 links respectively). And second, the domain names that host these pages once belonged to a reputable institution for number of years (i.e., a conference series) before being abandoned. Consequently, these abandoned domain names have value –not necessarily due to current network traffic but in the perception of their authority/validity. I could hypothesize that these pages are created to manipulate pageRank scores by utilizing a large number of links from a page that once had a high PageRank, but have been taken over by a third party. This problem becomes increasingly interesting when considering

that the cost of creating a web page is small and that some search engines (most notably Google) do not share the overall rankings for their indexed sites, which can lead some parties to abuse these malicious techniques. Examining the variation of other features across categories of change may provide additional insight into the motivations and characterizations of change.

Another potential source of information for identifying the severity of change is web archival services studies. I explored the use of such archives, but chose to leave them out of the current approach. In the specific case of the ACM conference list, some conference sites were not crawled at all. Thus, archival services would have provided an incomplete index that would not have helped me to fully answer my research questions in this dissertation. Additionally, irregular crawling intervals and data embargoes can reduce the value of information from such archives for timely identification of change. More so, the questions that I have explored in this dissertation allow future directions to be explored involving variations of the classification features that will require more computationally expensive operations.

It was not surprising that the documents in the “correct” category were more consistently identified during the user study – especially when considering that human subjects can be biased [96]. However, now that data collection phase of the study is over and its results are analyzed, I believe that the premises established beforehand were influential towards its outcome. In this study, subjects were operating under the assumption that they were identifying and categorizing conference websites. Therefore, I hypothesize that the nature of these sites – being institutional and backed up by

professional and academic organizations – might have led the subjects to believe that the documents were in the “correct” category despite showing explicit symptoms of change. This effect could potentially explain the incorrect classification of some of the documents the “incorrect categories”; most notably the ones in the “domain for sale”, “hello world” and other categories that displayed explicit error codes and content in different languages – which were clear indications of their incorrectness.

The fact that human subjects were unable to identify incorrect pages in the user study brings forth some alarming issues. For instance, is managing the effects of unexpected changes in digital collection a more severe problem than what was originally anticipated? I believe it is. Taking into account that the study participants for the most part identified incorrect pages as false positives (pages that belonged in the collection because their content was correct) is an indication that there is a need for computer-based methods to identify the validity of documents in digital collections. However, there are two questions that remain to be answered regarding the user study. First, did the way that the study was formulated introduce bias that could have influenced the outcome of the study? And second, did the study participants have the necessary background and experience to correctly assess the validity and correctness of the documents? The study was designed to use the common sense in participants when associating the anchor text of a page with its actual contents, thus eliminating any preconceived bias and the need to be familiar with notions about related to digital collections. More so, the majority of the study participants had a computer science background, which made them more than capable of assessing the correctness of the

documents – especially when taking into account their familiarity with error codes and messages.

It is important to highlight that different levels of preservation and curation are needed among different digital collections. Historically, preservation efforts have been primarily concerned with maintaining the primary artifacts in collections; relegating descriptive metadata to a lesser level of importance. There is an underlying notion that descriptive metadata is static: requiring minimal resources to maintain and consequently making it easier to preserve. However, the document collection that I analyzed in this dissertation describes a case study that proves that this is not always the case. In the case of the ACM digital library, the metadata and the external documents are important components of the collection as they provide context and help situate the conference papers. As I have stated in the previous chapters of this dissertation, it is inherently difficult for curators to keep track of external resources. Thus, the case study involving the distributed documents in this institutional repository serves to illustrate the need for automated methods to determine when resources are changing in potentially unexpectedly ways that can disrupt the semantic meaning of the collection as a whole.

There is a wide range of follow-on work related to the classification problem explored here. I limited my approach to features that could be computed quickly with a minimal number of http requests per collection resource. How would variations of these features or more computationally expensive features perform? A pragmatic direction of future work is to develop a software package that combines approaches for determining when http error-codes are likely temporary or permanent, recognizing Soft 404

responses, detecting Web spam, and categorizing the remaining changed pages as described here.

I am also interested in assessing the need for a framework to analyze how current tools affect the management of personal digital collections. Furthermore, people develop personal digital collections consisting of distributed web resources as both reminders that specific resources exist and as a mean to gain reliable access to them. Thus, managing collections of this type is necessary to preserve their value – specially when taking into account that unexpected changes can cause them to become outdated, requiring revisions and needing the removal of no-longer-appropriate resources or replacements for lost resources. I believe that my current findings in categorizing unexpected changes and the possibility of automating these detection methods will ultimately afford tools that support distributed collection management efforts more efficiently.

Another possible direction for future work is developing a framework that deals with unexpected changes in layout and preservation of Web documents. There are possible applications for machine learning algorithms that can detect certain changes in presentation, which in turn can point curators towards potentially problematic resources as the content of the documents are changing over time.

Finally, the research that I have outlined in this dissertation focuses on methods to detect unexpected changes in Web documents within a collection. However, the degree of change that I am focusing on varies depending on the current state of the resource: some documents might not display errors explicitly while others express

changes in content that diverge from the main focus of the collection. More so, my analyses are focused on some of the facets of unexpected change, which are characterized by their problematic detection using traditional methods –thus requiring the assistance of a classification system and automated detection methods such as the ones that we have described in this dissertation.

REFERENCES

- [1] V. Bush, "As We May Think," *The Atlantic Monthly*, pp. 641-649, July 1945.
- [2] F. McCown, C. C. Marshall, and M. L. Nelson, "Why web sites are lost (and how they're sometimes found)," *Communications of the ACM*, vol. 52, pp. 141-145, 2009.
- [3] M. Klein, J. Ware, and M. L. Nelson, "Rediscovering missing web pages using link neighborhood lexical signatures," in *Proceedings of the 11th annual international ACM/IEEE Joint Conference on Digital libraries*, Ottawa, Ontario, Canada, 2011.
- [4] Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins, "Sic transit gloria telae: towards an understanding of the web's decay," in *Proceedings of the 13th international conference on World Wide Web*, New York, NY, USA, 2004.
- [5] H. M. SalahEldeen and M. L. Nelson, "Losing My Revolution: How Many Resources Shared on Social Media Have Been Lost?," in *Proceedings of Theory and Practice of Digital Libraries 2012*, Paphos, Cyprus, 2012.
- [6] P. L. Bogen, R. Furuta, and F. Shipman, "A quantitative evaluation of techniques for detection of abnormal change events in blogs," in *Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries*, Washington, DC, USA, 2012.
- [7] P. Dave, U. P. Karadkar, R. Furuta, L. Francisco-Revilla, F. Shipman, S. Dash, *et al.*, "Browsing intricately interconnected paths," in *Proceedings of the fourteenth*

- ACM conference on Hypertext and hypermedia - HYPERTEXT '03*, Nottingham, UK, 2003.
- [8] D. M. Levy, *Scrolling Forward: Making Sense of Documents in the Digital Age*. New York, NY: Arcade Publishing, 2001.
- [9] J. Rothenberg, "Ensuring the longevity of digital documents," *Scientific American*, vol. 272, pp. 42-47, 1995.
- [10] J. Rothenberg, *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation.*: Council on Library and Information Resources, 1999.
- [11] T. Berners-Lee. (5/3/2015). *The original proposal of the WWW, HTMLized*. Available: <http://www.w3.org/History/1989/proposal.html>
- [12] F. McCown, J. A. Smith, and M. L. Nelson, "Lazy preservation: reconstructing websites by crawling the crawlers," in *Proceedings of the 8th annual ACM international workshop on Web Information and Data Management*, Arlington, Virginia, USA, 2006.
- [13] L. Francisco-Revilla, F. Shipman, R. Furuta, U. Karadkar, and A. Arora, "Perception of content, structure, and presentation changes in Web-based hypertext," in *Proceedings of the 12th ACM conference on Hypertext and Hypermedia*, Arhus, Denmark, 2001.
- [14] Google Inc. (5/7/2015). *Google*. Available: <https://www.google.com>
- [15] Yahoo! Inc. (5/7/2015). *Yahoo*. Available: <https://www.yahoo.com>
- [16] Microsoft. (5/7/2015). *Bing*. Available: <http://www.bing.com>

- [17] Internet Archive. (5/7/2015). *Internet Archive: Digital Library of Free Books, Movies, Music & Wayback Machine*. Available: <https://archive.org>
- [18] European Archive. (5/7/2015). *the european archive : home page*. Available: <http://www.europarchive.org>
- [19] CiteSeerX. (5/7/2015). *CiteSeerX*. Available: <http://citeseerx.ist.psu.edu/index>
- [20] National Science Digital Library. (5/7/2015). *NSDL | OER Commons*. Available: <https://nsdl.oercommons.org>
- [21] A. Jatowt, Y. Kawai, S. Nakamura, Y. Kidawara, and K. Tanaka, "A browser for browsing the past web," in *Proceedings of the 15th international conference on World Wide Web*, 2006.
- [22] F. McCown, N. Diawara, and M. L. Nelson, "Factors affecting website reconstruction from the web infrastructure," in *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, 2007.
- [23] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," ed: Stanford InfoLab, 1999.
- [24] F. McCown and M. L. Nelson, "A framework for describing web repositories," in *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, 2009.
- [25] F. McCown and M. L. Nelson, "Search engines and their public interfaces: which apis are the most synchronized?," in *Proceedings of the 16th international conference on World Wide Web*, Banff, Alberta, Canada, 2007.

- [26] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, pp. 11-21, 1972.
- [27] W. B. Croft, D. Metzler, and T. Strohman, *Search engines: Information retrieval in practice*: Addison-Wesley Reading, 2010.
- [28] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval* vol. 1: Cambridge university press Cambridge, 2008.
- [29] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Harlow, England: Addison Wesley, 1999.
- [30] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, pp. 613-620, 1975.
- [31] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, "Syntactic clustering of the Web," *Computer Networks*, vol. 29, pp. 1157-1166, 1997.
- [32] A. Z. Broder, "On the resemblance and containment of documents," in *Proceedings of Compression and Complexity of Sequences 1997*, 1997.
- [33] C. J. Fox, "Lexical Analysis and Stoplists," ed: Prentice-Hall, 1992.
- [34] G. Salton, "Automatic text processing: The transformation, analysis, and retrieval of," *Reading: Addison-Wesley*, 1989.
- [35] P. L. Bogen, J. Johnston, U. P. Karadkar, R. Furuta, and F. Shipman, "Application of kalman filters to identify unexpected change in blogs," in *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, Pittsburgh, PA, USA, 2008.

- [36] The JSON Data Interchange Format. (7/30/2015). *JSON*. Available: <http://json.org>
- [37] R. Baeza-Yates, I. Pereira, and N. Ziviani, "Genealogical trees on the web: a search engine user perspective," in *Proceedings of the 17th international conference on World Wide Web*, Beijing, China, 2008.
- [38] H. Ashman, "Electronic document addressing: dealing with change," *ACM Computing Surveys*, vol. 32, pp. 201-212, 2000.
- [39] H. C. Davis, "Hypertext link integrity," *ACM Computing Surveys*, vol. 31, p. 28, 1999.
- [40] F. McCown and M. L. Nelson, "Usage analysis of a public website reconstruction tool," in *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, Pittsburgh, PA, USA, 2008.
- [41] M. Klein and M. L. Nelson, "Moved but not gone: an evaluation of real-time methods for discovering replacement web pages," *International Journal on Digital Libraries*, vol. 14, pp. 17-38, 2014.
- [42] B. Kahle, "Preserving the Internet," *Scientific American*, vol. 276, pp. 82-83, March 1997.
- [43] W. Koehler, "Web page change and persistence---a four-year longitudinal study," *Journal of the American Society for Information Science and Technology*, vol. 53, pp. 162-171, 2002.
- [44] D. Spinellis, "The decay and failures of web references," *Communications of the ACM*, vol. 46, pp. 71-77, 2003.

- [45] F. Douglass, A. Feldmann, B. Krishnamurthy, and J. C. Mogul, "Rate of Change and other Metrics: a Live Study of the World Wide Web," in *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, 1997.
- [46] S. G. Ainsworth, M. L. Nelson, and H. V. d. Sompel, "Only One Out of Five Archived Web Pages Existed as Presented," in *Proceedings of the 26th ACM Conference on Hypertext & Social Media*, Guzelyurt, Northern Cyprus, 2015.
- [47] M. Kelly, M. L. Nelson, and M. C. Weigle, "The archival acid test: evaluating archive performance on advanced HTML and JavaScript," in *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, London, United Kingdom, 2014.
- [48] J. Cho and H. Garcia-Molina, "The evolution of the web and implications for an incremental crawler," in *Proceedings of the 26th International Conference on Very Large Data Bases*, 2000.
- [49] J. Cho and H. Garcia-Molina, "Estimating frequency of change," *ACM Transactions on Internet Technology*, vol. 3, pp. 256-290, 2003.
- [50] D. Fetterly, M. Manasse, M. Najork, and J. Wiener, "A large-scale study of the evolution of web pages," in *Proceedings of the 12th international conference on World Wide Web*, 2003.
- [51] A. Ntoulas, J. Cho, and C. Olston, "What's new on the web?: the evolution of the web from a search engine perspective," in *Proceedings of the 13th international conference on World Wide Web*, New York, NY, USA, 2004.

- [52] E. Adar, J. Teevan, S. T. Dumais, and J. L. Elsas, "The web changes everything: understanding the dynamics of web content," in *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, 2009.
- [53] C. Olston and S. Pandey, "Recrawl scheduling based on information longevity," in *Proceedings of the 17th international conference on World Wide Web*, Beijing, China, 2008.
- [54] T. A. Phelps and R. Wilensky, "Robust Hyperlinks Cost Just Five Words Each," ed: University of California at Berkeley, 2000.
- [55] J. Teevan, E. Adar, R. Jones, and M. A. S. Potts, "Information re-retrieval: repeat queries in Yahoo's logs," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, Amsterdam, The Netherlands, 2007.
- [56] S.-T. Park, D. M. Pennock, C. L. Giles, and R. Krovetz, "Analysis of lexical signatures for improving information persistence on the World Wide Web," *Transactions on Information Systems*, vol. 22, pp. 540-572, 2004.
- [57] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thirty-fourth annual ACM symposium on Theory of Computing*, Montreal, Quebec, Canada, 2002.
- [58] U. Manber, "Finding similar files in a large file system," in *Proceedings of the USENIX Winter 1994 Technical Conference*, San Francisco, California, 1994.

- [59] N. Shivakumar and H. Garcia-Molina, "Finding Near-Replicas of Documents and Servers on the Web," in *Selected papers from the International Workshop on The World Wide Web and Databases*, 1999.
- [60] S. Brin, J. Davis, and H. Garcia-Molina, "Copy detection mechanisms for digital documents," in *Proceedings of the 1995 ACM SIGMOD international conference on Management of Data*, San Jose, California, USA, 1995.
- [61] G. Forman, K. Eshghi, and S. Chiochetti, "Finding similar files in large document repositories," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge Discovery in Data Mining*, Chicago, Illinois, USA, 2005.
- [62] Z. Dalal, S. Dash, P. Dave, L. Francisco-Revilla, R. Furuta, U. Karadkar, *et al.*, "Managing distributed collections: evaluating web page changes, movement, and replacement," in *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital Libraries*, Tuscon, AZ, USA, 2004.
- [63] M. Nelson and D. Allen, "Object Persistence and Availability in Digital Libraries," *D-Lib Magazine*, vol. 8, 2002.
- [64] W. Koehler, "A longitudinal study of Web pages continued: a consideration of document persistence," *Information Research*, vol. 9, 2004.
- [65] D. H. L. Goh and P. K. Ng, "Link decay in leading information science journals," *Journal of the American Society for Information Science and Technology*, vol. 58, pp. 15-24, 2007.

- [66] F. McCown, S. Chan, M. L. Nelson, and J. Bollen, "The availability and persistence of web references in D-Lib Magazine," *arXiv preprint cs/0511077*, 2005.
- [67] J. Markwell and D. W. Brooks, "Broken Links: The Ephemeral Nature of Educational WWW Hyperlinks," *Journal of Science Education and Technology*, vol. 11, pp. 105-108, 2002.
- [68] F. Harmelen and J. Meer, "WebMaster: Knowledge-Based Verification of Web-Pages," in *Proceedings of the International Conference on Industrial and Engineering Applications of Artificial Intelligence Systems*, 1999.
- [69] M. Ohye. (6/2/2015). *Official Google Webmaster Central Blog: Farewell to soft 404s*. Available: <http://googlewebmastercentral.blogspot.ca/2008/08/farewell-to-soft-404s.html>
- [70] J. Simon. (6/2/2015). *Official Google Webmaster Central Blog: Crawl Errors now reports soft 404s*. Available: <http://googlewebmastercentral.blogspot.ca/2010/06/crawl-errors-now-reports-soft-404s.html>
- [71] B. Tedeschi. (6/2/2015). *Google's Soft 404s are Inaccurate and Often Times Outdated*. Available: <https://web.archive.org/web/20110903071012/http://x-pose.org/2010/06/googles-soft-404s-are-inaccurate-and-often-times-outdated/>
- [72] NLTK Project. (5/3/2015). *Natural Language Toolkit*. Available: <http://www.nltk.org>

- [73] L. Richardson. (5/3/2015). *Beautiful Soup: We called him Tortoise because he taught us*. Available: <http://www.crummy.com/software/BeautifulSoup/>
- [74] A. Jatowt, "Web page summarization using dynamic content," in *Proceedings of the 13th international World Wide Web conference on Alternate track papers and posters*, New York, NY, USA, 2004.
- [75] M. Henzinger, "Finding near-duplicate web pages: a large-scale evaluation of algorithms," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006.
- [76] D. Fetterly, M. Manasse, and M. Najork, "On the evolution of clusters of near-duplicate web pages," *Journal of Web Engineering*, vol. 2, pp. 228-246, 2003.
- [77] B. J. Jansen, A. Spink, and T. Saracevic, "Real life, real users, and real needs: a study and analysis of user queries on the web," *Information processing & management*, vol. 36, pp. 207-227, 2000.
- [78] B. J. Jansen and A. Spink, "How are we searching the World Wide Web? A comparison of nine search engine transaction logs," *Information Processing & Management*, vol. 42, pp. 248-263, 2006.
- [79] I. Weber and A. Jaimes, "Who uses web search for what: and how," in *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011.
- [80] R. Baeza-Yates, L. Calderón-Benavides, and C. González-Caro, "The intention behind web queries," in *Proceedings of the International Symposium on String Processing and Information Retrieval*, 2006.

- [81] H. Zaragoza, B. B. Cambazoglu, and R. Baeza-Yates, "Web search solved?: all result rankings the same?," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010.
- [82] G. Pass, A. Chowdhury, and C. Torgeson, "A picture of search," in *Proceedings of InfoScale*, 2006.
- [83] E. Adar, J. Teevan, and S. T. Dumais, "Resonance on the web: web dynamics and revisitation patterns," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009.
- [84] L. Meneses, R. Furuta, and F. M. Shipman, "Identifying "Soft 404" Error Pages: Analyzing the Lexical Signatures of Documents in Distributed Collections," in *Proceedings of Theory and Practice of Digital Libraries 2012*, Paphos, Cyprus, 2012.
- [85] G. van Rossum, "Python tutorial, Technical Report CS-R9526," ed. Amsterdam: Centrum voor Wiskunde en Informatica (CWI), 1995.
- [86] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993-1022, 2003.
- [87] C. M. Bishop, *Pattern recognition and machine learning* vol. 1: springer New York, 2006.
- [88] H. M. SalahEldeen and M. L. Nelson, "Carbon dating the web: estimating the age of web resources," in *Proceedings of the 22nd international conference on World Wide Web companion*, Rio de Janeiro, Brazil, 2013.

- [89] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995.
- [90] C. Drummond and R. C. Holte, "Severe class imbalance: Why better algorithms aren't the answer," in *Proceedings of the 16th European Conference on Machine Learning: ECML 2005*, Porto, Portugal, 2005.
- [91] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 1263-1284, 2009.
- [92] M. C. Monard and G. Batista, "Learning with skewed class distributions," in *Proceedings of the 3rd Congress of Logic Applied to Technology – LAPTEC 2002*, São Paulo, Brazil, 2002.
- [93] L. Manevitz and M. Yousef, "One-class document classification via neural networks," *Neurocomputing*, vol. 70, pp. 1466-1481, 2007.
- [94] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Burlington, MA, USA: Morgan Kaufmann, 2005.
- [95] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, pp. 10-18, 2009.
- [96] A. Rüping, "Taming the biases: a few patterns on successful decision-making," in *Proceedings of the 19th European Conference on Pattern Languages of Programs*, Irsee, Germany, 2014.