Understanding and Leveraging the Social Web for Information Retrieval

Dissertation

zur Erlangung des akademischen Grades
des Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachgebiet Internet-Technologien und -Systeme
des Hasso-Plattner-Instituts

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Universität Potsdam

im Rahmen der Cotutelle-de-thèse mit der
Faculté des Sciences, de la Technologie et de la Communication
de l'Université du Luxembourg

vorgelegt von Diplom-Wirtschaftsinformatiker

Michael G. Noll

Potsdam, April 2010

Dedicated to my grandpa Karl.

Dissertation Reviewers:

- Prof. Dr. Christoph Meinel, Hasso Plattner Institute, Germany
- Prof. Dr. Thomas Engel, University of Luxembourg, Luxembourg
- Prof. Dr. Matthias Krause, University of Mannheim, Germany

Dissertation Defense Committee:

- Prof. Dr. Felix Naumann, Hasso Plattner Institute, Germany (chairman)
- Prof. Dr. Holger Giese, Hasso Plattner Institute, Germany
- Prof. Dr. Christoph Meinel, Hasso Plattner Institute, Germany
- Prof. Dr. Raymond Bisdorff, University of Luxembourg, Luxembourg
- Prof. Dr. Thomas Engel, University of Luxembourg, Luxembourg
- Prof. Dr. Björn Ottersten, University of Luxembourg, Luxembourg

Date of Defense: July 14, 2010

Abstract

Since its inception in the early 1990s, the World Wide Web (or simply the Web) has shown tremendous growth, providing access to large volumes of information to millions of people from all over the world. In recent years, the Web has also been increasingly used for social interactions and user collaboration. This development has been coined the *Web 2.0* or *Social Web*. The Social Web is based upon a technical architecture and a culture of participation that reduce the barriers of online collaboration and encourage the creation, reuse and distribution of Web content. Particularly, one of the most prominent features of the Social Web is the concept of *collaborative tagging*. Tagging is the act of manually annotating Web resources (e.g. Web pages, images and videos) with keywords called tags that are created on the fly by users. Basically, it allows Web users to express their opinions on how the Web should be organized. As more and more users participate and contribute to collaborative tagging, a new form of classification scheme emerges that is now commonly referred to as a *folksonomy*, a portmanteau of "folk" and "taxonomy". Folksonomies represent a bottom-up approach to annotating and organizing resources that is focused on and driven by end users. As such, they are structurally different from formal, top-down categorization schemes such as ontologies or taxonomies.

In this thesis, we focus on the analysis of folksonomies and collaborative tagging in the context of information retrieval on the Web. We conduct empirical studies of folksonomies and demonstrate how their results can be leveraged to enhance and improve techniques in the research domain. In the first part of the thesis, we present a comprehensive review of state-of-the-art research on folksonomies and collaborative tagging. Next, we describe our empirical and explorative studies of the information and hidden semantics of folksonomies in the context of Web information retrieval. Our results show that user-contributed data in folksonomies provides new, complimentary information about Web resources that is not available through an inspection of the contents of these resources or through traditional types of Web metadata, such as information provided by the authors of these resources. In the second part of the thesis, we present three use cases that demonstrate how the knowledge and experimental results described in the first part can be leveraged for enhancing and improving Web information retrieval. Firstly, we investigate the notion of expertise or "trustworthiness" of users in folksonomies and present our proposed algorithm, SPEAR, for ranking users by their expertise. We evaluate the algorithm and show that it is also resistant to spamming activities. Secondly, we present our approach to personalization of Web search by exploiting folksonomies for profiling of users and Web resources and demonstrate how it can be implemented in practice. Lastly, we explore how the concepts of collaborative tagging and folksonomies can be exploited for Web filtering. We present a case study of a working prototype, TaggyBear, and describe and evaluate its system design and anatomy.

Zusammenfassung

Seit seinen Ursprüngen in den 1990er Jahren ist das World Wide Web stetig und stark gewachsen. Heute ermöglicht es Millionen von Menschen den Zugriff auf eine gewaltige Menge an Daten und Informationen. In den letzten Jahren wird das Web zunehmend auch für soziale Interaktionen und Kollaborationen von Nutzern verwendet. Für diese Entwicklung wurde der Begriff *Web 2.0* bzw. *Soziales Web* geprägt. Das Soziale Web basiert auf einer technischen Architektur und einer Kultur der Partizipation, welche die Barrieren für Online-Kollaborationen über das Web beseitigen und die Nutzer dazu animieren, Web-Inhalte zu erstellen, wiederzuverwenden und auszutauschen. Eines der prominentesten Merkmale des Sozialen Webs ist das Konzept des *kollaborativen Taggen*. Unter Taggen versteht man das manuelle Annotieren von Web-Ressourcen (z.B. Webseiten, Bilder, Videos) mit Schlagworten, welche *Tags* genannt werden. Im Wesentlichen können Nutzer durch Taggen ausdrücken, wie ihrer Meinung nach das Web organisiert werden sollte. Aus diesem kollaborativen Taggen entsteht mit der Zeit eine neue Form von Klassifizierungsschema, für das sich der Begriff *Folksonomie* durchgesetzt hat. Folksonomien stellen eine Art basisdemokratischen Ansatz zum Verschlagworten und Organisieren von Web-Inhalten dar und sind in diesem Sinne strukturell verschieden von formellen, hierarchischen Ansätzen wie etwa Ontologien oder Taxonomien.

In der vorliegenden Dissertation konzentrieren wir uns auf die Analyse von Folksonomien und kollaborativem Taggen im Forschungsbereich des Web Information Retrieval. Wir führen empirische Studien über Folksonomien durch und demonstrieren, wie deren Ergebnisse dazu verwendet werden können, verschiedene Techniken des Forschungsbereichs zu erweitern und zu verbessern. Im ersten Teil der Dissertation präsentieren wir eine umfassende Literaturübersicht aktueller Forschung über Folksonomien und kollaborativem Taggen. Im Anschluss beschreiben wir unsere explorativen Studien über die in Folksonomien versteckten Informationen mit Blick auf das Web Information Retrieval. Unsere Forschungsergebnisse zeigen, dass Folksonomien neue, komplimentäre Informationen über Web-Ressourcen enthalten, welche nicht in deren Inhalten oder anderweitigen Metadaten enthalten sind, wie z.B. in Angaben der Autoren jener Ressourcen. Im zweiten Teil der Dissertation präsentieren wir anhand dreier Anwendungsfälle, wie das Wissen und die experimentellen Ergebnisse des ersten Teils dazu ausgenutzt werden können, Fortschritte im Bereich des Web Information Retrieval zu erzielen. Zunächst untersuchen wir die Expertise oder "Vertrauenswürdigkeit" von Nutzern in Folksonomien und stellen einen Algorithmus namens *SPEAR* vor, welcher Nutzer ihrer Expertise nach einordnen kann. Wir evaluieren diesen Algorithmus und weisen nach, dass dieser auch widerstandsfähig gegenüber Spam-Aktivitäten ist. Im zweiten Schritt präsentieren wir einen Ansatz zur personalisierten Suche im Web, welcher Profile von Nutzern und Web-Ressourcen anhand von Folksonomien erstellt, und zeigen auf, wie dieser Ansatz in der Praxis umgesetzt werden kann. Abschliessend untersuchen wir, wie die Konzepte des kollaborativen Taggen und der Folksonomien für das Filtern von Web-Inhalten verwendet werden können. Wir präsentieren eine Fallstudie einer prototypischen Implementierung namens *TaggyBear* und beschreiben und evaluieren deren Systemdesign und -anatomie.

Acknowledgments

First, I would like to express my sincere gratitude to Professor Dr. Christoph Meinel and Professor Dr. Thomas Engel, my supervisors at the Hasso Plattner Institute in Germany and at the University of Luxembourg, respectively. Both have given me a lot of support and invaluable advice throughout my doctoral research. Professor Meinel, Professor Engel, I am greatly indebted to you for sharing some of your valuable time, and your multi-tasking abilities and level of professionalism never cease to impress. Your contributions and encouragement helped to turn the past years into an amazing journey.

As part of the LIASIT project in the framework of the University of Luxembourg, I have been privileged with financial support from my industrial partner SES ASTRA S.A. and a research scholarship of the Fonds National de la Recherche Luxembourg. The time spent at SES was rewarding, providing commercial and technical insights into the satellite and communication business. I would like to thank all the people at SES for their support, most noteworthy Alexandre Dulaunoy, Thomas Schneider and Gerhard Bethscheider. Alexandre, I am particularly grateful for our brainstorming sessions as well as your stimulations and contributions which have gone beyond the research work described in this thesis.

Additionally, I would like to give many thanks to Ching-man Au Yeung, now a post-doctoral researcher at NTT Communication Science Laboratories in Japan. We got to know each other as Ph.D. students at the International Semantic Web Conference in South Korea in 2007. Since then we have been collaborating on different projects, driven by a joint passion for science and the unknown. Some sections in this thesis greatly benefited from Ching-man's contribution and discussion with him. Ching-man, our teamwork has been much appreciated.

Last but not least, I own a great debt of gratitude to my family and friends for their unconditional support throughout the years of my studies. Even though we are geographically separated, we have always remained close to each other. Words fail to express my appreciation to my love Sabine whose dedication, intelligence, patience and persistent confidence in me has made so many difficult things so much easier during my doctoral research and in my life. Finally, I would like to thank my parents without none of this would have been possible. I am very grateful to them for instilling the desire for learning in me, and for teaching me about integrity, dignity and respect.

To all of you, many thanks, xièxie, merci beaucoup, vielen Dank! —*Michael*

Contents

List of Figures

List of Tables

All truths are easy to understand once they are discovered; the point is to discover them.

Galileo Galilei (1564–1642)

1 Introduction

This thesis presents the phenomena of folksonomies and collaborative tagging on the Social Web. The central theme and main objective is to analyze these user-driven phenomena in order to deepen our understanding of folksonomies, and to leverage this knowledge for enhancing and improving techniques in the domain of Web information retrieval.

In this chapter, we will give a short introduction to the Social Web, folksonomies and the science of Web information retrieval. We will demonstrate the need for and the benefits of studying folksonomies, and present the research questions we aim to answer in this thesis. We will summarize our contributions and conclude the chapter with an overview of the structure of the thesis.

1.1 From the Web to the Social Web

Since its inception in the early 1990s, the *World Wide Web* (or simply ***the Web***) has shown tremendous growth. Starting from the first Web server and a few Web pages that were developed and operated by the "inventor of the Web" Sir Tim Berners-Lee and his team at CERN[1], Switzerland, in 1990, the Web has since evolved into an agglomeration of more than 230 million Web hosts on the Internet in 2009[2], which serve an estimated number of about 20 billion Web pages[3]. Figure 1.1 illustrates this exploding use of the Internet and the Web. Nowadays, the Web is a platform that is used by millions of people to publish and share information "online" and to reference related resources on the Web through so-called hyperlinks, thereby creating a vast, global network of information – true to its name, a literal world wide web – that has revolutionized the way people disseminate and exchange information. Arguably, the full impact of the Web on human society has yet to be understood, and even its inventor Sir Berners-Lee, now director of the World Wide Web Consortium (W3C) and professor at the Massachusetts Institute of Technology (MIT), has refrained from placing it historically [Lan06].

[1] European Organization for Nuclear Research, located near Geneva, Switzerland.

[2] Netcraft.com, November 2009 Web Server Survey, http://news.netcraft.com/archives/2009/11/10/november_2009_web_server_survey.html, last retrieved on March 01, 2010.

[3] Statistics by WorldWideWebSize.com, http://www.worldwidewebsize.com/, as reported in January 2010.

The convenience and speed with which people can retrieve and distribute information on the Web is unprecedented in human history [MRS08]. For example, people can inform themselves about global and local news in real-time, plan trips to remote locations with online route planners, purchase products via shopping or auction sites, consult each other through online forums, download instruction manuals from the homepages of vendors, or perform academic studies through online courses with multimedia content. The information one needs is often only a few clicks away on the Web.

In recent years, the Web has also been increasingly used for social interactions and user collaboration, particularly due to advances in Web technologies such as *JavaScript*, *AJAX (Asynchronous JavaScript and XML)* or *Adobe Flash* that provide a richer, more interactive user experience with the paradigm of "the Web browser is the platform". This development on both social and technical levels has been coined the trend of ***Web 2.0***[4] or ***Social Web***, where Web applications facilitate and stimulate user interactions on the Web and a surge of user-contributed data such as articles, photos or videos can be observed. In other words, the Social Web allows and entices users to not only retrieve but also contribute information, since it is based upon a technical architecture and a culture of participation that reduce the barriers of online collaboration and encourage the generation, reuse and distribution of Web content [AKTV07, CFL09]. Users are encouraged to provide *data* and *metadata* (data about data), particularly in simple and convenient ways such as tagging (which we will describe in detail in the following sections), ratings ("I like / I dislike", "1 out of 5 stars") and comments ("This book is great!", "Have arrived at the conference this morning."). It can thus be argued that the main difference between the "original" Web and the Web 2.0 is not a change in terms of technology but in the form and scale of participation of its users.

As a result, Social Web applications are collecting[5] large amounts of user-contributed data and metadata. For example, the largest encyclopedia ever assembled by mankind is the Web-based, collaborative *Wikipedia*[6] project, a feat it accomplished in 2007 – after just six years of operation – by surpassing the Yongle Encyclopedia which held the record for six centuries. Similarly, users increasingly participate in online communities and social networks such as *Facebook*[7] (400 million participants [Roo10]) or *MySpace*[8] (125 million participants [Arr09]), where users can keep track of what their friends are doing. There are also services such as *Delicious*[9] that allow users to collectively organize and share references to Web resources in an effort to discover and retrieve relevant, high quality content on the Web. The popularity and impact of these and other online

[4] The term "Web 2.0" is closely associated with publisher Tim O'Reilly because of the O'Reilly Media Web 2.0 conference in 2004. According to Tim O'Reilly, the notion of Web 2.0 emerged in a conference brainstorming session between his publishing company *O'Reilly* and its conference partner *MediaLive International* prior to the actual event [O'R05].

[5] The collection and analysis of such user-contributed data also lead to privacy issues [OS10], the discussion of which is however out of the scope of this thesis.

[6] Wikipedia, http://www.wikipedia.org/.

[7] Facebook, http://www.facebook.com/.

[8] MySpace, http://www.myspace.com/.

[9] Delicious, http://www.delicious.com/, a Yahoo! company.

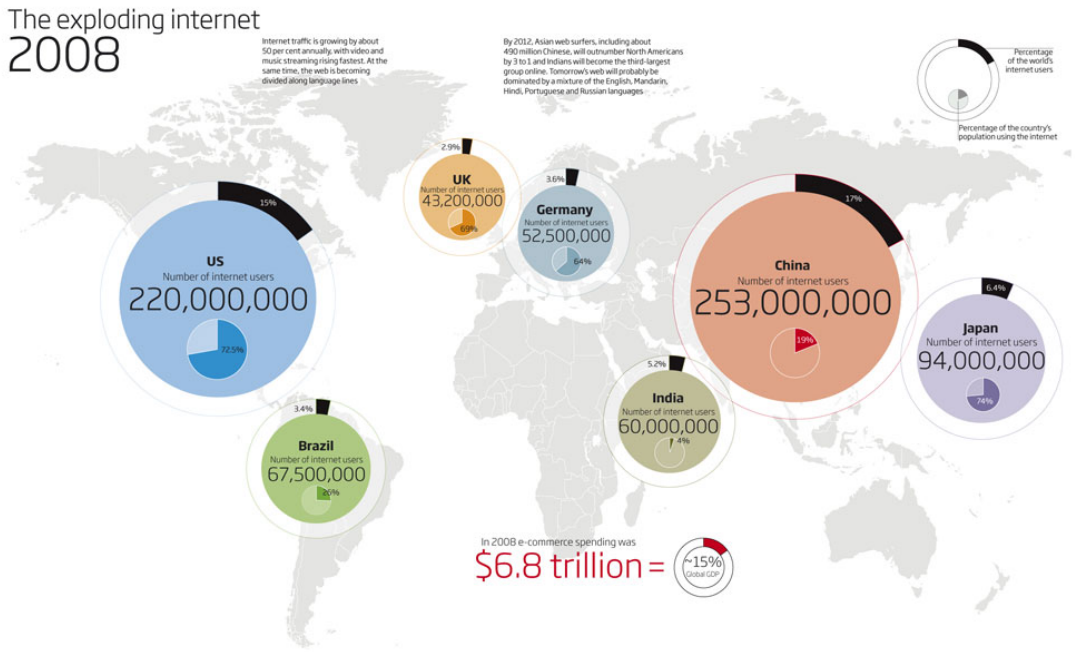


Figure 1.1: **Internet traffic in 2008.** The bubbles show the number of Internet users per country. In the same year, e-commerce spending was 6.8 trillion dollars – about 15 % of the global gross domestic product (GDP) [New09].

services such as *YouTube*[10] (sharing of videos) and *Twitter*[11] (social "microblogging", i.e. publishing very short messages of only a few words to people who have subscribed to one's message stream) have recently come to the point of being used, for example, as communication platforms for supporting political campaigns such as the US presidential elections in 2008.

1.2 From the Social Web to Folksonomies

As we presented in the previous section, user-contributed information on the Social Web also includes *metadata*. For example, users of the photo sharing service *Flickr*[12] may provide additional information (metadata) about a picture (data) stored on Flickr, such as when, where and how it was taken, what is depicted on the photo or other contextual information.

Particularly, one of the most prominent features of the Social Web and its applications is the concept of ***tagging***. Tagging is the act of manually annotating Web resources (e.g. Web documents, images and videos) with metadata in the form of keywords called *tags* that are created on the fly by human users. Basically, tagging allows Web users

[10] YouTube, http://www.youtube.com/, a Google company.
[11] Twitter, http://www.twitter.com/.
[12] Flickr, http://www.flickr.com/, a Yahoo! company.

to express their opinions on how the Web should be organized. This desire of users has in fact been around since the Web's early beginnings: Back in 1997, a user study already reported that organizing collected information for future retrieval is one of the most important problems with using the Web [Cor97]. Since there are certain "costs" associated with tagging a resource (e.g. cognitive costs, costs in terms of time), the act of tagging is also an indication of the perceived value or usefulness of a resource from the user's point of view. Additionally, resources are manually tagged *after* being read, watched or otherwise "processed" by users. Tagging thus represents an explicit user action that happens *a posteriori*, and is therefore believed to provide more useful and relevant information than, for example, search query logs [HRS07]. In the latter case, users must judge the value of a Web resource in the search results *a priori*, i.e. before they visit and process the resource[13].

When tagging is be performed collectively by a group or community of users on the Web, it is called ***collaborative tagging***. The collaborative aspect lies in publicly sharing the information of one's individual tagging activities with other users. As such, collaborative tagging can be considered as a form of the new type of interactions promoted by the Social Web that involves the three entities of *users*, *tags*, and *resources*. The technical, Web-based platforms that allow for collaborative tagging are called ***collaborative tagging systems***. As more and more users participate and contribute to a collaborative tagging system, a new form of classification scheme emerges. This result of collaborative tagging is now commonly referred to as a ***folksonomy***, a portmanteau of "folk" and "taxonomy" that was coined in 2004 by Thomas Vander Wal [Smi04]. The term folksonomy has since seen widespread use both in academia and on the Web at large, and it has also often been used interchangeably with terms such as *social classification*, *faceted hierarchy*, *ethno-classification* or even simply collaborative tagging [Smi04, Mer04, HHLS05].

Folksonomies represent a bottom-up approach to annotating and organizing resources that is focused on and driven by end users. As such, they are structurally different from formal, top-down categorization schemes such as ontologies or taxonomies – a typical example being the *Dewey Decimal Classification system* [OCL] – and represent a more democratic way of annotating resources on the Web. Users in folksonomies need neither to have knowledge of nor to conform to a predefined strategy or vocabulary of tagging. The advantages of such a light-weight approach include low entry barriers for new users and low participation costs for the existing user base, and indeed the popularity of collaborative tagging can partly be attributed to the benefits that users perceive in the ease of annotating resources [Mat04]. Additionally, a collaborative tagging system typically demands only a minimal set of requirements from new users joining the system, for example the provision of a valid email address for registering the user account. Hence, the user population is often very diverse and comprised of people from many different backgrounds. For all these reasons, folksonomies are also believed to allow for higher flexibility and faster adaptation to changes and emerging trends than

[13] To mitigate this problem, search engines nowadays include text snippets or other contextual information about a Web resource in search results, and infer implicit positive or negative user feedback on Web resources by analyzing user click patterns in search results.

more strict approaches that require *a priori* coordination and negotiation among their actors.

Apart from containing information about Web resources, folksonomies are also rich sources of data about Web users. The tripartite structure of folksonomies, which we will describe in detail in Chapter 2, means that tags serve a dual purpose as they can be used to infer information about both users (interests in topics) and resources (topics, aboutness). As such, tags also act as the intermediary element in folksonomies that can relate users of similar interests to resources of similar topics.

Finally, folksonomies share a striking similarity with the Web. Manning et al. commented on the Web in 2008 [MRS08]:

> "The Web is unprecedented in many ways: unprecedented in scale, unprecedented in the almost-complete lack of coordination in its creation, and unprecedented in the diversity of backgrounds and motives of its participants."

As we have described above, folksonomies have similar characteristics. Particularly, both feature a simple, local action that with increasing scale and usage eventually leads to a huge, complex network structure. On the Web, users can freely create hyperlinks from one Web resource to another (*resource* $\leftrightarrow$ *resource*) without conforming to any predefined rule set or joint strategy. Similarly, users in folksonomies can freely assign descriptive labels to Web resources (*user* $\leftrightarrow$ *tag* $\leftrightarrow$ *resource*). While at first glance these scenarios might result in unorganized, random or chaotic structures, the relations between the involved entities are in fact not arbitrary in practice. For example, folksonomies exhibit power law and scale-free behavior – the standard signature of self-organization and human activity – just like the Web itself [BA99, BAJ00]. Likewise, they show strong correlations with external trends[14] [WZB08].

For all these reasons, folksonomies and collaborative tagging have recently attracted the attention of the scientific community because they provide a lot of research opportunities in a variety of areas such as enhancing the Semantic Web [WZM06, Kne06, Mik07], improving recommendation systems [SGMB08, WUS09], extending Web search [BXW+07, NM07b, AGS08a, HKGM08, DMQU10] and computational linguistics [AKD07, XBCY07, CBHS08]. In Chapter 2, we will present a thorough review of folksonomies and collaborative tagging. In the following section, we will focus on the relation of folksonomies to Web information retrieval.

1.3 From Folksonomies to Information Retrieval on the Web

For thousands of years people have realized the importance of archiving and finding information. The practice of archiving written information can be traced back to around 3000 BC, when the Sumerians designated special areas to store clay tablets with

[14] A popular Web trend report is the illustrative annual *Google Zeitgeist*, available at http://www.google.com/intl/en/press/zeitgeist/index.html.

cuneiform inscriptions. Even then the Sumerians realized that proper organization and access to the archives was critical for efficient use of information. They developed special classifications to identify every tablet and its content. Fast forwarding to more recent times, librarians have relied on tools such as the *Dewey Decimal Classification system* [OCL], which is used to classify and index books according to a fixed categorization scheme. With the subsequent advent of computers and the Web, it became possible to store and access large amounts of information – and finding useful information from such collections became a necessity. The scientific field of ***Information Retrieval (IR)*** was born out of this necessity [Sin01]. The meaning of the term information retrieval can be very broad. Just getting a credit card out of your wallet so that you can type in the card number is a form of information retrieval. However, as an academic field of study, information retrieval might be defined thus [MRS08]:

> "Information retrieval is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."

Web information retrieval in particular focuses on the *search and retrieval* of relevant and high quality resources from the Web, and on how the most relevant resources can be presented to users first through *ranking* techniques [Cha03, MRS08]. It also covers supporting users in *browsing* (navigating) or *filtering* resource collections or further processing a set of retrieved resources.

The advent of folksonomies and collaborative tagging on the Social Web, on the other hand, has resulted in large volumes of user-contributed annotations of Web resources. A recent study reports that about one third of Web users has actively participated in tagging activities already [Rai07]. Consequently, the question arises how this new kind of information can be analyzed, understood and exploited to extend and improve information retrieval on the Web.

Traditionally, Web information retrieval has relied on approaches and techniques that extract data from Web resources directly (e.g. by examining the textual content of a Web document, similar to classic information retrieval), that analyze Web-specific features (e.g. the link structure of the Web graph), or that are based on an analysis of the metadata about Web resources as provided by their authors (e.g. `META DESCRIPTION` information, see below) [CDI98, Bro02, Kan04, KZ04]. The first, content-based approach has benefited from well-researched techniques in classic information retrieval such as *TF-IDF*[15] but still suffers from the difficulties of automatically inspecting and understanding non-textual Web content such as images and videos, and even textual data is not trivial to analyze given the huge amount and variety of content on the Web. The second approach has resulted in several improvements to and advancements in Web information retrieval such as the *PageRank* [BP98] and *HITS* [Kle98] algorithms. The third approach relies on optional metadata of Web resources that is generally specified

[15] Term Frequency–Inverse Document Frequency (TF-IDF) is a weight often used in information retrieval [SB88]. This weight is a statistical measure used to evaluate how important a word (term) is to a document in a collection or corpus.

in the META element as defined in the HTML and XHTML standards[16]. The META element allows the author of a Web resource to manually specify metadata such as a short DESCRIPTION of the document, the DATE of creation or modification, and KEYWORDS for facilitating the retrieval and analysis of the resource. For example, this metadata may be used by a search engine to improve the quality of search results.

As we have discussed in the previous section, tags are similarly used to describe Web resources in many different ways. As such, they can be considered as a form of metadata for those Web resources to which tags have been assigned by users. But most interestingly for exploiting folksonomies for Web information retrieval, there is a significant difference between tags and traditional Web metadata: Tags are provided by the readers or recipients of Web content, whereas traditional metadata is specified by the authors or publishers. Hence, folksonomies reflect the viewpoints of end users and their perspective on Web content. The information provided by folksonomies is therefore believed to be different from traditional data and metadata on the Web because tags represent the judgements of end users of what a Web resource is about, i.e. its topics or "aboutness". For example, user-contributed tags have been found [LGZ08] to be more appropriate to capture the aboutness of Web resources than automated techniques such as TF-IDF that extract data directly from the content of Web resources and thus rely on input data provided by the authors of these resources (see illustration in Table 1.1).

Web resource	http://ka1fsb.home.att.net/resolve.html
Top tags	linux, dns, networking, howto, sysadmin
Top TF-IDF terms	ampr, domain, jnos, nameserver, conf

Table 1.1: **Man versus machine.** An example of user-generated *tags* and machine-generated TF-IDF terms extracted from the content of a Web resource based on [LGZ08], for which we retrieved updated tag information from the social bookmarking service Delicious in February 2010. The Web document in question is a brief introduction to the /etc/resolv.conf file on Linux operating systems that is used for networking and DNS configuration. At the time of writing, the document was tagged by ten users, the first time in July 2005 and most recently in January 2010. The top tags linux and dns were assigned to the document by seven and six of these ten users, respectively.

We therefore believe that folksonomies provide new, complimentary information that can be exploited for enhancing and improving Web information retrieval. However, several important questions must be answered first, particularly with regard to the quantity and quality of data available in folksonomies in practice. For instance, is tagging information abundant or sparse? What kind of tags and what kind of Web resources are selected by users for annotation? How does information provided by folksonomies compare to other types of Web data and metadata? In the following section,

[16] XHTML2 Working Group Home Page at W3C, http://www.w3.org/MarkUp/, last retrieved on March 01, 2010.

we will describe several research questions in relation to understanding and leveraging folksonomies for Web information retrieval that we will investigate in this thesis.

1.4 Motivation and Scope of the Thesis

This thesis focuses on the activities of users and their contributed data on the Social Web with an emphasis on folksonomies and collaborative tagging, and how this information can be leveraged for Web information retrieval. Folksonomies and collaborative tagging have been popular social phenomenons on the Web in recent years, and therefore they have attracted the attention of researchers from a wide range of domains, including for example library science, media studies, social sciences, and of course, computer science. Although we are studying the dynamics and collective social behavior of human users in folksonomies, we approach the issue from the perspective of computer science and mathematics[17].

1.4.1 Research Questions

By conducting the research described in this thesis, we aim at answering the following research questions:

> **Research Question 1 (Understanding Folksonomies)**:
> How can we understand the meanings of and extract information from the user-contributed data in folksonomies for Web information retrieval?
>
> **Research Question 2 (Leveraging Folksonomies)**:
> How can we leverage this knowledge in order to create and improve new applications in Web information retrieval for the benefit of users?

While these two questions are rather general, they are central to this thesis. Firstly, researchers are still in the process of discovering the characteristics and dynamics of folksonomies and collaborative tagging systems as well as the hidden semantics of the data available in these systems as we will see in Chapter 2. Before we can make full use of folksonomies, we need to find out, for example, how much and what kind of user-contributed data is available in folksonomies, and how it relates to other types of data and metadata in the domain of Web information retrieval. Secondly, we are interested in whether and how folksonomies can be exploited for the benefits of their users. For example, can we leverage our knowledge of folksonomies to improve the user experience in collaborative tagging systems or even in other domains of information retrieval such as Web search? Answering these questions would allow us to gain a better understanding of the Social Web, to improve applications for the Social Web and for Web

[17] Albert-László Barabási, well-known for his work on scale-free networks [BA99], has described his mathematical studies of usage patterns in user communities and social networks as "computational social science" [Sti09].

information retrieval, and, by doing so, to contribute to advancing the current state of the World Wide Web at large.

In the research work described in this thesis, we will focus our analyses with regard to Web resources on *Web documents* (also called "Web pages"), i.e. resources with mainly *textual* content. The main reason is that it allows us to compare the content of the resource with other Web data and metadata such as tags in a folksonomy or anchor text of incoming hyperlinks of the resource, which would be very difficult or even impossible to achieve on a large scale for non-textual resource types such as images or videos[18].

1.4.2 Hypotheses

The first research question particularly involves exploratory studies of folksonomies. Some of the characteristics of folksonomies have yet to be discovered, particularly when trying to relate and integrate folksonomies into Web information retrieval. We believe folksonomies are a valuable source of information for Web information retrieval tasks. Compared to, for example, automated data extraction techniques such as TF-IDF, folksonomies are driven by the best data processor available – the human brain. Users can easily process and understand even complicated content and media types on the Web such as images or videos that still pose a challenge even for specialized computer algorithms. However, we will see in Chapter 2 that one defining feature of collaborative tagging, and arguably one important reason for its popularity and success in practice, is the freedom it gives to users in terms of how and which resources can be tagged. This means that we need to explore the dynamics of user behavior in collaborative tagging systems before we can deduce meaningful information from the data of the folksonomies that emerge from these systems. We believe that folksonomies provide a sufficient amount of information about Web resources in order to be used as the data base (no pun intended) on which other applications can be built. Particularly, we want to study how much data about Web resources is actually provided by users in folksonomies, and how it compares to traditional types of Web metadata. For example, are users focusing their tagging activities on resources that are already popular on the Web, or do they prefer to discover hidden "gems" on the Web? In summary, we test the following hypothesis regarding the data contributed by users in folksonomies:

> **Hypothesis 1 (New Perspective on the Web)**:
> User-contributed data in folksonomies provides new, complimentary information about Web resources that is not available through traditional types of data and metadata on the Web, such as metadata contributed by the authors of these resources.

[18] The *ESP game* (a collaborative tagging system described in Section 2.1.2) has collected a large volume of user-contributed *image* annotations. However, the ESP game makes use of so-called *taboo words* which players are not allowed to use for tagging a presented image [vAD04]. These words will usually be related to the image and make the game harder because they can be words that players commonly use as guesses. Taboo words thus represent an artificial input filter for user-contributed tags – particularly those tags that most users might choose if their was no such filter in place – that inhibits the use of ESP data sets for the research work described in this thesis.

Secondly, we will study the users of folksonomies in particular. The users are the actors in collaborative tagging systems and the sole source of data contributed to these systems. This means they are the primary factor that determines the quality of information in a folksonomy. Hence, an important question is to what extent we can rely on the inputs provided by a user. Many collaborative tagging systems in practice, particularly those that enjoy high popularity among users, demand only a minimal set of requirements from new users joining the system, for example the provision of a valid email address for registering the user account. Additionally, they allow anonymous user accounts where users can act under a pseudonymous username, i.e. the user identity is neither asked for nor verified. This means that there exists no *a priori* knowledge of the "credibility", "expertise" or "trustworthiness" of a user in a folksonomy. However, we believe that the expertise or trustworthiness of users can be understood by an analysis of their activity and implicit interactions in a folksonomy. We simultaneously believe that such an analysis can also reduce and mitigate the impact of spamming activity on a folksonomy. Additionally, such an approach could eventually also be helpful to other techniques and methodologies in Web information retrieval, for example by pre-processing or refining the input information derived from folksonomies before it is passed to the respective algorithms. In summary, we test the following hypothesis regarding users in folksonomies:

> **Hypothesis 2 (User Expertise)**:
> The expertise or trustworthiness of users in a folksonomy can be derived from an analysis of their activity and implicit interactions within the folksonomy.

Thirdly, we will study how to exploit the information about users and Web resources in a folksonomy. When users annotate Web resources with tags, they do not only provide information about the Web resources, they also provide information about themselves. As we will see in Chapter 2, the use case for tagging a resource is similar to the traditional notion of bookmarking a Web resource [ABC98]: Tagging represents an explicit user action that happens *a posteriori* because resources are manually tagged *after* being "processed" by users. Similarly, users tag resources that they are in one or the other way interested in, and when the "cost" of tagging yields a subjectively perceived benefit, for example a better future retrieval of the tagged resource. We therefore believe that folksonomy data provides relevant information that can be exploited for profiling both users and resources. For the former, tags can be used to model the interests of users in certain topics. For the later, tags can be used to model the topic(s) that a resource is about. In other words, we want to exploit the intermediary function of tags to relate users of similar interests to resources of similar topics. By developing a method to re-rank search results based on such user and resource profiles, we believe that folksonomies can thus be exploited for the personalization of Web search. In summary, we test the following hypothesis regarding information about users and Web resources in folksonomies:

Hypothesis 3 (Web Search Personalization):
Folksonomies provide sufficiently rich information about users and Web resources to allow for the personalization of Web search, i.e. an individualized search for resources on the Web.

Finally, we will investigate how the concept of collaborative tagging combined with the popularity of folksonomies among users can be leveraged for the scenario of Web filtering. Similarly to the descriptions of *Hypothesis 3 (Web Search Personalization)* above, user and resource profiles derived from folksonomies can serve as a starting point for developing a user-driven content filtering application of the Web, i.e. allowing or blocking access to Web resources based on the feedback of end users on these resources. While automated approaches to understanding Web content would generally be better suited than manual approaches to cope with the large scale and rapid growth of the Web, machines still cannot compete with the content processing capabilities and accuracy of humans for tasks such as Web page classification or detecting nudity in photographic images. As we will describe in Chapter 7, existing Web filtering approaches based on human input – so-called Internet content rating systems – suffer from scalability and acceptance problems in practice. However, based on the recent scientific findings on the dynamics of collaborative tagging and popularity of folksonomies in practice, we believe that folksonomies can be leveraged for a new approach to Web filtering that is based on human feedback on Web content. In summary, we test the following hypothesis regarding collaborative tagging and folksonomies:

Hypothesis 4 (Web Filtering):
The concepts of folksonomies and collaborative tagging can be exploited for user-driven filtering of the Web, i.e. allowing or blocking access to Web resources based on human input.

These four hypotheses and our corresponding research work investigate folksonomies from different angles and for different problem scenarios while still being very closely related to each other. All of these studies are highly relevant to the central theme of this thesis and to answering the research questions presented at the beginning of this section: How can we improve our understanding of folksonomies, and how can we derive value from this knowledge for the benefit of the users in the context of Web information retrieval? Additionally, the variety of our studies is also an indication of the versatility and applicability of the information provided by folksonomies on the Web.

1.5 Contributions and Publications

The focus of this thesis lies in the analysis of folksonomies and collaborative tagging in the context of information retrieval on the Web. It presents empirical studies of folksonomies and demonstrates how their results can be leveraged to enhance and improve techniques in the research domain. The contributions of this thesis include the followings:

- We present a comprehensive review of prior research and studies on folksonomies and collaborative tagging, and discuss related concepts such as manual subject indexing and ontologies (Chapter 2).
- We construct several large-scale corpuses of experimental data from a variety of sources such as the collaborative tagging system *Delicious*, the Web taxonomy of the *Open Directory Project*, the search engine *Google* and the Web itself for studying the dynamics and characteristics of folksonomies in the context of Web information retrieval (Chapter 3).
- We investigate the characteristics, dynamics and hidden semantics of folksonomies in various dimensions. We analyze and compare user-contributed metadata about Web resources from folksonomies with the contents of these resources and other types of Web metadata, for example information provided by the authors of resources, or search keywords in queries of users searching the Web (Chapter 4).
- We discuss the notion of expertise or "trustworthiness" of users in folksonomies. We propose an algorithm, *SPEAR*, that implements the idea of ranking users according to their expertise, and evaluate it in terms of its effectiveness of ranking experts and being resistant to malicious spamming activities (Chapter 5).
- We study how the information in folksonomies can be leveraged for personalizing Web search and demonstrate how it can be implemented in practice. We describe our experiments on evaluating the approach and present an analysis of our experimental results. (Chapter 6).
- We explore how the concepts of collaborative tagging and folksonomies can be exploited for user-driven filtering of the Web, i.e. allowing or blocking access to Web resources based on human input. We will present a case study of a working prototype, *TaggyBear*, and describe and evaluate its system design and anatomy (Chapter 7).

In addition, earlier versions of several different parts of this thesis have been published and presented at international scientific conferences in the past few years. Similarly, the author of this thesis wrote an invited article for the Internet company *Yahoo!*[19]

[19] Yahoo!, http://www.yahoo.com/.

about his work on identifying domain experts within their social bookmarking service *Delicious*[20]. The research work described in Chapter 7 has resulted in a European and international patent application[21], filed in 2008 together with the industrial partner SES ASTRA S.A.

The full list of publications include the followings:

- Ching-man Au Yeung, Michael G. Noll, Nicholas Gibbins, Christoph Meinel, Nigel Shadbolt. *SPEAR: Spamming-resistant Expertise Analysis and Ranking in Collaborative Tagging Systems*. International Journal of Computational Intelligence, Wiley-Blackwell, 2010 (to appear) [ANG+10].
- Michael G. Noll, Ching-man Au Yeung. *How SPEAR Identifies Domain Experts within Delicious*. Invited Article for Yahoo!, 2009 [NA09].
- Michael G. Noll, Ching-man Au Yeung, Nicholas Gibbins, Christoph Meinel, Nigel Shadbolt. *Telling Experts from Spammers: Expertise Ranking in Folksonomies*. SIGIR '09: Proceedings of 32nd ACM Special Interest Group on Information Retrieval Conference, USA, 2009, ISBN 978-1-60558-483-6 [NAG+09].
- Ching-man Au Yeung, Michael G. Noll, Nicholas Gibbins, Christoph Meinel, Nigel Shadbolt. *On Measuring Expertise in Collaborative Tagging Systems*. WebSci '09: Proceedings of 1st Web Science Conference '09, Greece, 2009 [ANG+09].
- Michael G. Noll. *Writing a Personal Link Recommendation Engine*. Python Magazine, Volume 3, Issue 2, 2009, ISSN 1913-6714 [Nol09].
- Michael G. Noll, Christoph Meinel. *The Metadata Triumvirate: Social Annotations, Anchor Texts and Search Queries*, WI '09: Proceedings of 7th IEEE/WIC/ACM International Conference on Web Intelligence. IEEE CS Press, Australia, 2008, ISBN 978-0-7695-3496-1 [NM08c]
- Michael G. Noll, Christoph Meinel. *Building a Scalable Collaborative Web Filter with Free and Open Source Software*. SITIS '08: Proceedings of 4th IEEE International Conference on Signal-Image Technology & Internet-based Systems, IEEE CS Press, Indonesia, 2008, ISBN 978-0-7695-3493-0 [NM08a].
- Michael G. Noll, Christoph Meinel. *Exploring Social Annotations for Web Document Classification*. SAC '08: Proceedings of 23rd International ACM Symposium on Applied Computing, Brazil, 2008, ISBN 978-1-59593-753-7 [NM08b].
- Michael G. Noll, Christoph Meinel. *Web Search Personalization via Social Bookmarking and Tagging*. ISWC '07: Proceedings of 6th International Semantic Web Conference & 2nd Asian Semantic Web Conference, Springer LNCS 4825, South Korea, 2007, ISBN 978-3-540-76297-3 [NM07b].

[20] Delicious, http://www.delicious.com/, a Yahoo! company.

[21] Patent application "Method for controlling the transfer of data entities from a server unit on a communication channel" (International Application Number: PCT/EP2008/067735), filed on December 17, 2008.

- Michael G. Noll, Christoph Meinel. *Authors vs. Readers: A Comparative Study of Document Metadata and Content in the WWW*. DocEng '07: Proceedings of 7th International ACM Symposium on Document Engineering, Canada, 2007, ISBN 978-1-59593-776-6 [NM07a].
- Michael G. Noll, Christoph Meinel. *Design and Anatomy of a Social Web Filtering Service*. CIC '06: Proceedings of 4th International Conference on Cooperative Internet Computing, Hong Kong, 2006, ISBN 978-981-281-109-7 [NM06].
- Michael G. Noll, Christoph Meinel. *Web Page Classification: An Exploratory Study of Internet Content Rating Systems*. HACK '05: Proceedings of 1st HACK Conference, Luxembourg, 2005, ISBN 978-2-9599708-0-1 [NM05].

1.6 Overview of the Thesis

The thesis is structured as follows. After the introduction presented in this chapter, **Part I** will focus on the *understanding* of folksonomies and collaborative tagging in the context of Web information retrieval. Chapter 2 will present a literature review, which will include formal models of folksonomies, a detailed description of their characteristics and dynamics as well as a comprehensive review of state-of-the-art research on folksonomies and collaborative tagging. Chapter 3 will describe the major experimental data sources used in this thesis and discuss why they are suitable targets for our research. It will introduce the technical tools that we have created and used to collect data from these sources, and give an overall description of the main experimental data sets that we have subsequently constructed for our studies presented in the later chapters. Chapter 4 will present our empirical and explorative studies of the information and hidden semantics of folksonomies in the context of Web information retrieval. It will describe our analyses and comparisons of metadata in folksonomies about Web resources with the contents of these resources and other types of Web metadata.

In **Part II** of the thesis, we will proceed by *leveraging* the knowledge and experimental results presented in the first part for enhancing and improving Web information retrieval. Chapter 5 will investigate the notion of expertise or "trustworthiness" of users in folksonomies and present our proposed algorithm, *SPEAR*, for ranking users by their expertise. It will describe our experiments on evaluating the algorithm in terms of its effectiveness of ranking experts and being resistant to malicious spamming activities. Chapter 6 will present our approach to personalization of Web search by exploiting folksonomies for profiling of users and Web resources. It will demonstrate how the approach can be implemented in practice and present our analysis of the experimental results. Chapter 7 will explore how the concepts of collaborative tagging and folksonomies can be exploited for Web filtering. It will present a case study of a working prototype, *TaggyBear*, and describe and evaluate its system design and anatomy.

Part III will conclude the thesis with summarizing its key results. It will discuss the implications and significance of our research and outline possible research directions for the future.

Part I

Understanding Folksonomies

I can calculate the motion of heavenly bodies, but not the madness of people.

Sir Isaac Newton (1643–1727)

2 A Review of Folksonomies

Folksonomies are a new form of classification scheme that emerges from the collective tagging activities of people on the Web, in which they organize and categorize Web resources through annotations with freely chosen keywords called "tags". The collaborative aspect lies in publicly sharing the information of one's individual tagging activities with other users, which in practice is carried out on Web-based platforms called collaborative tagging systems. The popularity of such systems and their derived folksonomies among Web users has already come to the point that the size of some user communities have exceeded the populations of countries such as Australia or Switzerland. As a result, folksonomies provide access to large amounts of user-contributed data that scientists have started to analyze for understanding the characteristics of tagging systems and the dynamics of user behavior, and for leveraging this knowledge in order to create new applications and services based on information extracted from folksonomies in a variety of domains.

In this chapter, we present a comprehensive review of folksonomies and collaborative tagging, describe formal models and summarize the results of prior research and scientific studies on the topic. Due to the scope of this thesis as described in Section 1.4, we mainly focus on the literature of computer science here, and refer to studies in other research domains where necessary and appropriate.

2.1 Collaborative Tagging

2.1.1 Overview

As we have noted in the introduction to this chapter, ***tagging*** is the act of manually annotating resources with an unstructured list of keywords or phrases called tags that are created or selected on the fly by human users [Mat04, HJSS06a]. The ***collaborative*** aspect lies in publicly sharing the information of one's individual tagging activities with other users. The technical, Web-based platforms that allow for such collaborative tagging are called ***collaborative tagging systems***. We have compiled an exemplary list of popular collaborative tagging systems in Section 2.1.2. Figure 2.1 illustrates the basic concept of collaborative tagging.

Since collaborative tagging involves human users assigning "labels" to resources, it can be considered as a form of manual subject indexing, which we describe in more de-

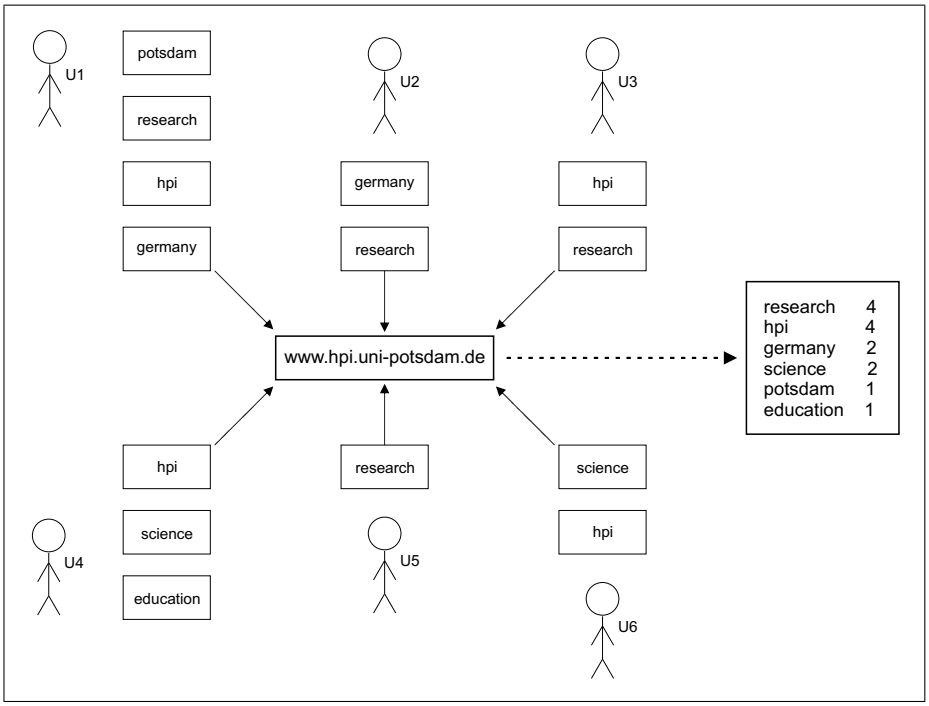


Figure 2.1: **Illustration of the concept of collaborative tagging.** In this example, six users *U1-U6* have annotated the homepage of the Hasso Plattner Institute (www.hpi.uni-potsdam.de) with tags such as `research` and `hpi`.

tail together with other related concepts in Section 2.3. However, collaborative tagging features some significant differences compared to this traditional approach of annotating resources.

- **Lack of a common strategy or goal**: In contrast to strict classification systems such as the *Dewey Decimal Classification system* [OCL], which is used by librarians to classify and index books according to a fixed categorization scheme, collaborative tagging in general does neither require nor expect a joint strategy or goal that all participants have agreed upon (*a priori*) or need to follow (*a posteriori*). Each user in a collaborative tagging system can freely act as he pleases. One effect is that users have different incentives and motivations for joining and participating in a collaborative tagging system as we will examine in Section 2.4.3.

- **Lack of a controlled vocabulary**: Users need neither to have knowledge of nor to conform to a predefined, controlled vocabulary for indexing resources. Instead, the tagging vocabulary of a folksonomy organically grows from the open-ended, collective tagging activities of its users. The advantages of such a light-weight approach include low entry barriers for new users and low participation costs for the existing user base. And indeed, the popularity of collaborative tagging can partly be attributed to the benefits that users perceive in the ease of annotating

resources [Mat04]. Additionally, it leads to higher flexibility and fast adaptation of a collaborative tagging system to changes and emerging trends with regard to its users and their vocabulary, its resources and its environment in general.

- **Powered by end users**: In traditional subject indexing, index terms are generally chosen by the creators or owners of a resource (e.g. keywords in HTML metadata specified by the author of a Web document), or expert users with specific domain knowledge (e.g. by a librarian in the case of books). Collaborative tagging however is performed by end users, i.e. the recipients or consumers of resources (e.g. by the reader of a Web document or book). As a result, the former concept involves a rather small set of actors, whereas the latter is driven by masses of people, hence the often-cited reference to "wisdom of the crowd" [Sur05]. We will study the effect of this difference between the two groups in Chapter 4 by comparing and analyzing data provided by authors and readers of Web documents. Similarly, a collaborative tagging system typically demands only a minimal set of requirements from new users joining the system, for example the provision of a valid email address for registering the user account. Hence, the user population is often very diverse and comprised of people from many different backgrounds. It may also range from absolute laymen to domain experts.

As more and more users participate in and contribute to a collaborative tagging system, a new form of classification scheme emerges. This result of collaborative tagging is now commonly referred to as a ***folksonomy***. We will discuss folksonomies in more detail later in this chapter.

2.1.2 Example Collaborative Tagging Systems

In this section, we provide a brief description of collaborative tagging systems in practice. There are many others in existence but we have chosen seven that are representative of the diversity of those that are currently popular and well used. As of 2010, all these tagging systems are still up and running. Where publicly available, we added statistical information to give an idea about their scale. Please keep in mind when reading the numbers below that these systems have existed only for a few years.

- **CiteULike (*2004)**[1]: A service for managing and tagging citations and references, for example academic papers or journals.
  210,000 users managing 3.4 million articles from 2004-2009[2]

- **Delicious (*2003)**[3]: A social bookmarking service allowing users to save and tag bookmarks of Web pages and other Web resources, and to share this information

[1] CiteULike, http://www.citeulike.org/.
[2] Statistics based on figures from the CiteULike home page, http://www.citeulike.org/, and a message by Kevin Emamy of CiteULike, http://www.citeulike.org/groupforum/1784, last retrieved on January 15, 2010.
[3] Delicious, http://www.delicious.com/.

with other users. In this thesis, the folksonomy of Delicious is a major source of experimental data, which we describe in more detail in Section 3.1.1.
5.3 million users managing 180 million unique Web resources from 2003-2008[4]

- **ESP game (*2003)**[5]: An online tagging game where users are randomly paired with each other, and try to guess tags the other would use when presented with a random image [vAD04].
100 million images tagged from 2003-2008[6]

- **Flickr (*2004)**[7]: A photo sharing service allowing users to store and tag their personal photos, and to share this information with other users. Users can maintain a network of contacts, join groups and tag photos of other users.
4 billion photos uploaded by users from 2004-2009[8]

- **Last.fm (*2002)**[9]: A music service allowing users to discover new music based on their past listening preferences. Users can tag both songs and artists.
30 million users[10] *in 2009; 36 billion tracks listened to by users from 2002-2009.*

- **LibraryThing (*2005)**[11]: An online service allowing users to save and tag their personal libraries of books, and to share this information with other users [Rit09]. Users can maintain a network of contacts, join groups and discover new books through the libraries of users with similar reading preferences.
1 million users, 45 million cataloged books (5 million unique works), and 50 million tag assignments from 2005-2009[12]

- **YouTube (*2005)**[13]: A video sharing service allowing users to upload video content and annotate it with tags.
Third most popular site of the Web[14]*, with 100 million viewers watching 6.3 billion videos in 2009*[15]

[4] Delicious announcement, http://blog.delicious.com/blog/2008/11/delicious-is-5.html, last retrieved on March 01, 2010.
[5] ESP game, http://www.espgame.org/.
[6] BBC news article, May 14, 2008; http://news.bbc.co.uk/2/hi/technology/7395751.stm, last retrieved on March 01, 2010.
[7] Flickr, http://www.flickr.com/.
[8] Flickr blog post, http://blog.flickr.net/en/2009/10/12/4000000000/, last retrieved on March 01, 2010.
[9] Last.fm, http://www.last.fm/.
[10] Announcement by Last.fm, http://blog.last.fm/2009/03/24/lastfm-radio-announcement, last retrieved on March 01, 2010
[11] LibraryThing, http://www.librarything.com/.
[12] LibraryThing Zeitgeist statistics, http://www.librarything.com/zeitgeist, last retrieved on March 01, 2010.
[13] YouTube, http://www.youtube.com/.
[14] Alexa Rank statistics ranks YouTube right after the websites Google.com and Facebook.com. http://www.alexa.com/siteinfo/youtube.com, last retrieved on March 01, 2010.
[15] ComScore press release, March 4, 2009; http://www.comscore.com/Press_Events/Press_Releases/2009/3/YouTube_Surpasses_100_Million_US_Viewers, last retrieved on March 01, 2010.

2.1.3 Design Dimensions

While the various collaborative tagging systems in practice follow the same general approach to collaborative tagging, there are design differences that have an impact on the resulting folksonomies [Wal05, MNBD06].

Marlow et al. [MNBD06] developed two tagging taxonomies to analyze how characteristics of system design and user incentives may influence the resultant tags in tagging systems. They identified the following key design dimensions of tagging systems that may have immediate and significant effect on the content and usefulness of tags in the system.

- **Tagging rights**: Who is allowed to tag resources? For example, tagging may be restricted to *self-tagging* (resources may be tagged only by their owners), *free-for-all-tagging* (any user can tag any resource) or a more granular, permission-based mixture of both extremes. According to Marlow et al., tagging rights is arguably the most important design dimension.

- **Tag aggregation**: Whether and how tags of a given resource are aggregated. The main distinction is whether the frequency of a tag applied to the resource is recorded (*bag model*, e.g. Delicious) or not (*set model*, e.g. YouTube and Flickr). In the former case of a bag model, each user may maintain his own set of tags per resource.

- **Tagging support**: Whether and how the tagging system supports users in selecting tags, most often in terms of the user interface. In the case of *blind tagging*, a user cannot view tags assigned to the same resource by other users while tagging himself. The opposite approach is *viewable tagging*, where a user can indeed view other users' tags of the same resource. In the case of *suggestive tagging*, the system actively suggests or recommends tags to the user. For example, these suggestions may be based on the user's personomy $\mathcal{P}_u$ or his tagging vocabulary $\mathcal{T}_u$, or be based on tags assigned to the resource by other users, i.e. the restriction of the folksonomy $\mathcal{F}$ to the given resource $r \in \mathcal{R}$[16].

- **Type of object**: The type of resource being tagged. For example, resources may be Web documents, images, or videos. See also Section 2.1.2. As Marlow et al. note, any object that can be virtually represented can be tagged or used in a tagging system.

- **Source of material**: The origin of the resources. Resources to be tagged can be supplied by the users (e.g. photos on Flickr, or videos on YouTube), by the system (e.g. images in the ESP game [vAD04]), or be publicly available (e.g. Web documents tagged on Delicious).

[16] Personomies, tagging vocabularies and other important terms in the context of folksonomies are defined in the next section on formal models.

- **Resource connectivity**: Whether resources can be linked to each other independent of tags. The main connectivity categories are *linked*, *grouped*, and *none*. For example, Web documents are connected by directed hyperlinks (*linked*), photos on Flickr can be assigned to photo pools (*grouped*).

- **Social connectivity**: Whether users can be linked to each other independent of tags, i.e. whether users can directly interact in one way or the other.

Collaborative tagging systems also differ on dimensions other than those described above, for example in terms of features that support users in their tagging activities (e.g. tag suggestions) or the availability of application programming interfaces (API). Some tagging systems also extend the general definition of folksonomies described in Section 2.2 to allow for additional functionality such as user-configurable tagging permissions for resources (photo sharing service Flickr) and hierarchical relations between tags (BibSonomy [HJSS06a]).

2.2 Formal Models of Folksonomies

As more and more users participate and contribute to a collaborative tagging system, a new form of classification scheme emerges. This result of collaborative tagging is now commonly referred to as a ***folksonomy*** [Smi04]. In the following sections, we present a thorough review of folksonomies and summarize the result of recent research. We start our discussions with definitions and a formal model of folksonomies.

2.2.1 Definitions

A folksonomy is in general comprised of three different sets of entities: users, tags, and resources [Mat04, Mik05, MNBD06].

- **Users** (set symbol $\mathcal{U}$)
  Users assign tags to resources and are thus the active element in collaborative tagging systems and folksonomies. In other words, users provide the actual input for such systems on which subsequent analyses are based – that's why we also speak of "user-generated content" in this context. The set of users is often called the *user community*. Depending on the system and its features, the actions and interactions of users may form implicit or explicit user groups and social networks.

- **Tags** (set symbol $\mathcal{T}$)
  Tags are keywords freely chosen by users to annotate resources, may it be for describing the content of Web resources (e.g. `article`), categorizing resources (e.g. `news`), expressing personal opinions (e.g. `funny`) or for other purposes. While we focus our description of folksonomies in this work on tags in the form

of single textual terms, tags can also be phrases or combinations of any kind of symbols and alphabets depending on the context of the system.[17]

- **Resources** (set symbol $\mathcal{R}$)
  Resources represent the items which are being annotated by users with tags, or simply *tagged* in short. The kind of resources depends on the actual system and may vary widely in practice. For example, users may tag books (e.g. LibraryThing), music (e.g. Last.fm) photos (e.g. Flickr), videos (e.g. YouTube) or Web resources in general (e.g. Delicious).

The collection of all users, resources and tags plus the assignments of tags to resources by users are called a ***folksonomy***. In this work, a folksonomy $\mathcal{F}$ is formally defined as follows (cf. [HJSS06a, DS08, MCM+09b]):

Definition 2.2-1 (Folksonomy). A folksonomy is a quadruple $\mathcal{F} := (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y})$, where $\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$ are finite sets whose elements are called *users*, *tags* and *resources*, respectively. $\mathcal{Y}$ is a ternary relation[18] between these sets, i.e. $\mathcal{Y} \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{R}$, called *tag assignments*.

An equivalent view of the structure $\mathcal{F}$ in Definition 2.2-1 is that of a 3-partite, 3-regular hypergraph $G = (V, E)$, in which the node set V is partitioned into three disjoint sets $V = \mathcal{U} \cup \mathcal{T} \cup \mathcal{R}$, and $E = \{ \{u, t, r\} \mid (u, t, r) \in \mathcal{Y} \}$ is the set of hyperedges with every hyperedge $\{u, t, r\}$ consisting exactly of one user, one tag, and one resource (and thus having a cardinality of three)[19].

In comparison to this graph model of folksonomies where undirected triadic hyperedges connect three different kinds of entities, the graph model of the World Wide Web [KRR+00] consists of directed binary edges (hyperlinks) which connect resources with resources (Web documents), i.e. only entities of the same kind. The relationship between resources and hyperlinks is a well-researched area, with PageRank [BP98] being one of the most prominent examples of these studies.

Depending on the context, extended variants of Definition 2.2-1 are used in the literature and in existing collaborative tagging systems. For instance, temporal information – e.g. the time when a user assigned a tag to a resource – might be integrated for tasks such as trend detection as shown exemplarily in Definition 2.2-2. Most studies of folksonomies in the literature, however, refer to the basic model comprising just three entities and their relation as described in Definition 2.2-1. Since we focus our studies

[17] For completeness, we may also allow the use of the special tag $t_\emptyset$, the *null tag*. The null tag can be used to create a relation between a user and a resource without requiring the user to specify any tag. The reasoning behind is that some collaborative tagging systems allow users to manage resources also in the absence of tag information. For example, a user of the social bookmarking service Delicious may bookmark a Web resource without specifying any tag. Similarly, a user of the photo sharing service Flickr may upload a picture without providing any tag information. However, we may safely omit the null tag for the context of this thesis without loss of generality.

[18] One way to represent the ternary relation $\mathcal{Y}$ is through tensors as described by Wetzker et al. [WZBA10].

[19] Mika [Mik05] notes that such a graph representation of a folksonomy effectively extends the traditional bipartite model of ontologies (concepts and instances) by incorporating actors (users) into the model. In Formal Concept Analysis [GW99], such data structures are called *triadic context* [LW95].

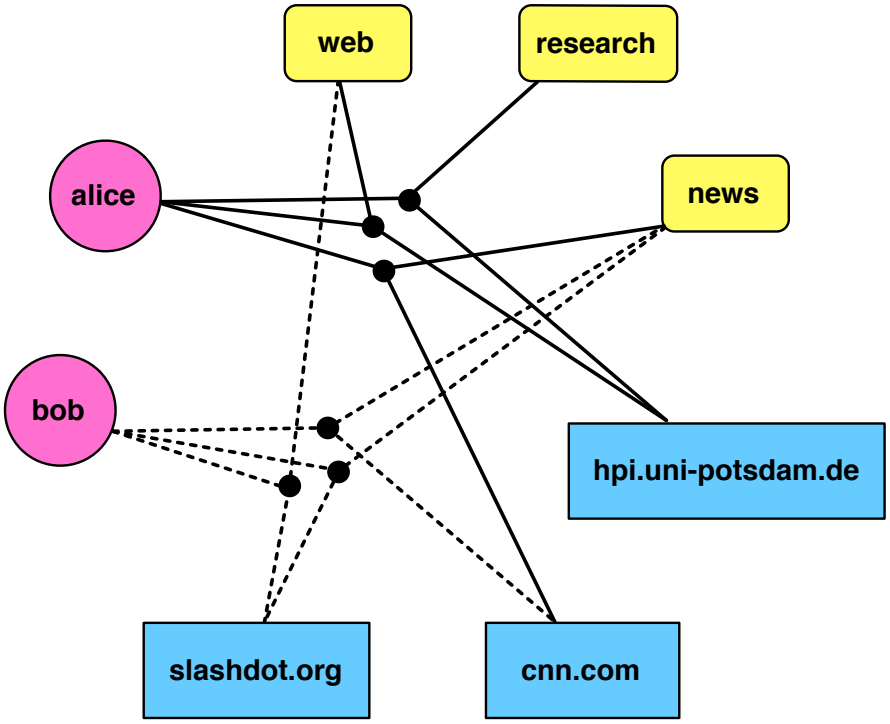


Figure 2.2: **An exemplary folksonomy**, adapted from [MCM+09b]. Two users `alice` and `bob` have annotated three resources (here: the home pages of CNN, the Hasso Plattner Institute and Slashdot) using the three tags `web`, `research` and `news`. The tag assignments (u, t, r) are represented by hyperedges connecting a user, a resource and a tag. These six triples correspond to the following four posts: (`alice`, {`research`, `web`}, `hpi.uni-potsdam.de`), (`alice`, {`news`}, `cnn.com`), (`bob`, {`news`, `web`}, `slashdot.org`) and (`bob`, {`news`}, `cnn.com`) The personomies of `alice` and `bob` comprise the solid and dotted hyper-edges, respectively.

and analyses on these three entities in this thesis as well, we adopt this basic model of folksonomies for the remainder of this work.

Definition 2.2-2 (Example: Extended Folksonomy). A folksonomy that includes temporal information of tagging activity is a quintuple $\mathcal{F}_{temp} := (\mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{Y}, \alpha)$, where α is a function $\alpha : \mathcal{Y} \to c$ which assigns to each tag assignment $y \in \mathcal{Y}$ a temporal marker $c \in \mathbb{N}$. It corresponds to the time at which a user assigned a tag to the resource. $\mathcal{U}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{Y}$ are defined as in Definition 2.2-1.

It is often interesting to filter out all the data of specific users within a folksonomy, the so-called ***personomies*** of these users [HJSS06a]. We can define the personomy $\mathcal{P}_u$ of individual users as follows:

Definition 2.2-3 (Personomy). The *personomy* $\mathcal{P}_u$ of a given user $u \in \mathcal{U}$ is the restric-

tion of $\mathcal{F}$ to u, i.e. $\mathcal{P}_u := (\mathcal{T}_u, \mathcal{R}_u, \mathcal{I}_u)$ with $\mathcal{Y}_u := \{ (t, r) \in \mathcal{T} \times \mathcal{R} \mid (u, t, r) \in \mathcal{Y} \}$, $\mathcal{T}_u := \pi_1(\mathcal{Y}_u)$ and $\mathcal{R}_u := \pi_2(\mathcal{Y}_u)$ where π_i denotes the projection on the i-th dimension.[20]

Definition 2.2-4 (Tagging Vocabulary). The set of tags $\mathcal{T}_u$ in Definition 2.2-3 is called the *tagging vocabulary* of user u.

For convenience in discussing tagging activity, we refer to a user who tagged a resource as *tagger*.

When discussing folksonomies, it is also helpful to group tag assignments into several so-called ***posts*** [HJSS06a, DS08]. A post contains all tag assignments made by the same user to the same resource. In the collaborative tagging system Delicious, for instance, where users create and share bookmarks of Web documents including any associated tags, these "social bookmarks" may be appropriately modeled by such posts[21]. For this reason, we use the terms *post* and *social bookmark* interchangeably throughout this thesis, and only note specific differences where needed. We define the set P of all posts in a collaborative tagging system as follows.

Definition 2.2-5 (Posts). The set $P(\mathcal{F})$ of all *posts* in a folksonomy $\mathcal{F}$ is defined as $P(\mathcal{F}) := \{ (u, \mathcal{Y}_{u,r}, r) \mid u \in \mathcal{U}, r \in \mathcal{R} \}$, where $\mathcal{Y}_{u,r} := \{ t \in \mathcal{T} \mid (t, r) \in \mathcal{Y}_u \}$.

An illustrative example of these definitions is shown in Figure 2.2.

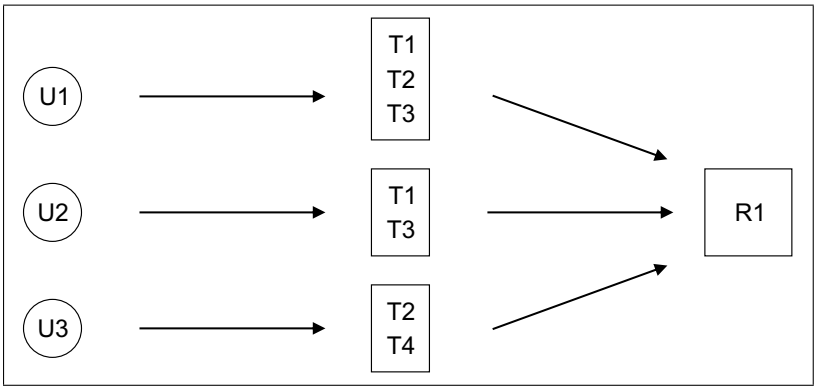
2.2.2 Broad and Narrow Folksonomies

Vander Wal [Wal05] describes the notions of *broad* and *narrow* folksonomies, which form as an effect of the tagging rights and tag aggregation within a collaborative tagging system (see Section 2.1.3). In a tagging system where multiple users can tag the same resource and every user can tag the resource with his own tags, a ***broad folksonomy*** is produced. An example of such a broad folksonomy is the social bookmarking service Delicious, which we describe in more detail in Chapter 3. When only a single, unified set of tags is maintained per resource, a ***narrow folksonomy*** is produced. An example of such a narrow folksonomy is the photo sharing service Flickr. Figure 2.3 illustrates the basic concept and differences of broad and narrow folksonomies.

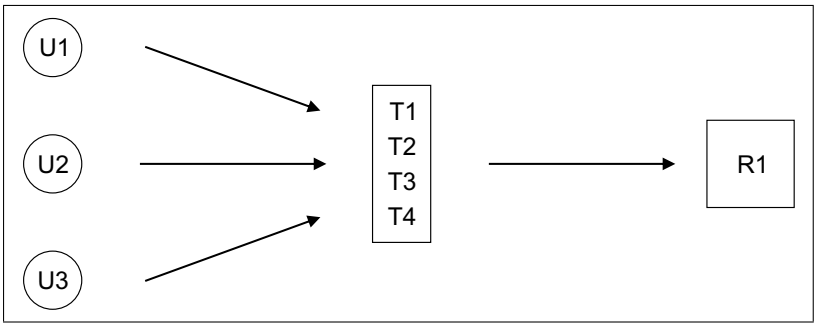
This means that broad folksonomies have a finer data granularity than narrow folksonomies because the latter do not record information such as the *frequency* of tag assignments on a resource or which particular *user* assigned a tag to the resource. Formally speaking, a broad folksonomy keeps track of tagging activity on a resource r via (u, t, r) triples whereas a narrow folksonomy is restricted to an association of r with an unstructured set of tags $\mathcal{T}_r \subset \mathcal{T}$. Broad folksonomies can thus be considered as a generalization of narrow folksonomies because they can be transformed to narrow folksonomies but not vice versa.

[20] In other words, $\mathcal{T}_u$ is the user's tagging vocabulary and $\mathcal{R}_u$ is the set of resources tagged by the user.

[21] Please note that in practice Delicious stores additional information in bookmarks such as the title of Web documents. We disregard these additional properties in this discussion for the sake of simplicity.



(a) Broad folksonomy



(b) Narrow folksonomy

Figure 2.3: **Broad and narrow folksonomies.** (a) shows a broad folksonomy, where each user U*i* maintains his own set of tags T*j* for each resource R*k*, i.e. tag assignments can be traced back to individual users. (b) shows a narrow folksonomy, where only one set of tags is jointly maintained for each resource.

In the context of this thesis, we therefore focus our attention on broad folksonomies due to the characteristics outlined above. For the same reason, we use the term *folksonomy* interchangeably with *broad folksonomy* in this work.

2.3 Concepts Related to Folksonomies

In this section, we describe three concepts related to folksonomies and tagging: *subject indexing*, *ontologies* and *taxonomies*.

2.3.1 Subject Indexing

Collaborative tagging and folksonomies have gained widespread use in the Internet in recent years. However, the underlying concept of using keywords (tags) to describe documents (resources) has been around for much longer. In various fields, the act of

describing a document by keywords in order to indicate what the document is about or to summarize its content is studied and known as *subject indexing* [Lan98]. Subject indexing is concerned with creating a representation of a resource in order to facilitate its retrieval at a later time. While there may be different incentives and motivations for annotating resources with tags on the World Wide Web [AN07, RW08a, Kip08], collaborative tagging can be regarded as a form of manual indexing and, fundamentally, as a vocabulary problem in indexing [MM06].

Manual indexing and subject indexing in general are not trivial tasks and face several challenges in practice. The indexing process is comprised of two main steps: *conceptual analysis* and *translation* [Lan98]. *Conceptual analysis* is concerned with deciding on the subject matter of a resource, i.e. the topics addressed by the resource. *Translation* is the subsequent task to assign a set of subject descriptors commonly known as *index terms* to a resource based on the previous conceptual analysis. These selected index terms are derived from a larger set of index terms known as an *indexing language* which constitutes a defined set of terms utilizing established conventions for ordering and combining terms [MM06]. As an example of subject indexing, a human indexer might decide that an article is a scientific essay and addresses research in biology and assign the index terms "research", "biology", "essay" to the article.

However, such judgements are influenced by the characteristics of the individual indexer such as his knowledge, background, interests, motivations or experience. Two indexers may analyze and perceive the same resource differently which may result in different index terms [Hoo65]. In Chapter 4, we will analyze usage patterns in collaborative tagging systems and folksonomies in the World Wide Web and show that they exhibit similar effects.

A notable difference between subject indexing and collaborative tagging is that the former involves a so-called *controlled vocabulary* that is used to control and structure the indexing process a priori, whereas the latter opts for an "everything goes" policy. The interested reader may refer to the study of Macgregor and McCulloch [MM06] for a comparison of such controlled vocabularies with uncontrolled vocabularies as in collaborative tagging.

2.3.2 Ontologies

The term *ontology* has its origin in philosophy, where it is the name of one fundamental branch of metaphysics concerned with analyzing various types or modes of existence. In the field of computer science, an ontology is "an explicit and formal specification of a conceptualization" [Gru95]. In the context of the Semantic Web, the *Web Ontology Language (OWL)* and the knowledge representation language *Resource Description Framework Schema (RDFS)* – both under the umbrella of the W3C and its director Sir Berners-Lee – are means to construct such ontologies. What an ontology has in common in both computer science and philosophy is the representation of ideas (classes) and entities (individuals) together with their properties (attributes) and relations.

Next to general studies of ontologies (the discussion of which is beyond the scope of this work), there have been several studies in the literature that compare ontologies

with folksonomies and analyze their relation.

Christiaens [Chr06] compared the weaknesses and strengths of ontologies and folksonomies. He observed that folksonomies tend to provide quantitative but flexible data, while ontologies could deliver more qualitative but static data. Halpin et al. argue that formal ontologies are of great utility in highly structured domains such as biology, but that collaborative tagging may be a better way of organizing information in other domains such as documents on the World Wide Web [HRS07]. They observed that even in an open-ended domain such as Web documents, there was some consensus about how to categorize the information. Zhang et al. [ZWY06] discuss some of the problems that the "top-down" approach of the Semantic Web and ontologies face as well as why and how the "bottom-up" approach of the Social Web and folksonomies could help.

Mika [Mik05, Mik07] extends the traditional bipartite model of ontologies to a tripartite model of actors, concepts and instances by integrating the social dimension (cf. Section 2.2). He demonstrates how community-based semantics emerge from this model through a process of graph transformation, and illustrates the emergence of a light-weight ontology with a case study of the folksonomy of social bookmarking service Delicious. Similarly, Schmitz [Sch06] describes how to induce an ontology from the tag vocabulary of the photo sharing service Flickr by using a subsumption-based model. On the opposite end of the spectrum, there are approaches such as those of Knerr [Kne06] and Kim et al. [KSB+08] to create ontologies for describing the concept of tagging, i.e. an ontology for folksonomies. Lastly, there is the Annotea project [KK01] of the W3C. Even though it does not provide an ontology for describing the concept of a tagging, it defines an RDF Schema for bookmarking and annotating resources [KSKP03], which is a similar concept.

2.3.3 Taxonomies

A *taxonomy* is a classification scheme with hierarchical structure [Kne06]. A typical and often cited example for a taxonomy is the *Dewey Decimal Classification* system [OCL], which is used by librarians to classify books according to a fixed categorization scheme and organize them into shelves. In the digital world, typical examples of taxonomies are the *Open Directory Project (ODP)* (described in Section 3.1.2) and the *Yahoo! Directory*[22]. Both of them provide a directory of Web resources organized in a fixed set of categories.

While a taxonomy is "hierarchical and exclusive", a folksonomy is "non-hierarchical and inclusive" [GH06]. A folksonomy arises from the free-form annotation of Web resources, done by its users, and without the constraints of a predefined taxonomy [WZY06]. According to Rui Li et al.[LBY+07], similar tags are assigned to similar resources in a folksonomy, and vice versa. They show that there is not a neat tree structure for folksonomies, i.e. they do not exhibit rigid hierarchies or pre-defined categories with clear tag boundaries as is the case for taxonomies or ontologies. Tags are rather located at different semantic levels in the tagging space.

[22] Yahoo! Directory, http://dir.yahoo.com/.

While folksonomies evolve without adhering to any pre-defined policies as is the case for taxonomies and ontologies, we will see in Section 2.4.4 that the tag distribution for each resource still yields a "stable" pattern over time. This means that – given a critical mass of taggers – there will be a few tags for each resource that are most prominent and selected by most of the users to describe that particular resource [GH06], and thus represent the common understanding of the resource from the viewpoint of users.

2.4 Characteristics of Folksonomies

Folksonomies are driven by the collaborative actions of their users. The characteristics of folksonomies therefore depend on the dynamics and patterns of such user behavior. In the following sections, we examine some of these characteristics and summarize the current state of research. Due to the frequent debate about the benefits and drawbacks of using folksonomies as a means for organizing and annotating resources, we start our discussion with an overview of the strengths and weaknesses of folksonomies.

2.4.1 Strengths

Several studies have analyzed the semantic aspects of collaborative tagging and folksonomies, why they are so popular and successful in practice [Mat04, GH06, MNBD06, WZY06, AN07]. In the following paragraphs, we summarize these findings.

Low entry and participation costs

In contrast to more "heavy-weight" approaches such as ontologies, a user does not need extensive domain knowledge or training prior to using a collaborative tagging system [WZY06]. Basically, he can just start using the system as he sees fit. That said, system designers can use various tricks and techniques[23] to guide their user community in finding common usage patterns and support them in finding a common direction.

Individual and community aspects

Collaborative tagging strikes a balance between the individual and the community: the cost of participation – particularly for entering data – is low, and tagging a resource benefits both the individual and the community. This has also a positive effect on the quality of data in a folksonomy: the study of Heymann et al. [HKGM08] found that most tags were deemed relevant and objective by users, i.e. tags are on the whole accurate.

[23] These techniques include tag suggestions and recommendations, for instance showing the most popular tags of a resource within the community, i.e. through analyzing existing information in $\mathcal{F}$ about the resource, or inferring potentially useful tags from the user's tagging vocabulary $\mathcal{T}_u$ by identifying similar resources already stored in his personomy $\mathcal{P}_u$.

Personalization and customization

Users can pick their own terms for labeling resources with tags, i.e. they have full control over their personal tagging vocabulary $\mathcal{T}_u$. Strictly speaking, the tags used to annotate a resource with need to make sense only to them (which includes using their native language instead of being limited to the user interface language that the collaborative tagging system offers), and as such they are not required to use – or to know about – any predefined categories or taxonomies. This freedom also improves the retrieval of any resources in a user's personomy $\mathcal{P}_u$ because he can obtain information more quickly by using his own words to describe resources.

Adaptability

Top-down classification schemes such as taxonomies or ontologies are usually slow to respond to changes to their environment. By definition, the process to predefine such schemes is time-consuming and may depend on the consensus of any involved parties. In contrast, collaborative tagging systems do not put any constraints on their users – a folksonomy $\mathcal{F}$ evolves bottom-up based on its users' individual personomies $\mathcal{P}_u$ (and tagging vocabularies $\mathcal{T}_u$). The effect is that new terms $t' \notin \mathcal{T}$ and new resources $r' \notin \mathcal{R}$ – and thus new information in general – can enter the folksonomy very quickly and can be readily used by its participants.

Feedback and asymmetric communication

Udell [Ude04] argues that the idea of abandoning top-down categorization approaches such as taxonomies in favor of flat lists of keywords (as is tagging) is not new, and that the fundamental difference between these systems is feedback. In collaborative tagging systems such as Delicious or Flickr, the feedback is instantaneous: As soon as a user assigns a tag to a resource, he can see how other users tagged the same item, or see the cluster of resources annotated with the same tag. The user can immediately verify how he relates to the community, and he is given an incentive to rethink his actions. He may chose to adapt to the group norm by changing the tag or adding another, to stick to his actions in a bid to influence the group norm, or to do both. Mathes [Mat04] argues that this tight feedback loop leads to a form of asymmetrical communication between users through tags. The users of a collaborative tagging system are negotiating the meaning of the terms in the folksonomy, whether purposefully or not, through their individual choices of tags to describe resources for themselves. A well-known example for such user behavior is the ESP game [vAD04], albeit in a slightly different context.

2.4.2 Weaknesses

Similarly to the discussion of their strengths in the previous section, several limitations and challenges of collaborative tagging systems and folksonomies have been identified, particularly when compared to more structured or formal schemes such as taxonomies

or ontologies [Smi04, ZWY06, MNBD06, GH06, WZY06]. In the following paragraphs, we summarize these findings.

Lack of semantics

The general structure of folksonomies is very simple as described in Section 2.2. This simplicity is both a blessing and a curse – the curse being a lack of semantics that results in problems such as polysemy, synonymy and variation of specificity [GH06, TT91, ZWY06]. While users of a collaborative tagging system can easily associate tags to resources, the type of these associations cannot be specified. The common (or intuitive) understanding of a tag assignment (u, t, r) is that of "user u thinks that resource r is about t". However, the true reasoning behind the tag assignment is unknown. Similarly, it is generally not possible to specify relations between tags. For this reason, some systems such as BibSonomy [HJSS06a] or TaggyBear (see Chapter 7) have extended the basic definition of folksonomies for their specific problem domains.

To summarize, the ease of use and low entry barriers of collaborative tagging systems and folksonomies come at the cost of a lack of semantics, the result of which could be loosely described as a higher level of uncertainty when compared to more structured approaches such as taxonomies or ontologies. Whether these drawbacks outweigh the benefits must be decided on a case-by-case basis – "there is no free lunch".

Ambiguity of tags

Since users are free to create and pick tags as they please in a collaborative tagging system, it is more difficult to understand the true meaning of tags [Smi04, ZWY06, Sim08]. Firstly, there is the problem of *polysemy*. For example, it is not clear whether a tag t corresponds to the same concept when it is used in two different posts – even when it is assigned by the same user. The tag `golf` is polysemous and could be a reference to a German automobile, a sport or a geographical position. Secondly, there is the problem of *synonymy*. For example, the tags `model` and `mannequin` are equivalent in a fashion context. Other examples of ambiguity are the use of different parts of speech (`fun` versus `funny`), singular versus plural (`tutorial` versus `tutorials`), spelling differences (`color` versus `colour`) or simply spelling mistakes (`biology` versus `bioolgy`) [Sim08].

The empirical study of Zhang et al. [ZWY06] on emergent semantics from folksonomies analyzed to how many concepts a given tag maps on the social bookmarking service Delicious. They identified ambiguous tags such as `todo` or `.imported`[24] that indeed appeared in more than one concept, i.e. their concept distributions (or equivalently, their semantic representations) were different from less ambiguous tags such as `cooking`.

[24] The tag `.imported` is a special tag on Delicious: When users import their existing bookmarks from applications such as the Web browser Internet Explorer, the tag `.imported` is automatically added to any of these bookmarks.

Ambiguity of tags makes tasks such as resource retrieval in folksonomies more difficult than in ontologies, for example, and require proper solutions for disambiguation [GH06]. For instance, Zhang et al. [ZWY06] propose to use a probabilistic generative model to model the user's tag annotation behavior and to automatically derive the emergent semantics of the tags. Au Yeung, Gibbins and Shadbolt [AGS07] demonstrate how different meanings of ambiguous tags can be discovered through an analysis of the tripartite graph of folksonomies, a process they call *mutual contextualization*.

Syntax problems

While the previous paragraphs discussed structural problems of collaborative tagging systems and folksonomies, there are also rather technical problems of collaborative tagging systems in practice. A large number of systems allow users to annotate resources by entering a space-separated list of tags. This means that users have to resort to workarounds for specifying phrases of more than one word. For example, the use of the underscore character "_" or hyphens is very popular on the social bookmarking system Delicious for writing phrases such as `new_york_times` or `science-fiction`.

2.4.3 User Motivation and Functions of Tags

Why do users tag resources on the Web? And how do they decide which tags to use? Next to the discussions of strengths and weaknesses in the previous sections, several studies have analyzed the motivations and incentives of users to contribute to collaborative tagging systems and folksonomies [MNBD06, SLR+06, AN07], or studied the functions of tags and the process of tag choice [GH06, SLR+06, RW08a, Kip08, BH09].

User Motivation

In 1998 – before the advent of tagging and the Social Web – Abrams et al. analyzed why people use bookmarks, based on a survey of 450 users [ABC98]:

> "Bookmarks serve as convenient shortcuts to frequently used Web pages as well as historical pointers to useful information that may otherwise be forgotten. [...] They are created and stored for archival purposes, and often not visited for months. Users must weigh the costs of organizing bookmarks against the expected gains. Thus bookmarking takes place within the context of the users' ongoing information requirements and their assessment of how important current bookmarks will be to them in the future."

Already more than ten years ago, Abrams et al. identified some of the user incentives that are also observed for tagging folksonomies (see below): archiving and organization of information. However, since information stored in bookmarks were not conveniently shared with other users at that time, the social dimension of folksonomies is obviously missing. Anecdotally, one participant of the survey of Abrams et al. seemed

to foreshadow the later developments of the Social Web – storing and sharing bookmark information online – by responding, "I cannot reference a single bookmark file across multiple platforms. I need NFS-like networkable bookmarks" [ABC98].

An early study of user incentives in folksonomies is the study by Marlow et al. [MNBD06], who identified six main incentives. Unfortunately, detailed information about how these incentives were identified and verified are not provided. Ames and Naaman [AN07] analyzed the motivations for tagging on the photo sharing service Flickr. Their qualitative study is based on in-depth, semi-structured interviews with 13 participants. The participants ranged in age from 25 to 45.

Ames and Naaman identified four main incentives of tagging as listed below. We only describe these four motivations because, firstly, they cover the six incentives described in the previous study by Marlow et al., and secondly, the experimental setup of Ames and Naaman is more transparent.

- **Self/Organization - Search and Retrieval**: Tags are used for indexing and categorizing a resource so that the latter can be found again in the future by the user himself.

- **Self/Communication - Memory and Context**: Tags are used to provide metadata about a resource. This includes contextual information that cannot be directly derived from the resource itself (e.g. location where a photo was taken), particularly such information that the user himself might forget over time.

- **Social/Organization - Public Search and Resource Pools**: Tags are used to expose the resource – and the user himself – to the community. Here, a user tags a resource for future retrieval and organization by users other than himself, i.e. he is contributing to and sharing with the community. While seemingly altruistic, there are also personal motivations involved in this sharing process such as self-promotion, i.e. attracting attention from the community and gaining reputation.

- **Social/Communication - Context and Signaling**: Tags are used to communicate contextual information to others about the image and the user himself, including opinion expression and value judgement such as `funny` or `scary` (cf. [MNBD06]). Ames and Naaman report that in most cases, participants of their study added these contextual tags for the benefit of *known* others, such as friends or family.

Ames and Naaman suggest that most participants of their survey were motivated to tag by organization for the general public, i.e. *Social/Organization*, including the aspect of self-promotion. *Self/Organization* and *Social/Communication* were tied for second place. On the other hand, they found that there were often multiple motivations involved even in the use of a specific tag for a specific photo. Generally, most of the participants of their study had one or two primary motivations for tagging as listed above.

Functions of Tags

In this section, we focus our description on the functions of tags in collaborative tagging systems where the type of resources $\mathcal{R}$ is Web documents. Studies of collaborative tagging systems involving other types of resources – for example, photos on Flickr – have observed slightly different tagging behavior. In other words, tagging activity may vary depending on the application context and system design [HHLS05] (cf. Section 2.1.3). The interested reader may refer to the work of Bischoff et al. [BFNP08] for an analysis and comparison of collaborative tagging systems with different types of resources.

In their early study of the social bookmarking service Delicious, Golder and Huberman [GH06] argue that users' tag choices are not random. They analyzed the tags that were used to describe these Web documents, and identified the following seven tag categories:

- **Topic or subject of the resource**: Identifying what or who a resource is about (cf. Section 2.3.1 on Subject Indexing).
- **Type or nature of the resource**[25]: Identifying what kind of thing the resource is (e.g. a resource could be an `article` or a `book`).
- **Category refinement**: Refinements of categories, i.e. adding more specific tags (e.g. `cat`) in order to refine broader tags (e.g. `animals`) applied to the same resource.
- **Ownership of the resource**: Identifying who owns the resource.
- **Subjective opinions of the user about the resource**: Identifying qualities or characteristics (e.g. `scary`, `funny`, or `stupid`).
- **Self-reference**: Identifying content in terms of its relation to the individual user (e.g. tags with the prefix `my` such as `mycomments` or `mystuff`).
- **Task organization of the user**: Labeling resources for later processing or as references to pending tasks (e.g. `toread` or `jobsearch`).

Xu et al. [XFMS06] report a similar categorization of tags in their study of the social bookmarking service *Yahoo! MyWeb 2.0*[26], and in which they also propose several criteria for "good" tags. They describe five tag categories: (1) *content-based* tags; (2)

[25] The study of Golder and Huberman analyzed the folksonomy of Delicious, where resources are always bookmarks of URLs, i.e. references to Web documents. While not explicitly mentioned, the "type" of a resource they describe is not referring to the fact that resources are URLs but which content (type) is depicted on the corresponding Web documents. In other words, the type of a URL such as the news story of the New York Times about US president Barack Obama receiving the Nobel Peace Prize, available at `http://www.nytimes.com/2009/10/10/world/10nobel.html`, could be described as `article`.

[26] Yahoo! MyWeb 2.0 was formerly available at `http://myweb2.search.yahoo.com`. However, Yahoo! discontinued the MyWeb service on March 18, 2008 and moved all users to Yahoo! Bookmarks, which is available at `http://bookmarks.yahoo.com/`, last retrieved on March 01, 2010.

context-based tags; (3) *attribute* tags; (4) *subjective* tags; and (5) *organizational* tags. These categories can be interpreted as a generalization of the categorization by Golder and Huberman.

On the opposite end, Bischoff et al. [BFNP08] further refine the seven categories of Golder and Huberman by introducing the dimensions *time* and *location*. They also describe a mapping between the various categorizations, which is shown in Table 2.1.

No.	Bischoff et al. [BFNP08]	Golder & Huberman [GH06]	Xu et al. [XFMS06]
1	Topic	What or who it is about	Content-based
2	Time	*replaced Refining categories*	Context-based
3	Location		
4	Type	What it is	Attribute
5	Author/Owner	Who owns it	
6	Opinions/Qualities	Qualities & Characteristics	Subjective
7	Usage context	Task organization	Organizational
8	Self-reference	Self-reference	

Table 2.1: **Mapping between tag classification schemes**, based on [BFNP08].

An interesting observation regarding subjective tags, i.e. expression of the users' personal opinions, was described by Paolillo and Penumarthy [PP07]. They analyzed the tagging of Internet videos on Delicious[27] and found that the overwhelming majority of tags with subjective value judgements had positive evaluations. In their data, only about 5% of tags had negative connotations (e.g. `stupid`). Among the positive ones, nearly 70% were associated with humor (e.g. `funny` or `humor`). The rest expressed approval in some form (e.g. `cool` or `amazing`). This evidence supports the observation of Abrams et al. that users strongly prefer to add *useful* resources to their collections [ABC98].

Community Influence on Individual Users

Most tagging systems support the user in the tag selection process by providing tag suggestions, or recommendations, for a resource $r \in \mathcal{R}$. Generally, these suggestions are based on an analysis of the community's tagging behavior, i.e. data derived from the folksonomy $\mathcal{F}$ restricted to r. The majority of theories and mathematical models of tagging found in the literature assume that such tag suggestions influence the individual user's tag selection process. For the same reason, it is assumed that the emergence of power laws in folksonomies (see Section 2.4.4) is mainly driven by the imitation behavior of users when observing tag suggestions provided by the user interface of the tagging system.

In their study of the movie recommendation service MovieLens[28], Sen et al. [SLR+06]

[27] Since video content can be embedded in Web documents, it is possible to indirectly annotate Internet videos on Delicious by tagging (the bookmark of) the corresponding Web document.

[28] MovieLens is run by GroupLens Research, which is part of the Department of Computer Science and Engineering at the University of Minnesota, USA; `http://www.movielens.org`, last retrieved on

examined factors that influence both the way people choose tags, and ultimately, the degree to which community members share a vocabulary. For this, they analyzed three factors that are likely to influence how users apply tags: (1) *personal tendency*, i.e. to apply tags based on a user's past tagging behavior derived from his personomy $\mathcal{P}_u$ [29]; (2) *community influence*, i.e. the effect of the tagging behavior of other users derived from the full folksonomy $\mathcal{F}$; and (3) the *tag selection algorithm* of the collaborative tagging system that chooses which tags to display on its user interface. They found that recommending tags from the community to the individual user via the user interface does indeed influence his tagging behavior, and leads particularly to an increased use of factual tags. Their results also indicate that the user interface has some effect on tag convergence within the folksonomy (see Section 2.4.4).

Similarly, Dellschaft and Staab [DS08] advocate the hypothesis that both the *background knowledge* of a given user and his *imitation* of other users are needed for explaining and understanding the tagging behavior of users. They integrate both aspects into a dynamic generative model of folksonomies that better approximates behavior found in actual tagging systems than previous models that focused on either aspect. Their experiments suggest that the imitation rate during tag assignment is in the range of 60% and 90%, meaning that imitation of other users is indeed more prevalent than a user's personal characteristics.

However, other studies came to different results. Rader and Wash [RW08a] modeled five tag choice strategies for users on Delicious and analyzed whether their simulation results fit to real-world data: One model was based on Zipf's law[30], another model simulated the strategy *Self/Organization* (see Section 2.4.3) based on an individual user's past tag choices (i.e. his personomy $\mathcal{P}_u$), and the remaining three models were imitation-based variants. In their experiments, they could rule out any of these strategies with the exception of the personomy-based organizing model. Their results indicate that a user's past tag choices had a large influence on future tag choices, while the fact that a tag had been used before on a resource by other users (i.e. imitational behavior) had little influence. In other words, the effect of the community on the individual were smaller than previously reported, and tag selection – at least on Delicious – might rather be governed by individual, idiosyncratic processes. It has to be noted though that the study of Rader and Wash is based on a rather small experimental data set that consists of the tagging histories of just thirty Web documents, and as such might not be representative for the full Delicious folksonomy.

March 01, 2010.

[29] Additionally, new users – i.e. users with empty or very small personomies – have an initial tendency based on their experiences with other tagging systems. Other aspects are their comfort with technology, or their interests and knowledge [GH06]. The *personal tendency* thus evolves as people interact with the tagging system.

[30] Zipf's Law [Zip49, New06] is an empirical law that refers to the fact that many types of data in the real-world, e.g. terms in natural language corpora or population distributions, can be approximated with a Zipfian (power-law) distribution. In a Zipfian distribution, the frequency of an item or event is inversely proportional to its frequency rank, i.e. the most frequent item will occur approximately twice as often as the second most frequent item, which occurs twice as often as the fourth most frequent item, etc.

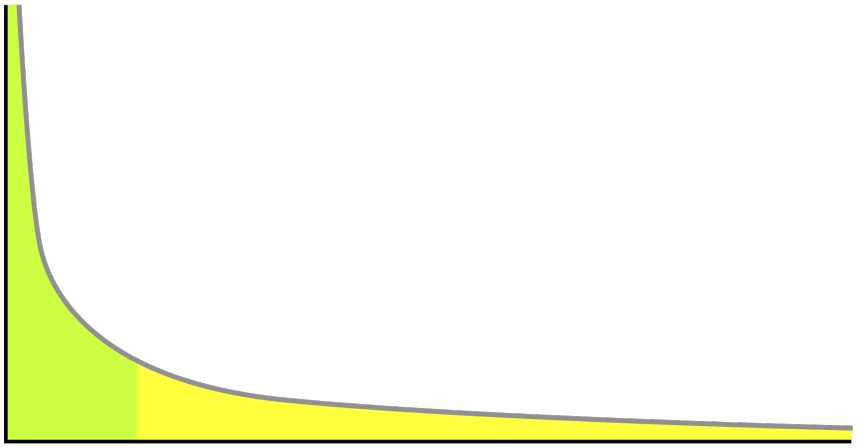


Figure 2.4: **Example of a power-law graph.** To the left are the large values that dominate the graph, to the right is the long tail. Picture by Hay Kranen.

To further complicate the quest for truth, Bollen and Halpin present experimental results that show that power-law distributions in folksonomies form regardless of whether or not tag suggestions are presented to users [BH09]. They argue that this also clarifies how power-law distributions were observed in studies such as [GH06, CLP07] even before tagging systems integrated tag suggestions into their user interfaces.

We can conclude this section by summarizing that the referenced studies show the direction of research in this area, and that tags are clearly not limited to indicating the topic of a document as in traditional subject indexing. Still, the influences on users' tag choices are not fully understood yet. As Abrams et al. already noted in 1998, "given the ever increasing importance of the Web and its role as general repository of information, understanding the bookmarking process and developing appropriate tools for organizing large numbers of bookmarks are likely to become pressing issues" [ABC98].

2.4.4 Dynamics and Usage Patterns

In this section, we report recent findings on the dynamics of folksonomies, i.e. user behavior and the patterns that derive from it.

Power-Law Behavior, Scale-Free Networks and Preferential Attachment

Power-law distributions appear in many real-world contexts such as the distributions of city populations, the number of academic citations, telephone calls, frequency of family names, or BGP routing topologies in the Internet [BA99, New06, MC08]. Well-known examples of power-law functions are the Pareto principle, also known as the "80-20 rule", and Zipf's law [Zip49, Ree01, New06]. In the area of the Web, for example, Adar et al. observed power-law behavior for Web revisitation patterns, i.e. user patterns for (re-)visiting Web pages [ATD08]. Power-law distributions have highly skewed

populations with "long tails", i.e. a limited number of large values appear several orders of magnitude beyond the much-smaller median value. More precisely, a quantity x obeys a power law if it is drawn from a probability distribution [CSN09]:

$$PDF : p(x) \propto x^{-\alpha} \tag{2.1}$$

where α is a positive constant of the distribution known as the *exponent* or *scaling parameter* with $\alpha > 1$. In practice, few empirical phenomena obey power laws for all values of x. More often, the power law applies only for values greater than some minimum x_{min}. In such cases, one says that the *tail* of the distribution follows a power law. An example of a power-law graph is shown in Figure 2.4.

Closely related to power laws in the context of folksonomies are the notions of *scale-free networks* and *preferential attachment* [BA99, BAJ00]. A *scale-free network* is a network (or graph) whose degree distribution follows a power law, at least asymptotically. In other words, the fraction $P(k)$ of nodes $v \in V$ in the network having k connections to other nodes goes for large values of k as $P(k) \sim k^{-\alpha}$, where the scaling parameter α is typically in the range $2 < \alpha < 3$ [CSN09]. As we have described in Section 2.2, a folksonomy can be seen as a tripartite hypergraph $G = (V, E)$, in which the node set V is partitioned into three disjoint sets of users $\mathcal{U}$, tags $\mathcal{T}$, and resources $\mathcal{R}$.

Preferential attachment is the phenomenon of "the rich get richer", which thus affects the *growth* of a network. Due to the preferential attachment, a vertex v_1 that acquired more connections than another vertex v_2 will increase its connectivity at a higher rate, thus an initial difference in the connectivity between two vertices will increase further as the network grows (i.e. $degree(v_1) \gg degree(v_2)$ over time) [BA99]. Preferential attachment can, under suitable circumstances, generate power-law distributions such as those observed in scale-free networks.

Several studies have observed power-law behavior and preferential attachment in folksonomies [HJSS06c, CLP07, HRS07, WZB08, SNRI08, LGZ08, DS08, HKGM08]. Since power laws are the standard signature of self-organization and human activity [Bar05, New06], the presence of a power laws in folksonomies is not surprising.

Some of the power laws observed in folksonomies are:

- **User activity**: A small group of users $\mathcal{U}^* \subset \mathcal{U}$ with $|\mathcal{U}^*| \ll |\mathcal{U}|$ account for most of the activity, i.e. posts $P(\mathcal{F})$ in a folksonomy $\mathcal{F}$.
- **Resources usage**: A small set of resources $\mathcal{R}^* \subset \mathcal{R}$ with $|\mathcal{R}^*| \ll |\mathcal{R}|$ is the center of attention for the activity $P(\mathcal{F})$ in a folksonomy.
- **Tag usage**: A small set of tags $\mathcal{T}^* \subset \mathcal{T}$ with $|\mathcal{T}^*| \ll |\mathcal{T}|$ is used for most of the posts in $P(\mathcal{F})$.

For example, Wetzker et al. [WZB08] report that the Top 1% of users on Delicious account for about one quarter of all posts in the system, and the Top 10% contribute about 60%. They note though that user activity did not completely follow a true power-law distribution (see the "bent" curve of the left log-log plot in Figure 2.5), similar to the

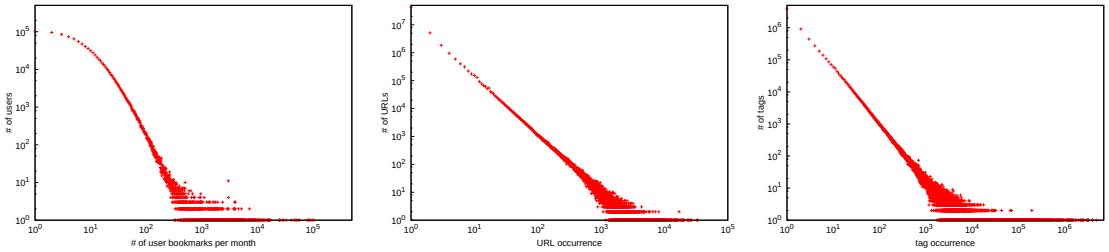


Figure 2.5: **Power laws in folksonomies.** Some power-law distributions found on Delicious as reported by Wetzker et al. [WZB08]. The left, center, and right graphs show user posts per month, posts per Web document, and tag occurence, respectively. Li et al. observed similar patters in their study of Delicious [LGZ08].

findings of [KJHS08]. However, Robert Wetzker remarked in a personal discussion[31] that the x-range from 10^0 to 10^1 shown in Figure 2.5 might be biased due to their data selection strategy for scraping experimental data from Delicious.

Wetzker et al. also observed power-law behavior for the occurrence frequencies of Web documents, where the Top 1% of Web documents were referenced by about 40% of all posts, and the Top 10% documents by 61%. On the opposite end, 80% of all Web documents were posted only once in their experimental data set. Tag occurrence also followed a power law, with the Top 1% of tags accounting for more than 50% of all tag assignments. Their results shown in Figure 2.5 are similar to the observations of Li et al. [LGZ08].

One implication of the power-law distributions of user activity and resource usage, for example, is that discovering the common interests of users on resources in folksonomies differs significantly from discovering the common interests of customers in online shopping systems [LGZ08]. In a shopping system, it is reasonable to assume that although each individual customer may have a small number of purchases, most items should at least have a moderate number of purchases on them; otherwise these items are non-profitable. In a folksonomy, however, the distributions of user activity and resource usage are both long-tailed: most resources are only posted once or twice, and most users only post one or two resources. This makes the discovery of common user interests with techniques such as clustering or traditional collaborative filtering more difficult [LGZ08]. On the other hand, approaches such as hybrid probabilistic latent semantic analysis (PLSA) by Said et al. [SWUH09] benefit from the characteristics of folksonomies by exploiting the higher connectivity within the social graph caused by tags[32] for tackling problems such as the cold start of recommendation systems.

[31] Personal discussion in a VoIP conference between Robert Wetzker and Michael Noll, August 2009.

[32] For a small dataset, the number of user-resource co-occurrences is too low to allow a collaborative filtering recommender to make satisfying predictions for recommending resources to users. Tags and tag-resource co-occurrences, on the other hand, provide higher resource-resource similarities as tags are more abundant and contain contextual information about the resources [WUS09, SWUH09].

Halpin et al. [HRS07] also found power-law behavior for tag usage in their study of Delicious. An interesting observation they report is a significant sharper drop in frequency for Top tags at position seven to ten than the general trend line would predict. They hypothesize that this effect may have a cognitive explanation (i.e. it may be based on the number of tags the average user annotates a post with), or it may be an artifact specific to the Delicious user interface (e.g. Delicious displays seven tags that are popular in the community when a user opens the bookmarking dialogue window as of 2009). The second explanation has later been supported by the experimental results of Dellschaft and Staab [DS08]. It is thus recommended to factor in the effects of a tagging system's user interface when evaluating scientific experiments about user behavior in folksonomies.

The finding that tag distributions in particular tend to stabilize into power-law distributions is very important. A stable tag distribution is an essential aspect of what might be user "consensus" around the classification scheme of a folksonomy [HRS07]:

> "Therefore, given sufficient active users, over time a stable distribution with a limited number of stable tags and a much larger long tail of more idiosyncratic tags develops. One might consider this stabilized distribution an emergent categorization scheme. This stable categorization scheme is described by a scale-free power law, such that in the future, further tagging will only reinforce the pre-existing categorization scheme given by the limited number of stable tags. One might claim that the users have collectively discovered a collective categorization scheme [i.e. a folksonomy]."

We have a more detailed look at stabilization of folksonomies in the next section.

Stabilization of Folksonomies and User Consensus

Generally, a collaborative tagging system does not impose any restrictions on the tagging activity of its users. Intuitively, one might think that this freedom would lead to random or chaotic behavior and noisy data within the system. Under such circumstances it is difficult to imagine that any kind of categorization scheme with meaningful information could emerge. However, in one of the earliest empirical studies of collaborative tagging, Golder and Huberman report that tagging data in fact stabilizes very quickly over time [GH06]. They show that resources tagged in Delicious demonstrate a stable tag distribution – usually after about 100 users have tagged the resource – that follows a power-law pattern in which the same few tags are chosen by many users, while most other tags are selected by only a few people.

Following the argumentation in [GH06, HRS07], these stable patterns can be interpreted as the emergence of a consensus among the community of users for which tags best represent a given resource. Because of the power-law behavior found in folksonomies (see the previous section), such a consensus is not likely to be changed over time even though users may continually assign more tags to the resource. Halpin et al. studied the trend of tag distribution convergence [HRS07]. They report that, for most

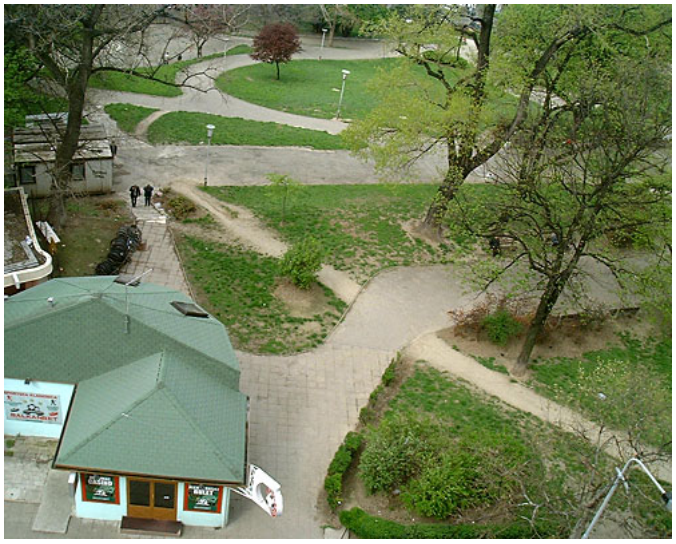


Figure 2.6: **Desire lines in landscapes.** These lines are the foot-worn paths that sometimes appear in a landscape over time [Mer04]. Photo by Phil Gyford.

resources in their experimental data, tag distributions usually converge to a power-law distribution within several months. Additionally, the study of Li et al. reveals that tag convergence for resources is independent from the social popularity of the resource, i.e. the total number of distinct tags assigned by users to a resource is (empirically) limited no matter how popular the resource is within the community [LGZ08].

A recent study by Wetzker et al. [WZBA10] reveals more details with regard to the stabilization patterns of folksonomies. They investigated the social bookmarking systems Delicious and Bibsonomy, and observed that *resources* indeed develop a characteristic and stable tag distribution that is strongly dominated by very few tags. The tagging vocabularies of *users*, on the other hand, react more dynamically due to shifting user interests or tagging behavior. They also note that the tag distributions of users (derived from their personomies $\mathcal{P}_u$) do exhibit power-law characteristics, whereas the tag distributions of resources (derived from the restriction of $\mathcal{F}$ to r) even exceed power-law behavior, i.e. the divergence between very frequent and very rare tags is even larger for resources.

In this context, Merholz [Mer04] makes an analogy of folksonomies with *desire lines*. These lines are the foot-worn paths that sometimes appear in a landscape over time as shown in Figure 2.6. Emergent semantics of folksonomies are similar to desire lines: They emerge from the actual use of tags and resources, directly reflect the user's vocabulary, and they can be used back immediately to serve the users who created them [ZWY06].

One important question is why the reported "consensus" within folksonomies emerges in the first place. In traditional subject indexing, such a consensus is called *indexing consistency*. It refers to the extent of agreement among different indexers with regard to the

terms that are used to index resources. It is found very difficult to achieve high levels of consistency in practice: Hooper [Hoo65] reports a wide range for consistencies in real-world scenarios, ranging from 10% (low) to 80% (high)[33]. While stabilization and convergence of tag distributions in folksonomies are related to indexing consistency, they are not equivalent. Given the power-law behavior of folksonomies, one must expect that the *total* consensus for tag selections among users must be very low due to the negative impact of the long tail, where most tags are chosen by only one or two users (cf. the right side of Figure 2.4 and the right graph in Figure 2.5). And indeed, we observed such a low global consensus in our own studies as described in Chapter 4, where we investigate the diversity of data in folksonomies. On the other hand, however, the power-law behavior of tag distributions also implicates that a large number of users agree on a small set of tags (cf. left side of Figure 2.4). This means that users collectively arrive at a consensus on which tags are *the most important* to describe a given resource. When we talk about emergent consensus in folksonomies with regard to tags, we are thus mainly referring to those tags that have managed to "escape" the long tail of tag distributions for both users and resources, i.e. cumulative effects dominate over local heterogeneities.

Golder and Huberman suggest two possible reasons for the emergence of consensus in folksonomies [GH06]: Firstly, because of *imitation* of the tagging behavior of other users, and secondly, because of *shared knowledge* within the user community. And indeed, imitation-based behavior models have been shown to simulate user behavior quite effectively as described in Section 2.4.3. On the other hand, studies such as [MNBD06, WZBA10] indicate that shared knowledge is also playing an important role in this process. For example, Marlow et al. [MNBD06] show that the overlap of tagging vocabularies is higher with a user's contacts (users in his network) than with random users. They argue that this could be caused by shared knowledge such as dialect, sociolect, ethnolect, ecolect and idiolect [MNBD06]. Still, the final answer to the question of why a consensus emerges has yet to be found.

While the effects of power-law patterns in folksonomies appear to be primarily advantageous, there is also a potentially negative aspect in this context. As Moore et al. state in [MC08] in their study of the collaborative anti-phishing service PhishTank[34], the intuitive argument put forth in favor of the robustness of "crowd-sourced" applications such as collaborative tagging systems is that the opinions of many users outweigh the occasional statistical outlier, or even the views of a malicious user. If the activity in a system follows a power-law distribution, however, it also means that a single highly active user can greatly impact the system's overall accuracy. Moore et al. continue by concluding that this is why a power-law distribution invalidates the standard Byzantine fault tolerance[35] view of reliability [LSP82], because the subversion of even a single

[33] Hooper [Hoo65] measured the consistency of a pair of indexers via the formula $CP = \frac{|T_M \cap T_N|}{|T_M \cup T_N|}$, where CP is the consistency (expressed as a percentage) of term agreements between the two indexers M and N, and T_M and T_N are the sets of terms used by indexer M and N, respectively.

[34] PhishTank, http://www.phishtank.com/, last retrieved on March 01, 2010.

[35] The Byzantine fault tolerance is a generalized version of the *Two Generals' Problem*. In a multi-component system, the goal of Byzantine fault tolerance is to defend against a Byzantine failure, i.e. a component

highly active participant could undermine the system. As we will see in Section 2.6, popular collaborative tagging systems suffer from spamming activity. One common spam strategy is to flood the system with large amounts of junk information, resulting in the spammers becoming part of the highly active user group in the system. For example, Wetzker et al. [WZB08] found that 19 of the Top 20 most active Delicious users in their experimental data set were spammers who posted ten thousands of Web documents pointing to only few Web domains. In total, these 19 spammers alone accounted for 1.3 million bookmarks or around 1% of their data corpus. For this very reason, we propose the *SPEAR* algorithm in Chapter 5 in order to favor *quality* over *quantity* of user activity in folksonomies, and thus harden a folksonomy against "faulty" or "malicious" users.

2.5 Folksonomies and Recommender Systems

Like tagging systems, *collaborative filtering* [HKTR04] is concerned with the relationships between users and resources ($\mathcal{U} \times \mathcal{R}$), and the extent to which these connections can be leveraged for tasks such as supporting users in finding new resources or users with similar interests. Collaborative filtering tries to find solutions to problems such as "Find books related to the book I'm buying" and "Recommend movies I might want to watch". It is typically used in domains such as online shopping systems (support customers in buying items based on purchase histories) or social networks (support people in finding other people).

An important requirement for collaborative filtering is an adequate understanding of the interests and preferences of users. The process which models these user preferences is called *user profiling*, and the effectiveness of a collaborative filtering system heavily depends on its accuracy. User profiles can be built from implicit and explicit user feedback on resources and other users. For example, the purchase of the movie "The Godfather" by a user on Amazon.com can be considered as implicit, positive feedback on both the quality of the movie and the interests of the user in its genre or topic. Rating the same movie on Amazon.com with five out of five stars represents a direct expression of user opinion and is thus an example of explicit user feedback.

Collaborative tagging systems can also be considered as a form of collaborative filtering [MNBD06] in which the act of tagging a resource by a user represents explicit user feedback on the resource, with tags serving as the voting element. Since tagging resources is a very subjective user task (cf. Section 2.4.3), the data in a folksonomy provides a lot of information for understanding the interests and preferences of users. An encouraging observation for leveraging folksonomies for Web information retrieval is reported by Al-Khalifa and Davis [AKD07]. They show that tags of a folksonomy

that does not only behave erroneously but also fails to behave consistently when interacting with other components of the system. A Byzantine fault tolerant system is able to function correctly in spite of faulty components as long as the number of faulty components does not exceed a certain threshold (also called its *resilience*). It can be shown that if n is the total number of components, and $t < n$ is the number of faulty components in that n, then there are solutions to the problem only when $n \geq 3t + 1$.

for resources agree more closely with human-generated keywords than those automatically generated. They also report that trained human indexers preferred the semantics of tags compared to machine-generated keywords in their study. Similarly, Li et al. [LGZ08] found that tags are in general better than term weighting schemes such as TF-IDF in representing a human being's judgements about Web content. Therefore, tags are good candidates for profiling tasks, particularly since they serve a dual purpose as they allow for a profiling of both users (interests in topics) and resources (topics).

Several studies in the literature have exploited folksonomies for recommendation of resources. Niwa et al. [NDH06] propose an approach for Web resource recommendation by leveraging folksonomy information from the social bookmarking service Delicious. User interests are modeled by associating each user with a tag cluster, and Web resources are subsequently recommended based on their relatedness to these clusters. Soyanovich et al. [SYMY08] model a user's interests based on his tagging vocabulary as well as his explicitly stated and implicitly derived social ties within a folksonomy. They demonstrate how such the resulting user profiles can be leveraged to create so-called "hotlists" of resources that are recommended to individual users. Wetzer et al. [WUS09] propose a hybrid approach that recommends resources based on the user-resource distribution (as in collaborative filtering) combined with the tagging information of resources. In a follow-up work by Said et al. [SWUH09], the authors note that the inclusion of tagging information is especially helpful for mitigating the cold start phase of newly created tagging systems. At this point in time, the user-resource distribution by itself is normally quite sparse, and tags help to achieve a higher connectivity between users and resources.

In Chapter 6, we will describe our studies on leveraging folksonomy information for constructing user and resource profiles in order to personalize Web search.

2.6 Folksonomies and Spam

The convenience of collaborative tagging systems has attracted an increasing number of users over the recent years. However, the rising popularity of these systems has also encouraged malicious individuals to abuse these systems for their own benefits. Nowadays, *spam* has already become a significant problem for collaborative tagging systems in practice. While there exists a variety of definitions of spam depending on the specific context, a common notion of spam is that of "unsolicited bulk messages". It refers to the practice of sending unwanted messages and information – frequently with commercial content – in large quantities to an indiscriminate set of recipients. Long before collaborative tagging became popular, spam was already a problem in other domains such as fax transmission, email messaging [AKCS00, Hid02, Cor07] and Web search [DWV99, GGMP04, GGM05, BCSU05, AS08]. Back in 2006, Hotho et al. [HJSS06c] still stated, "spam is [currently] not a serious problem for social bookmarking systems", but they already anticipated the advent of spam in collaborative tagging systems. On the popular social bookmarking service Delicious, for example, most of the highly active users have been found to be spammers [WZB08]. While some features and techniques

that were developed for fighting spam in domains such as email and Web spam, e.g. content-based or link-based spam detection [WD05, NNMF06, CDG+07, GS07], can be transferred to collaborative tagging systems, most of these countermeasures do not directly apply to these systems [HKGM07], or at least may be improved upon. This can be achieved, for instance, by integrating additional, domain-specific features such as user activity profiles or an analysis of the co-occurence of tags and resources [KSHS08].

The general objective of spammers is to bring their content to the attention of legitimate users. However, their means of doing so depends on the characteristics of the target environment. In collaborative tagging systems, for example, spammers [KEG+07, KFG+07, KSHS08, WZB08] may register multiple user accounts that they abuse for collectively posting the same resource multiple times. Such a strategy is quite effective on tagging systems that rely on a quantitative measure like the number of posts of a resource to derive its popularity within the community. Another strategy of spammers is to assign popular tags indiscriminately to their own resources, so that these resources are more likely to be displayed to users when they navigate or browse the folksonomy by tag. This type of spam pollutes the folksonomy by creating artificial links between resources and tags that would otherwise be unrelated. It thus affects the measures of tag and resource similarity that are grounded in tags, and thereby has impacts on recommendation, ranking and search [MCM09a, NO09]. For instance, a spammer in Delicious could attach the tags `apple` and `iphone` to a Web site that in fact sells the viagra drug in order to trick unaware users into visiting that resource because they expect information about Apple's mobile phone. A third strategy of spammers is to mimic legitimate user behavior in order to gain reputation within the community by re-posting content that is already known to be popular within the folksonomy, and then abuse this reputation to promote their own content.

The attack vector of the first strategy is to intentionally increase the post count of a resource r, i.e. it increases the set of posts $P_r(\mathcal{F})$[36] by increasing the number of distinct users u_i of a resource through multiple tag assignments $(\mathbf{u_i}, t, \mathbf{r})$, whereas the attack vector of the second strategy mislabels r by manipulating the tags t_i in tag assignments $(u, \mathbf{t_i}, \mathbf{r})$. The attack vector of the third strategy depends on the way a tagging system calculates a user's "reputation" within the community, and how much information is available to spammers about this process in order to game the system. For example, a spammer could construct the set of popular resources $R_{pop} \subset R$ within $\mathcal{F}$ and start from there[37]. Of course, such strategies can be combined for increased effect.

Studies such as [CSB+07, KEG+07, KFG+07, HKGM07] are the first to deal with spam in tagging systems explicitly. Heymann et al. [HKGM07] discuss the spam problem on social Web sites, while putting a focus on collaborative tagging systems. They differentiate three main approaches to tackle spam: *prevention*, *detection* and *demotion*.

Prevention-based approaches operate *a priori*. They try to stop spamming activity before it actually happens. For example, such approaches can be integrated into the

[36] $P_r(\mathcal{F})$ refers to the restriction of the total set of posts $P(\mathcal{F})$ to r within the folksonomy $\mathcal{F}$.

[37] What helps spammers in this context is that information about R_{pop} is normally readily available: Most tagging systems in practice are leveraging such popular resources themselves in order to attract more users and thus display those resources prominently on their Web sites.

Figure 2.7: **An example of a CAPTCHA.** Here, it is put in place to protect Google accounts.

user registration process in order to prevent spammers from creating user accounts in the tagging system. A popular technique is the use of CAPTCHA [vABHL03, vABL04], a reverse turing test that presents challenges to users that are easy to solve by humans but difficult to solve by machines (see Figure 2.7). Such computationally expensive barriers hamper the ability of spammers to automate user activity, thereby reducing the scale of spam.

Detection-based approaches, on the other hand, operate *a posteriori*. They include techniques such as analyzing tagging behavior within a system in order to detect malicious user activity. Cattuto et al. [CSB+07], for example, observe that tagging behavior creates characteristic power-law distributions (see Section 2.4.4), with major derivations from these patterns caused by spam entries. Similarly, Neubauer and Obermayer [NO09] propose a spam detection approach that constructs hyperincident networks from folksonomies. In these networks, vertices represent the hyperedges of the folksonomy graph (see Section 2.2). They report that the connected components of these networks show structural differences for spammed and spam-free networks, with the respective "gap" depending on the extent of spammer activity. Neubauer and Obermayer argue that spammers do not merely post different resources with different tags, but they behave differently from legitimate users in such a fundamental way that it structurally changes the resulting networks. This argument is supported by Markines et al. [MCM09a] who report that legitimate users share a prevalent vocabulary to annotate resources [XFMS06], whereas spammers often use tags and tag combinations that are statistically unlikely to appear in legitimate posts.

Demotion-based approaches operate *a posteriori* as well. They try to reduce the impact of spamming activity by assigning lower ranks to malicious users (and their subsequent activities) compared to regular users, thereby hindering spammers from bringing their content to the attention of regular users. Such techniques are often used in the context of ranking users, tags, or resources. Typical examples are the cleaning of search results, navigational overviews or browsing features that rely on a list-type display.

For example, Koutrika et al. [KEG+07] propose an approach based on the reliability of a user's tag assignments (u, t, r). They estimate this reliability by measuring how often a user's posts coincide with other users' postings. This measure is then used to demote malicious users who purposely assign unrelated tags to resources (see discussion above).

We can conclude that a thorough understanding of spamming activities and finding proper countermeasures are important challenges for folksonomies and collaborative tagging. A direct consequence of spam is that a vulnerable tagging system will see its quality and value for legitimate users deteriorate. As an indirect consequence, research on collaborative tagging must bear in mind that experimental data might be polluted by spam, which can impact the correctness of experimental results and their interpretations. Any of the anti-spam approaches described above comes with its own benefits and drawbacks. Firstly, prevention-based approaches can lead to collateral damage by putting off both spammers and legitimate users because they increase the burden for both groups. Secondly, detection-based approaches often require large sets of tagging data in order to operate effectively. On the one hand, this requirement can limit their value for small to medium-sized tagging systems, and on the other hand, it increases the technical difficulty to use these approaches in a practical setting where time and computational resources are restricted. Lastly, demotion-based approaches are vulnerable to focused spammer attacks. It is thus likely that a hybrid approach will prove to be the most effective and efficient in both theory and practice.

2.7 Ranking in Folksonomies

The science of Web information retrieval is primarily concerned with searching and finding relevant and high quality resources of information on the Web. Here, an important task is to *rank* resources in order to present the "best" resources to users first. In the context of Web search, the most prominent example of ranking is search results, where resources are ranked by relevance in descending order [Kle98, BP98].

Folksonomies are no exception in this context, as indicated by research such as the study of Chi and Mytkowicz [CM08]. The latter analyzed the efficiency of collaborative tagging systems with information theory. One of their observations is that the number of documents assigned with a specific tag keeps increasing – particularly for highly popular tags (cf. Section 2.4.4) – which means that navigating a folksonomy or finding resources through tags will become increasingly difficult for users over time. A possible solution to this problem is the use of ranking techniques. In traditional Web information retrieval, ranking is based on the bipartite structure of the Web graph and is thus applied only to resources $\mathcal{R}$. In the tripartite structure of folksonomies, however, it is often desirable to rank users $\mathcal{U}$ and tags $\mathcal{T}$ in addition to resources $\mathcal{R}$.

Several algorithms for ranking in folksonomies have been proposed in recent years. The topic-sensitive *FolkRank* algorithm by Hotho et al. [HJSS06c] is an adaptation of the *PageRank* algorithm by Brin and Page [HJSS06c], and ranks the three entities of a folksonomy – users, tags, resources – at the same time. The algorithm is a topic-specific

ranking method that can make use of a user preference vector to allow for personalized ranking. John and Seligman propose *ExpertRank* [JS06] for ranking users by their expertise in a particular topic represented by a tag. They describe two implementations of ExpertRank: The first variant relies solely on a user's number of posts assigned with a specific tag to determine his expertise, the second variant additionally integrates a tag-occurence analysis to account for dependencies between tags. A user's expertise level as returned by ExpertRank is therefore primarily determined by his own actions and only to a lesser extent by the behavior or opinions of other users. Bao et al. [BXW+07] propose the *SocialPageRank*, which measures the popularity of Web resources from the perspective of Web users through a mutual reinforcement scheme of the levels of popularity between the tree entities of a folksonomy. Similarly, Abel et al. propose *Social-HITS* [ABB+09], which is an adaptation of the *HITS* algorithm [Kle98] to the tripartite structure of folksonomies.

The common assumption of all these algorithms is that a *resource* which is tagged with popular tags by active users becomes popular itself (with similar notions for *users* and *tags*), in other words a mutual reinforcement of importance. However, as we have described in Section 2.6, the notions of the popularity and activeness are susceptible to spamming activities, which makes them in practice not as reliable as they might seem on first glance. This means that rankings produced by algorithms as those described above may be biased due to putting too much emphasis on the quantity of activities in a folksonomy. In Chapter 5, we will further elaborate on the relations of ranking and spamming in folksonomies and also propose a new approach for ranking users and resources in folksonomies.

2.8 Summary

In this chapter, we have provided a review of collaborative tagging and folksonomies. We have put folksonomies into context with related concepts, and presented the current state of research in areas such as user motivations, dynamics, and the impact of spam on folksonomies in practice.

The advent of collaborative tagging and folksonomies provides ample opportunities for research in a wide range of fields. Given the prior research we have discussed in this chapter, the current challenge for scientists is two-fold. Firstly, it is important to improve our understanding of the nature of folksonomies. Secondly, we need to find out how the richness of information and the hidden semantics contained in these user-driven systems can be exploited to create and improve techniques and methodologies in research domains such as Web information retrieval.

In the following chapters, we will describe our contributions to a better understanding of folksonomies and how we leverage this knowledge for enhancing and improving techniques in the domain of Web information retrieval. We will start these discussions in the next chapter with an overview of the experimental data sources and experimental data sets that we have created and used for these studies.

Measure what is measurable, and make measurable what is not so.

Galileo Galilei (1564–1642)

3 Experimental Data

An essential prerequisite for most scientific studies in our research domain is the compilation of appropriate experimental data sets. For the research work described in this thesis, we need access to large volumes of real-world data that is sufficiently general and representative of a broad range of domains to allow for *generalization* of experimental results and any derived conclusions. Additionally, this data must support us in measuring and quantifying qualitative aspects where appropriate and necessary. Our studies and analyses leverage various sources for building such experimental data sets, of which the most important source is the collaborative tagging system *Delicious*.

In this chapter, we present the major data sources used in this thesis and discuss why they are suitable targets for our research. Additionally, we describe the technical tools that we have created and used to collect data from these sources. We also give an overall description of the main experimental data sets that we have subsequently constructed and that will be used in our studies presented in the later chapters.

3.1 Main Data Sources

3.1.1 Delicious

The social bookmarking service *Delicious*[1] is one of the most popular tagging systems in the Internet and the main source of experimental data for the work described in this thesis. For this reason, we provide a more detailed view of Delicious in this section. We have already described in Section 2.2 that bookmarks in Delicious can be adequately modeled by our definition of *posts* in folksonomies. As such, we will use the term *social bookmark*, or simply *bookmark*, interchangeably with "post" for the remainder of this thesis[2].

In a nutshell, Delicious allows users to manage, organize and share references to Web resources – commonly known as "bookmarks" – including related metadata[3]. By referencing its URL, a user can create and save a bookmark of a Web resource to Delicious,

[1] Delicious, http://delicious.com/, a Yahoo! company.

[2] Another reason is that prior research studies often uses the term *social bookmark* as well, thereby making it more convenient for the reader of this thesis to perform comparisons across works.

[3] Based on the self-description of Delicious as stated on its *About* page at http://delicious.com/about, last retrieved on March 01, 2010.

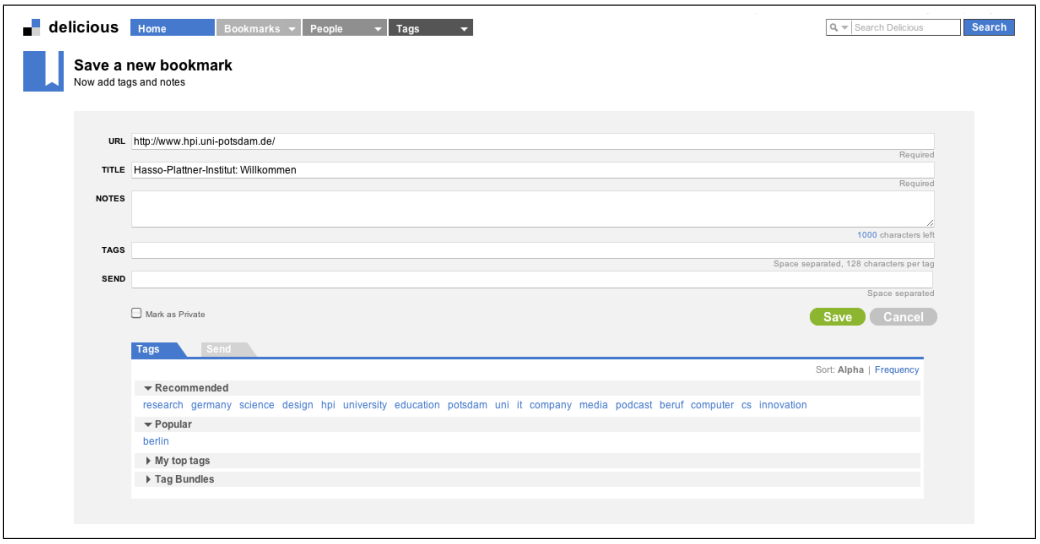
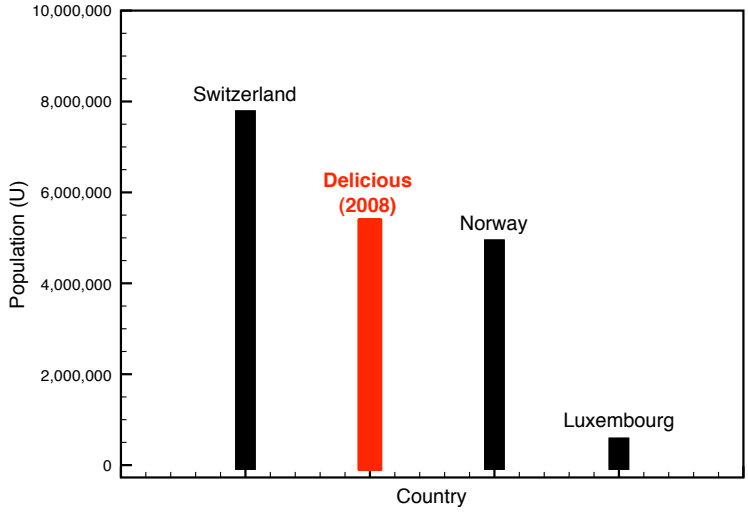


Figure 3.1: **Delicious user interface (UI) - Posting a new bookmark.** The UI includes features such as presenting the user with a list of tag suggestions ("Recommended" and "Popular" tags) and marking the bookmark as private, i.e. only viewable by the user himself. In this example, tag suggestions for the homepage of the Hasso Plattner Institute include `research`, `germany` and `science`.

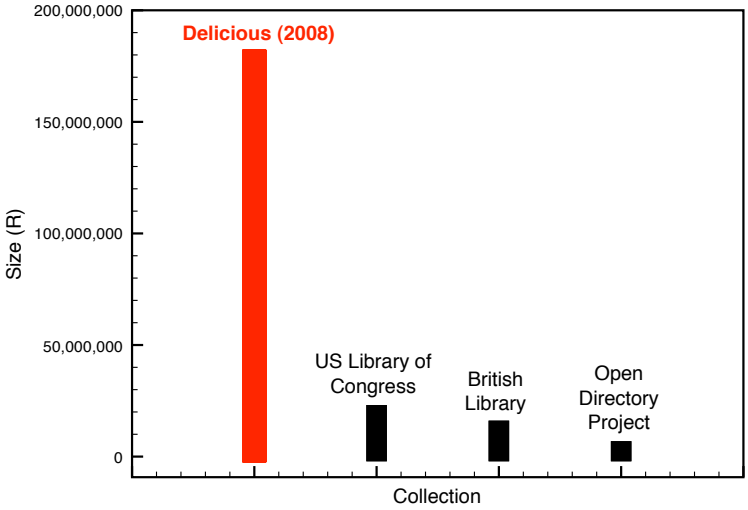
and by doing so, add the resource to his personal collection (see Figure 3.1)[4]. He may optionally store additional metadata about the resource in the bookmark such as a title and tags, with the latter metadata representing the act of tagging the resource. This also means that users may create bookmarks that do *not* contain any tagging data. For completeness, we therefore differentiate between *bookmarked* (at least one bookmark) and *tagged* (at least one bookmark with tags) Web resources in the analyses of our experimental data sets presented in the later chapters.

Users can create bookmarks through the Delicious Web site, Web browser plugins, mobile phones or other third-party applications and online services, i.e. basically from anywhere on the Web. The social aspect of bookmarking on Delicious is similar to the general case of collaborative tagging: All bookmarks on Delicious are publicly viewable by default, i.e. shared with the community, although users can mark specific bookmarks as private. Since resources are identified by URLs, virtually any Web resource with a valid URL – Web pages (aka Web documents), images, videos, etc. – can be posted to Delicious. We have chosen to use the folksonomy of Delicious for our studies due to four major reasons: size and popularity, diversity of information, data granularity, and interfaces. A comparison of the dimensions of Delicious with the "offline world" is illustrated in Figure 3.2.

[4] A very early prototype of a system for personalized Web resource organization called *PowerBookmarks* is described by Li et al. [LVC+99]. PowerBookmarks already included the means to share bookmarks with other users but lacked any other social features such as collaborative tagging.



(a) Population of Delicious versus countries



(b) Size of Delicious versus other collections

Figure 3.2: **Comparison of the dimensions of the collaborative tagging system Delicious with the "offline" world**. Figure (a) shows the user population ($\mathcal{U}$) of Delicious in relation to some of the world's countries. If Delicious was a country, its population size would rank it at position 113 of 223 in the world. Figure (b) shows the number of resources managed by users on Delicious ($\mathcal{R}$) in relation to other collections of mankind such as the US Library of Congress, the largest library in the world. The Open Directory Project, the largest human-edited directory of the Web, trails behind with about 5 million resources (see Section 3.1.2).

- **Size and popularity**: Launched in 2003, Delicious has since become the most studied folksonomy on the Web [WZBA10]. It has a large *user population* of more than five million users as of 2008 ($|\mathcal{U}| > 5,300,000$) who have created bookmarks of more than 180 million unique Web resources ($|\mathcal{R}| > 180,000,000$)[5] – in other words, it is a huge collection of user-contributed data. Judging from the reported rate of growth between 2006 and 2008[6], the Delicious user community might have superseded by now the populations of countries such as Switzerland and Austria. Figure 3.2 puts these numbers into context. Similarly, the *resources* managed on Delicious already cover a considerable fraction of the Web [NM08c, HKGM08]. The related tagging system *BibSonomy* [HJSS06a], for example, has been found to be smaller than Delicious by two orders of magnitude [CSB+07]. Lastly, Delicious is reasonably well-researched in the literature, which helps to make our results comparable with other studies.

- **Diversity of information**: Any resource with a valid URL can be posted to Delicious, which allows for a diverse range of content topics and resource types in the Delicious folksonomy. Even though some studies argue against the topical diversity on Delicious such as [HJSS06d], its diversity and the range of interests of its users are still larger than those of other tagging systems (particularly systems that focus on a single resource type such as images) such as LibraryThing, Flickr or Bibsonomy [BFNP08, AGS08b]. Parts of this can also be attributed to its size: a folksonomy with more than five million users and 180 million unique resources can hardly be considered as a uniform platform – even in the presence of power laws. Additionally, Delicious has been found to be less reliant on its "Top Users" than other tagging systems [HKGM08], thus limiting a potential global bias of its folksonomy caused by a relatively small number of highly active users (cf. Section 2.4.4 on power laws in folksonomies).

- **Data granularity**: As described in Section 2.2.2, Delicious is a broad folksonomy, and as such it yields a finer data granularity for experiments and scientific studies than narrow folksonomies such as Flickr or YouTube. Every user on Delicious may maintain his own personomy $\mathcal{P}_u$ and tagging vocabulary $\mathcal{T}_u$, thereby increasing the data available within the folksonomy. For example, this also means that Delicious provides a better starting ground for constructing user and resource profiles from tagging data. Additionally, the finer granularity of a broad folksonomy such as Delicious ensures that the results of our studies can be generalized to other folksonomies and collaborative tagging systems.

- **Human and Machine Interfaces**: In addition to the user interface for human users, Delicious provides an application programming interface (API), i.e. an in-

[5] Delicious announcement, http://blog.delicious.com/blog/2008/11/delicious-is-5.html, last retrieved on March 01, 2010.

[6] Delicious reported one million users in September 2006, two million users in March 2007, and five million users in November 2008. Roughly speaking, its user population doubled every twelve months in the past.

terface for machines. Since this API and other communication channels such as RSS[7] streams were provided early on by Delicious, they have resulted in the creation of many third-party online services that interact with Delicious, thus further stimulating the growth of its folksonomy. These technical interfaces also facilitate our data collection tasks, even though we still have to fall back to non-standard data retrieval techniques to collect all the data we need (see discussion below).

While most of the information on Delicious is publicly accessible, a convenient method of retrieving sufficiently large amounts of data for scientific research is missing, and an official Delicious data set has yet to be released. Due to business secrets, privacy rights and other legal or commercial implications, services such as Delicious are very reluctant to offer large-scale access to their user data. For example, it is impossible to obtain the full bookmarking history of a particular document, or a large number of documents that are assigned a particular tag. Similarly, the Delicious platform technically restricts and throttles mass requests to its Web site and machine interfaces (API, RSS streams, etc.), thus limiting the speed and efficiency of crawling and mining the Web site. Due to these restrictions, researchers have been forced to fall back to improvised means to build experimental data sets from Delicious, for example by directly crawling its Web site and extracting any relevant information from the raw HTML sources. However, such workarounds come with their own set of limitations, particularly regarding the amount and granularity of information that can be extracted from such crawls. For example, it is only possible to retrieve the creation date ("February 05, 2009") of public bookmarks from Delicious' Web pages but not the creation time ("8.30 am"). One might indeed argue that some of the differences in experimental results between scientific studies of folksonomies can be attributed to the difficulty of collecting and compiling proper experimental data sets [KS10].

In view of the above limitations, we have developed a custom programming library called *DeliciousAPI* for working around Delicious' technical restrictions on data retrieval. It combines a crawler component for mining the Delicious Web site and can also collect data through the official Delicious API and news feeds (in RSS and JSON formats) where available. Among its features, DeliciousAPI allows for the retrieval of the bookmarking histories of Web resources (see Figure 3.3) and the public bookmark collections of users. For example, the former feature can be used to compile the restriction of $\mathcal{F}$ to a particular resource r, whereas the latter extracts a user's posts in $P(\mathcal{F})$, which can be used for building his personomy $\mathcal{P}_u$. Since its creation and publication in July 2007, DeliciousAPI has been downloaded about 10,000 times, bookmarked by more than 250 Delicious users, and used by or integrated into several academic, commercial and non-profit projects. DeliciousAPI is written in the Python programming language and available as free and open source software licensed under the GNU General Public License (GPL). It can be downloaded from the official Python Package Index[8]. With regard to the time duration required to build experimental data sets, Delicious states in its terms of use that developers should wait one second in between HTTP requests

[7] *Really Simple Syndication*, an XML-based feed format used to publish frequently updated content.
[8] DeliciousAPI, http://pypi.python.org/pypi/DeliciousAPI, last retrieved on March 01, 2010.

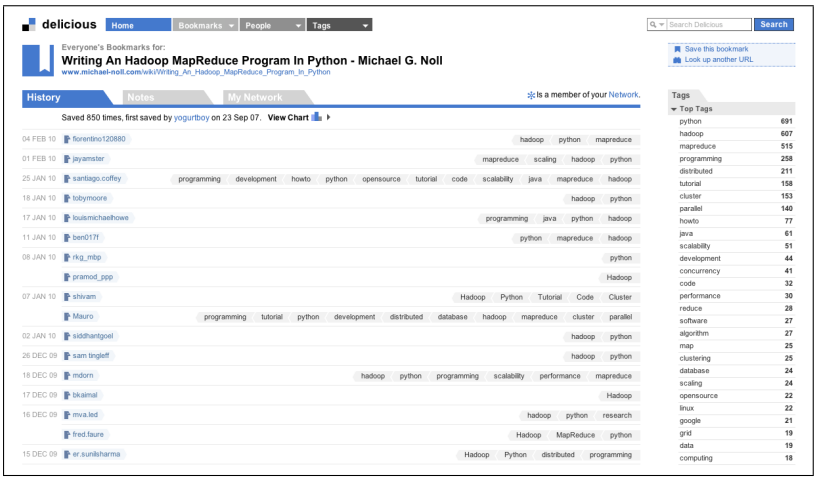


Figure 3.3: **Delicious user interface (UI) - Bookmarking history of a Web resource.** The "history" of a Web resource on Delicious includes all public bookmarks of the document including related metadata such as the creation date or tags of bookmarks. In this example, the history of the article "Writing an Hadoop MapReduce Program in Python" by the author of this thesis is shown [Nol07c]. The article has been bookmarked by 850 users since its publication on September 21, 2007, with the first Delicious bookmark being created just two days later.

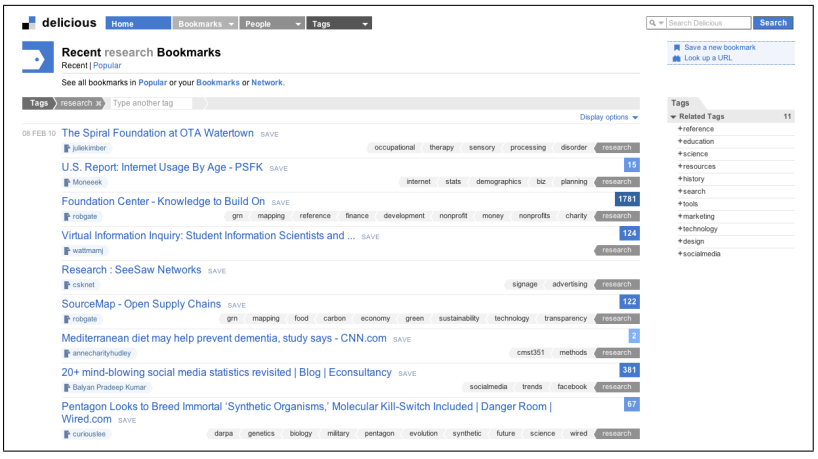


Figure 3.4: **Delicious user interface (UI) - Browsing by tag.** Delicious provides various means to navigate and browse its folksonomy. For example, the most recent bookmarks annotated with the tag `research` are available at the URL `http://delicious.com/tag/research`. On the right side, tags related to `research` are listed so that users can easily drill down through the list.

to its services, with violations to this rule resulting in temporary access restrictions as described above. However, we have found that in practice this interval is rather six to seven seconds, particularly for extended crawling runs. This means, for example, that it may take from several hours to several days to collect the data of a particular tag. The long creation times of our experimental data sets described in Section 3.2 can be mainly attributed to this restriction of Delicious.

3.1.2 Open Directory Project

In Web resource taxonomy, the *Open Directory Project (ODP)*[9] is the largest, most comprehensive human-edited directory of the Web. Launched in 1998, the directory is constructed and maintained by a global community of 84,000 volunteer editors[10] ($\mathcal{U}_{ODP}$) who evaluate and categorize Web resources based on common standards and best practices to ensure consistency. For instance, prospective editors have to file an application form which includes a categorization test, and senior Open Directory editors must review and evaluate the application before a new candidate can become an editor. This review process helps to ensure that Web documents will properly organized and categorized in the catalog by all its editors.

At the time of writing, the ODP contained 4.8 million Web documents ($\mathcal{R}_{ODP}$) in about 590,000 categories. Documents are assigned to one or more category hierarchies such as *Arts* > *Crafts* > *Textiles* > *Weaving*. ODP data is stored in RDF format[11] and freely available for download[12] from the ODP homepage. At the time of writing, the size of the ODP RDF data corpus was about 2 GB. We developed custom applications written in the Python programming language for parsing and extracting the required information from the RDF data corpus.

While the controlled Web taxonomy of the ODP is the largest of its kind, even a first comparison with the dimension of the uncontrolled folksonomy of Delicious shows that the latter system operates on a much large scale even though it is twice as young:

- Users: $\frac{|\mathcal{U}_{ODP}|}{|\mathcal{U}_{Delicious}|} = \frac{84*10^3}{5{,}300*10^3} \sim \frac{1}{60}$
- Resources: $\frac{|\mathcal{R}_{ODP}|}{|\mathcal{R}_{Delicious}|} = \frac{4{,}8*10^6}{180*10^6} \sim \frac{1}{40}$

Still, the ODP is a valuable data source that has been used in quite a number of prior research studies such as [PSC+02, LYM02, Hav02, CNPK05, QC06, XBF+08, XXYY08]. It also forms the basis of the Web directory services of companies such as AOL and Google[13]. In this thesis, we use the ODP data primarily as the ground truth for classi-

[9] Open Directory Project, http://www.dmoz.org/, last retrieved on March 01, 2010.

[10] Article on the Open Directory Blog, January 29, 2010; http://blog.dmoz.org/2010/01/29/dmoz-a-decade-in-review/, last retrieved March 01, 2010.

[11] *Resource Description Framework (RDF)*, described by the W3C, is a standard model for data interchange on the Web. Available at http://www.w3.org/RDF/, last retrieved on March 01, 2010.

[12] Open Directory RDF Dump, http://rdf.dmoz.org/, last retrieved on March 01, 2010.

[13] Statement on ODP's *About* page, http://www.dmoz.org/about.html, last retrieved on March 01, 2010.

fication analyses and for integrating categorization information into our experimental data.

3.1.3 Google

Google[14] is currently the most popular search engine on the Internet[15]. What makes Google interesting for researchers is that it provides various technical interfaces for retrieving information from its databases that is very difficult – or even practically impossible – to compile with computing resources that are more limited than those of Google.

In our work, we refer to Google for two purposes: First, for estimating the *popularity* of a resource on the Web as measured by its *PageRank* [BP98], and second, for collecting any known *incoming hyperlinks* of a given resource (also called its *inlinks* or *backlinks*) [CDI98, FÖ99]. Both types of information rely on the retrieval and analysis of very large volumes of real-world data – predominantly full Web crawls and subsequent data analyses – of which the results are readily accessible from Google.

We consider the PageRank of a Web resource as a traditional measure of its popularity on the Web. We say "traditional" in this context because PageRank is based on an analysis of the Web graph whereas the popularity of a Web resource may also be estimated by other means, for example by an analysis of a collaborative tagging system such as Delicious and the social graph of its folksonomy (we propose such an approach in Chapter 5). PageRank is a well-studied link analysis algorithm that assigns a weighting to each element of a hyperlinked set of documents – such as the Web graph – with the purpose of estimating its relative importance within the set.

In this context, we have to note some important differences between theory and practice with regard to PageRank. While a Web resource's PageRank in the theory of Brin and Page [BP98] is a rational number, the PageRank PR of a resource r as returned by Google in practice is a natural number $0 \leq PR(r) \leq 10$, where higher numbers denote higher popularity. Additionally, the lowest PageRank with a value of zero (also called *PR0*) does not necessarily mean that the resource is unpopular [Sob02]: It is rather a special value that can have several different meanings, for example 1) the PageRank value is not yet calculated because it is a new Web resource in Google's search index; 2) the resource has been banned by Google (e.g., a spam or phishing Web page); or 3) the resource is considered as duplicate content.

For our experiments, we have developed custom applications based on the *Google SOAP Search API*[16] for retrieving PageRank information and incoming hyperlinks of a Web resource from Google. In the latter case, we use these lists of incoming hyperlinks

[14] Google, http://www.google.com/.

[15] While an authoritative source for such statistics is missing, various market reports such as http://comscore.com/Press_Events/Press_Releases/2010/1/comScore_Releases_December_2009_U.S._Search_Engine_Rankings and http://googlesystem.blogspot.com/2009/03/googles-market-share-in-your-country.html clearly rank Google Search before competitors such as Microsoft Bing or Yahoo! Search.

[16] Google SOAP Search API, http://code.google.com/apis/soapsearch/. Please note that as of December 5, 2006, Google stopped issuing new API keys for using its SOAP Search API.

as input data for collecting incoming anchor texts [McB94] of a Web resource through a targeted Web crawl as described in Chapter 4.

3.1.4 AOL500k

America On Line (AOL)[17] is a US company providing various types of Internet services. Particularly, it offers its own, AOL-branded Web search engine called *AOL Search* in cooperation with Google, i.e. "[search results] are administered, sorted and maintained by Google"[18]. The *AOL500k* corpus [PCT06] is one of the few publicly available large-scale collections of search queries. It consists of a random sample of 20 million Web queries collected from 650,000 users on *AOL Search* over three months in 2006. As the result of these search queries, about 1.6 million different Web documents were visited by users. What makes this collection so interesting for researchers is the vast amount of data, which "represents real world users, un-edited and randomly sampled" [PCT06].

AnonID	Query	QueryTime	Item Rank	ClickURL
8760	jojo the singer	2006-03-26 16:02:04	5	http://www.jojofan.com
8760	jennifer lopez	2006-03-26 16:05:29	4	http://www.allstarz.org
8760	jennifer lopez	2006-03-26 16:05:29	10	http://www.starpulse.com
8760	nicole richie	2006-03-26 17:28:58		

Table 3.1: **Excerpt of the AOL500k search query collection.** The data set includes {*AnonID*, *Query*, *QueryTime*, *ItemRank*, *ClickURL*} entries, where *AnonID* is an anonymous user ID number, *Query* is the query – i.e. search keywords – issued by the user (case shifted with most punctuation removed), *QueryTime* is the time at which the query was submitted for search. If and only if the user clicked on a search result, *ItemRank* and *ClickURL* information is available, which represent the rank of the clicked item in the search result list and the domain portion of the clicked item's URL, respectively.

In the research work described in this thesis, we use AOL500k for integrating data of search queries for Web documents into our experiments, particularly for our CABS120k08 data set (see Section 3.2.2 below). An excerpt of AOL500k is shown in Table 3.1.

3.1.5 The World Wide Web

The Web itself is another data source used in this thesis. In experiments that required, for example, access to the HTML source codes or incoming anchor texts of Web documents, we directly retrieved this information from the World Wide Web. Here, we built

[17] AOL, http://www.aol.com/.

[18] Official statement on the About page of AOL Search, http://search.aol.com/aol/about#webhome, last retrieved on March 01, 2010.

custom applications on top of the free and open source *Hadoop* platform[19]. Hadoop implements the concepts of Google's patented *MapReduce* framework [DG04] and allows for a distributed, parallel execution of programs on clusters of commodity hardware.

In our experiments, we have used Hadoop extensively for tasks in the areas of data retrieval, distributed data storage, and data analysis. Most of our Hadoop applications were run on a multi-node Hadoop cluster consisting of six physical machines[20] with a total of 24 cores running under a Linux operating system. During the course of these experiments, the author of this thesis also published several Web articles [Nol07c, Nol07b, Nol07a] on Hadoop that have since been contributed back into the Hadoop project and been used by various universities as references for students in the context of distributed programming[21].

3.2 Main Data Sets

In the following sections, we give a brief overview of the main experimental data sets that we have created and used for our research work described in this thesis.

3.2.1 DMOZ100k06

We published the DMOZ100k06 data set in two versions in June and August 2007, respectively [NM07a, NM08b]. Initially created in 2006, we subsequently enhanced the data set significantly over the course of 2007 through improved crawling and data extraction techniques, particularly with regard to folksonomy data retrieved with our DeliciousAPI library from the collaborative tagging system Delicious, whose technical restrictions on bulk data retrieval required several workarounds to overcome on our side (see Section 3.1.1). As we describe in detail in Chapter 4, we use this data set primarily for studying the volume and availability of folksonomy data about Web resources in practice, and how this user-contributed information compares to the contents and metadata of these resources as provided by their authors or publishers.

For the initialization of this data set, we randomly sampled 100,000 Web documents from the Web taxonomy of the Open Directory Project, which contained 4,818,944 Web documents in more than 590,000 categories at that time. We discarded any documents that could repeatedly not be downloaded from the Web, which yielded a final sample of 97,574 documents. Then, we retrieved over the course of four months various data about the sampled documents from the social bookmarking service Delicious (folksonomy data), the Open Directory Project (categorization information), the search engine Google (popularity), the Internet Content Rating Association[22] (content labels) and the

[19] Hadoop, http://hadoop.apache.org/, last retrieved on March 01, 2010.

[20] Machine specifications: Intel Xeon E5335 2.0 GHz Quad Core CPU, 4 GB of RAM, 150 GB RAID5 disk space, Gigabit network interface.

[21] For example, the articles have been used by the Instructional Support Group of the Dept. of Electrical Engineering & Computer Science at University of California, Berkeley [otDoEECS09].

[22] Internet Content Rating Association (ICRA), http://www.fosi.org/icra/. We present ICRA in Chapter 7.

World Wide Web (HTML content and metadata).

The size of the data set including HTML sources is about 1.6 Gigabytes. The data set and our studies of it are described in detail in Chapter 4. An overview is shown in Table 3.2. The data set is available for download at the homepage of the Hasso Plattner Institute[23] and the author's homepage[24].

Description	Number	Note	Folksonomy Symbol	Data Source
Total documents	97,574	100.0%		see text
Total users	165,192		$\mathcal{U}$	D
Total bookmarks	282,529		$P(\mathcal{F})$	D
Total tags	63,594		$\mathcal{T}$	D
Total tag assignments	758,242		$\mathcal{Y}$	D
Total categories	84,663			O
Total category assignments	115,458			O
*Categorized documents**	*97,574*	*100.0%*		
Bookmarked documents	18,220	18.7%		
Tagged documents	17,342	17.8%	$\mathcal{R}$	

**100% due to experimental setup*

Table 3.2: **Overview of the DMOZ100k06 experimental data set.** Data sources: Delicious (D), Open Directory Project (O). Data from the sources Google, Internet Content Rating Association and World Wide Web are not shown in this overview.

3.2.2 CABS120k08

We created and published the CABS120k08 data set in 2008 [NM08c]. As we describe in detail in Chapter 4, we use this data set primarily for studying the relations of posts in folksonomies provided by readers of Web documents, *hyperlink anchor text* provided by authors of Web documents, and *search queries* of users trying to find these documents on the Web. In other words, we use the data set to compare folksonomy data about Web resources with other types of metadata in the research domain of Web information retrieval.

For the initialization of CABS120k08, we constructed a sample list of Web documents by an intersection of the Web taxonomy of the Open Directory Project and the search query collection AOL500k[25] Only such documents were included in the sample that

[23] Hasso Platter Institute, http://www.hpi.uni-potsdam.de/.

[24] Michael G. Noll, http://www.michael-noll.com/wiki/DMOZ100k06.

[25] Because the publication of AOL500k was accompanied with strong privacy concerns, we discarded any AOL500k user IDs during data sampling. For more information about the issue of search query logs and user privacy, e.g. how information about "anonymous" users can be derived from query log analysis, we refer the interested reader to works such as [JKPT07, NS08].

were both searched for and subsequently visited (AOL500k) as well as categorized (ODP). Then, we retrieved over the course of three months data about the sampled documents from the social bookmarking service Delicious (*full* tagging information), the search engine Google (popularity as well as incoming hyperlinks) and the World Wide Web (HTML content and metadata; additionally, incoming anchor texts[26] from other Web documents). We removed any documents that could repeatedly not be downloaded from the Web, which yielded a final sample of 117,434 documents.

Description	Number	Note	Folksonomy Symbol	Data Source
Total documents	117,434	100.0%		see text
Total folksonomy users	388,963		$\mathcal{U}$	D
Total bookmarks	1,289,563		$P(\mathcal{F})$	D
Total tags	889,879		$\mathcal{T}$	D
Total tag assignments	3,383,571		$\mathcal{Y}$	D
Total searches	2,617,326			A
Total anchor texts	2,242,321			G,W
Total categories	84,663			O
Total category assignments	144,850			O
*Categorized documents**	*117,434*	*100.0%*		
*Searched documents**	*117,434*	*100.0%*		
Anchored documents	95,230	81.1%		
Bookmarked documents	59,126	50.3%		
Tagged documents	56,457	48.1%	$\mathcal{R}$	

**100% due to experimental setup*

Table 3.3: **Overview of the CABS120k08 experimental data set.** To prevent confusion, we explicitly denote users in this table as folksonomy users because the AOL500k search query corpus also includes (anonymized) user information. Data sources: AOL500k (A), Delicious (D), Google (G), Open Directory Project (O), World Wide Web (W). Popularity information retrieved from Google is not shown in this overview.

The size of the data set including HTML sources is about 4.4 Gigabytes. The data set and our studies of it are described in detail in Chapter 4. An overview is shown in Table 3.3. Coincidentally, the numbers of ODP categories are identical for the CABS120k08 and DMOZ100k06 data sets. The data set is available for download at the homepage of the Hasso Plattner Institute[27] and the author's homepage[28].

[26] Due to technical restrictions on the side of Google, we processed a maximum of 100 referring documents for retrieving and extracting incoming anchor texts per Web document in the sample.

[27] Hasso Platter Institute, http://www.hpi.uni-potsdam.de/.

[28] Michael G. Noll, http://www.michael-noll.com/wiki/CABS120k08.

3.2.3 SPEAR Collection

We created the SPEAR collection of data sets in 2009 [NAG+09, ANG+09]. As we describe in detail in Chapter 5, we use this collection for studying whether the expertise of users in a collaborative tagging system can be derived from their activities and implicit interactions in the folksonomy.

For the initialization of the SPEAR collection, we randomly sampled 110 "seed" tags (see Table 3.4) from a pool of tags that consisted of all the 200 "popular tags" reported by Delicious[29] as well as over 200 additional tags collected by monitoring the front page of Delicious. For each of these 110 tags, we created a separate data set by retrieving the Web resources that had been assigned the tag, and subsequently downloaded the bookmarking and tagging histories for each of these resources. The data collection process took three months until completion.

The size of the full collection is about 2.5 Gigabytes. The collection and our studies of it are described in detail in Chapter 5. An overview is shown in Table 3.5.

3d, admin, adobe, advertising, ajax, algorithms, api, apple, architecture, argument, art, articles, audio, bath, blogs, blogsjava, books, bridge, browser, business, car, cms, collection, comics, computer, convention, cooking, cool, culture, czaby, eShopping, economics, electronics, email, entertainment, environment, fashion, fic, film, finance, firebug, firefox, flash, flex, flickr, food, forum, framework, free, freeware, fun, funny, gallery, games, geek, google, government, graphics, green, guide, hal, hardware, health, history, home, hosting, house, howto, html, humor, icons, illustration, illustrator, images, imported, information, inspiration, interactive, interesting, internet, iphone, japan, java, javascript, jobs, jquery, kernel, kids, kubrick, language, later, learning, library, list, mention, nu, online, opera, sf, soap, sun, the, todo, tube, tutorial, ukquake, wine, wsj, xp

Table 3.4: **The 110 seed tags used for creating the SPEAR collection.**

Description	Number	Folksonomy Symbol	Data Source
Total seed tags $\mathcal{T}_0$ (with $\mathcal{T}_0 \subset \mathcal{T}$)	110	$\mathcal{T}_0$	see text
Total users	1,198,863	$\mathcal{U}$	D
Total bookmarks	15,987,386	$P(\mathcal{F})$	D
Total tags	809,167	$\mathcal{T}$	D
Total tag assignments	52,435,158	$\mathcal{Y}$	D
Total documents	132,165	$\mathcal{R}$	D

Table 3.5: **Overview of the SPEAR collection of data sets.** Data sources: Delicious (D).

[29] Delicious Popular Tags: http://delicious.com/tag/.

3.3 Summary

In this chapter, we have presented the major experimental data sources for the research described in this thesis, as well as the experimental data sets that we have created from these sources. In the next chapters, we will describe our different studies that make use of these data sets for the purposes of gaining new insights into folksonomies (Chapter 4) and leveraging folksonomies for information retrieval tasks (Chapters 5, 6 and 7).

If I have ever made any valuable discoveries, it has been owing more to patient attention, than to any other talent.

Sir Isaac Newton (1643–1727)

4 Exploring Folksonomies for Web Information Retrieval

Folksonomies and collaborative tagging provide a large volume of user-contributed data about resources on the Web. In addition to the recently emerged folksonomies, several other sources of data have already existed that are being actively used in the domain of Web information retrieval, for example the contents of Web resources themselves, metadata about these resources as provided by their authors, or users' search queries for resources as collected by Web search engines. When studying folksonomies for Web information retrieval, it is therefore essential to not only examine folksonomies by themselves, i.e. independently from their surroundings, but also to place them into context with other data sources on the Web.

In this chapter, we describe in detail our empirical and explorative studies of folksonomies in the context of Web information retrieval, investigate how much and what kind of data is available in practice, how it compares and relates to other types of data and metadata on the Web, and test our hypothesis with regard to the data contributed by users in folksonomies:

> **Hypothesis 1 (New Perspective on the Web)**:
> User-contributed data in folksonomies provides new, complimentary information about Web resources that is not available through traditional types of data and metadata on the Web, such as metadata contributed by the authors of these resources.

4.1 Types of Web Data and Metadata

In the following sections, we will describe the different types of data and metadata on the Web that we will analyze and compare in our subsequent experiments. An overview is presented in Table 4.1. Our terminology in this chapter with regard to posts in folksonomies, anchor text and search queries is listed in Table 4.2. We have already described in Section 2.2 that bookmarks in Delicious can be adequately modeled by our definition of *posts* in folksonomies. As such, we will use the term *social bookmark*, or

simply *bookmark*, interchangeably with "post" for the remainder of this thesis[1].

Name	Type	Who / Role	Where / Source	Section
Bookmarks	M	Web readers	Folksonomies	4.1.1
Textual content	D	Web authors (owner)	Web documents	4.1.2
HTML metadata	M	Web authors (owner)	Web documents	4.1.3
Anchor text	M	Web authors (others)	Web documents (others)	4.1.4
Search queries	M	Web searchers	Search engines	4.1.5
Classification	M	Experts	Web taxonomies	4.1.6

Table 4.1: **Analyzed data and metadata types**. The column *Type* differentiates between data (D) and metadata (M) with regard to resources, i.e. Web documents.

Item	Contains	Example
Bookmark (a.k.a. post)	Tags	{ `research`, `web`, `folksonomy` }
Anchor text	Anchor words	"`homepage of the HPI`"
Search query	Search keywords	"`hpi potsdam research`"

Table 4.2: **Terminology and examples for posts (bookmarks), anchor text and search queries.** As explained at the beginning of this section, we use the term *social bookmark*, or simply *bookmark*, interchangeably with "post".

4.1.1 Folksonomies and Tags

First and foremost, we investigate the data in folksonomies, which is contributed by users through collaborative tagging. We have already presented a comprehensive review of folksonomies and collaborative tagging in Chapter 2. As we have noted previously, folksonomies are driven by end users, i.e. the "readers" or "recipients" of Web documents. In contrast, other data types described in the following sections are provided by the authors or publishers of Web documents, or groups of expert users with certain domain knowledge. Folksonomies may therefore be considered as a new kind of Web metadata, which may provide a new perspective on the Web from the viewpoint of end users. We will analyze how much and what kind of data is available in folksonomies in practice, and how it relates to other types of metadata in the domain of Web information retrieval.

4.1.2 Document Content

Traditionally, Web information retrieval has relied on techniques that extract data from Web documents directly [Sin01, MRS08] – in other words, data that is provided by the

[1] Another reason is that prior research studies often uses the term *social bookmark* as well, thereby making it more convenient for the reader of this thesis to perform comparisons across works.

authors or publishers of Web documents. The first generation of search engines, for instance, examined mostly the text and formatting of a Web document's content – an approach that is actually very close to classic information retrieval [Bro02]. In the fields of classification and clustering of documents, *bag-of-words* and *n-gram* (also called "shingles") techniques are common [MRS08], and a plethora of refinements help to increase the performance of these techniques such as the use of stop words or term-weighting schemes like TF-IDF [SB88]. Unfortunately, such content-based approaches still suffer from the difficulties of automatically inspecting and understanding non-textual Web resources such as images, videos or Adobe Flash, and even textual data is not trivial to analyze given the huge amount and variety of content on the Web. While it is very easy for a human to analyze such content, it is a much harder task for machines even with modern processing power. For example, image processing algorithms may be able to identify human faces or nudity in images up to a certain reliability, but such techniques are often restricted to very specific problem domains [WWG05, RJB06, JR02]. Secondly, results of machine learning algorithms depend heavily on quantity and quality of training input, and training input may vary with a user's individual preferences and characteristics [DHS01]. An algorithm for binary classification, for instance, will not yield optimal results if it is not trained with a sufficient number of samples from both classes, even though training tricks such as *PEBL* [YHC02, YHC04] may help up to a certain extent.

In the past, researchers have tried to mitigate the problems of data extraction from Web documents with techniques that use additional sources of information. Next to analyzing special metadata about Web documents provided by their authors (see Section 4.1.3), these techniques may use Web-specific features such as incoming or outgoing hyperlinks of a Web document to infer information about the document and its neighbors [GGMP04, Kan04], a typical example being the *HITS* algorithm [Kle98] proposed by Kleinberg in 1998. Hybrid solutions combine content-based and link-based approaches, for instance by integrating the incoming anchor text of Web documents into the analysis (see Section 4.1.4). Since we similarly believe that folksonomies represent a new, complimentary source of information about Web documents, we will analyze how user-contributed tags assigned to Web documents compare to their content.

4.1.3 HTML Metadata

The traditional method of providing metadata about Web documents is described in the HTML and XHTML standards[2], which define elements and attributes for specifying metadata in the HTML source code of the documents for the purpose of helping users find relevant content on the Web. Embedding this information into the HTML source code of a Web document implies that it is provided by its authors or publishers. For example, authors should use the `TITLE` element to identify the contents of a document. While adding a title to a document is indeed common practice as we will see later, other HTML metadata such as `META KEYWORDS` or `META DESCRIPTION` is often omitted

[2] XHTML2 Working Group Home Page at W3C, http://www.w3.org/MarkUp/, last retrieved on March 01, 2010.

by Web authors. The most likely reason is that search engines like Google or Yahoo do not fully trust in and therefore may discard a large amount of HTML metadata because it is being abused by spammers [NNMF06]. For example, Google uses more than 200 signals extracted from the Web graph, ranging from the language of a Web document to the number and quality of other documents pointing to it [Cza09], but `META KEYWORDS` is not one of them [Cut09]. Still, HTML metadata is actively being used by Web authors [Hic05].

In our experiments, we will analyze how much and what kind of HTML metadata is available in practice, and compare it with metadata that is provided by the users of folksonomies, i.e. the readers of Web documents. We will focus our experiments on the `TITLE`, `META KEYWORDS` and `META DESCRIPTION` elements since they are those elements that are predominantly used in practice [Hic05].

4.1.4 Anchor Texts

As we have described above, early indexing, retrieval and ranking techniques in Web information retrieval relied mostly on on-page content of a Web document, i.e. text and formatting of its content. Nowadays, also off-page, Web-specific data such as link analysis and anchor text are used to infer information about a Web document and its neighbors [CDI98, Bro02, Kan04, KZ04]. Even though anchor text is not officially defined as *meta*data in the HTML standard (see Section 4.1.3), it is commonly used as such by Web search engines. It was first suggested in *WWW Worm* [McB94] in 1994, and the most prominent example of today is the *PageRank* algorithm [BP98] proposed by Brin and Page that powers their search engine Google.

Anchor text is defined as the text that appears within the bounds of an HTML `<A>` tag[3], i.e. the words associated with a hyperlink [KZ04]. The *incoming* anchor text of a Web document is the anchor text of any incoming hyperlinks (also called its "inlinks" or "backlinks" [CDI98, Fö99]) to the Web document. If, for instance, Web document *D1* is referenced by document *D2* through a hyperlink with the anchor text "HPI homepage", then "HPI homepage" is considered incoming anchor text of *D1*.

Anchor texts can therefore be exploited for associating terms with a Web document that are not part of the document itself. Additionally, a Web document's incoming anchor text is mainly created by *other* Web authors, i.e. it represents – similar to tags in a folksonomy created by Web readers – a different perspective on the Web document than its original author might have. For convenience, we will use the plural "anchor texts" when referring to multiple instances of anchor text throughout the remainder of this chapter.

4.1.5 Search Queries

Search query log files are a major source of data used in Web information retrieval for tasks such as user profiling and modeling search behavior [JP01, Bro02, LLC05], clas-

[3] For example, the anchor text of the hyperlink `<a href="labs.html">quux</a>` is "quux" and associated with the page `labs.html`.

sification of search queries [BJL+07], re-ranking of search results [ZC06] or extracting semantics [BYT07]. For instance, analyzing a user's past search queries may help to understand his goals, intents and interests in order to improve future searches – a user who is deeply interested in technology might deem search results related to Mac computers or the iPhone more relevant than those related to the fruit when searching for "apple". In other words, the search queries are mainly leveraged as data source for deriving *contextual information* about users and Web resources.

We focus our analyses of search queries in this thesis on the textual *search keywords* that users have specified when searching for information on the Web (in the example above, the word "apple"). While both tags and search keywords are provided by end users, it happens in different scenarios. In the case of Web search, users must judge the value of a Web resource in the search results *a priori*, i.e. before they actually visit the resource. Tagging a Web resource, on the other hand, is an explicit user action that is performed *a posteriori*, i.e. after having read or otherwise "processed" the resource.

A first hint at similarities between tags and search keywords is provided by Krause et al. [KJHS08]. They derive a "folksonomy" from Web search by extracting *(user, search keyword, resource)* triples from search query logs. They observed that the distribution between tags (derived from Delicious) and queries as well as resources is very similar, and that the clicking behavior of search engine users based on the displayed search results and the tagging behavior of users in a folksonomy is driven by similar dynamics. In this chapter, we will extend these studies by analyzing search keywords in various dimensions and also comparing them with user-contributed data in folksonomies and other Web-related metadata.

4.1.6 Classification

Another source of information about Web documents are taxonomies such as the *Open Directory Project* (ODP; see Section 3.1.2) and the *Yahoo! Directory*[4]. These databases provide a directory of Web resources organized into a fixed set of categories. As we have described in Chapter 3, we use the data of the Open Directory Project in our experiments as the ground truth for classification analyses and for integrating categorization information into our experimental data. Similar to the *Dewey Decimal Classification* system [OCL], the ODP is maintained by a community of domain experts who share a common strategy and policy for performing their work. We will compare this controlled, top-down expert classification of Web documents with the uncontrolled, bottom-up approach of folksonomies.

4.2 Experimental Setup

For our experiments, we have constructed and analyzed two large-scale corpuses of real-world data, *DMOZ100k06* and *CABS120k08*, which we have introduced in Section 3.2. Both comprise a variety of experimental data from several sources on the

[4]Yahoo! Directory, http://dir.yahoo.com/.

Web as shown in Table 4.3.

Data Source	DMOZ100k06	CABS120k08
Delicious	x	x
Open Directory Project	x	x
Google	x	x
Internet Content Rating Association	x	
AOL500k		x
World Wide Web	x	x

Table 4.3: **Comparison of data sources between DMOZ100k06 and CABS120k08**. In addition to the difference with regard to the selected data sources, the information extracted from these sources also varies between the two data sets as described in Section 3.2.

While both data sets have been built using similar data sources, there are notable differences between the two. Particularly, there are differences in the data sampling processes and in the time of creation. DMOZ100k06, which we created in 2006 [NM07a] and extended in 2007 [NM08b], is based on a random sample of Web documents from a single data source (ODP), whereas CABS120k08, which we created in 2008 [NM08c], is based on an intersection of Web documents from two data sources (ODP and AOL500k). For the initialization of DMOZ100k06, we randomly sampled 100,000 Web documents from the Web taxonomy of the ODP, which contained 4,818,944 Web documents in more than 590,000 categories at the time. The full data sampling process eventually resulted in 97,574 Web documents (see Section 3.2.1 for details on the construction of DMOZ100k06). For the initialization of CABS120k08, we built a sample list of Web documents by creating an intersection of the ODP and the search query collection AOL500k, which contains 20 million Web queries collected from 650,000 users who subsequently visited about 1.6 million different Web documents as a result of these queries. Only such documents were included in the sample list that were categorized (ODP) as well as searched for *and* actually visited (AOL500k). The full data sampling process eventually yielded a final set of 117,434 documents (see Section 3.2.2 for details on the construction of CABS120k08).

We have had several reasons for the creation and use of *two* such data sets. On the one hand, our decision to leverage AOL500k as a data source was mainly born out of necessity: Only a few large-scale data sets of real-world search queries have been published so far, which inevitably means that there do not exist many alternatives to choose from[5]. Of these, AOL500k has arguably been the most studied (cf. [PCT06,

[5]Releasing any kind of user data – such as a collection of search queries – is a delicate subject. For example, shortly after the release of the *AOL500k* corpus in 2006 (see Section 3.1.4), AOL spokesman Andrew Weinstein had to apologize publicly for the release [Arr06]. Even though there was no personally identifiable data provided by AOL with the data records, search queries themselves could be analyzed to infer such information. Eventually, the publication of AOL500k caused AOL to dismiss the two responsible scientists and the company's Chief Technology Officer, but has since also stimulated

JKPT07, JKHS08, KHS08, KJHS08, HKGM08, Rad09, NO09]). For these reasons, we have decided to use it for integrating user data from Web search into our experiments, namely in the form of the CABS120k08 corpus.

On the other hand, the AOL500k data set – and thus our CABS120k08 data set, which relies on the former – has some potential drawbacks for the research work described in this thesis. Firstly, AOL500k includes only the domain portion of Web documents, i.e. the "true" URLs of clicked search results are truncated. For example, the URL of the Wikipedia article on the World Wide Web

http://en.wikipedia.org/wiki/World_Wide_Web

would be shortened to

http://en.wikipedia.org

in AOL500k. This truncation negatively impacts the granularity of the data set with regard to Web documents and would make analyses such as the study of the spatial granularity of folksonomies impossible (see Section 4.3.3).

Secondly, AOL500k is solely comprised of Web documents displayed in search results, which by definition means that these documents are thought to be the most relevant and most popular documents on the Web that match the user's query. We may therefore expect that, for example, the mean popularity of Web documents in AOL500k (and our CABS120k08 data set) is higher than a truly random sample of documents on the Web. And indeed, we found that the average popularity of Web documents in CABS120k08 is higher than in DMOZ100k06: the mean PageRank values are $\mu_C = 3.93$ (standard deviation $\sigma_C = 2.45$) and $\mu_D = 3.13$ ($\sigma_D = 1.66$), respectively. Similarly, CABS120k08 does not contain any Web documents with a PageRank of zero (cf. Section 3.1.3).

For these reasons, we have decided to use the DMOZ100k06 data set in addition to CABS120k08 for our experiments described in this chapter. While it is still an open research question how to create a truly random, unbiased sample of documents from the Web [BHK+09, Sne06, BYG06], we argue that our decision to use a random sample of the Open Directory Project is a reasonable approach that balances our requirements of unbiasedness and of having access to categorization information of Web documents, which we need for a comparative analysis and evaluation of folksonomies with regard to classification tasks (see Section 4.1.6).

4.3 Experimental Results

In the following sections, we describe the outcomes of our experiments and discuss their implications. We will note differences between the analyses of the DMOZ100k06 and CABS120k08 data sets where necessary and appropriate. Whenever terms in our experimental data were analyzed or compared (e.g. tags or words in anchor texts), it

and enabled a number of research studies in the area of Web information retrieval.

was performed without case sensitivity, i.e. terms such as "Philosophy" and "philosophy" were considered the same.

4.3.1 Overview

The data sets DMOZ100k06 and CABS120k08 consist of 97,574 and 117,434 Web documents, respectively. Due to the creation processes of the data sets as described in the previous section, each document in DMOZ100k06 and CABS120k08 was categorized into at least one category. Additionally, each document in CABS120k08 was searched for at least once. We discarded the special tags `system:unfiled` and `imported` from all of our analyses described in the following sections – the former is automatically added by Delicious to bookmarks without user-provided tags, and the latter is automatically added to bookmarks during data import to Delicious from other applications[6] – but kept them in the data set as-is. A statistical overview of both data sets is shown in Table 4.4 and in Figure 4.1. Per-document statistics are shown in Table 4.6, and Table 4.5 lists the most popular tags for each data set.

In accordance with previous studies described in Section 2.4.4, we found power-law distributions for users, tags and resources in our data sets. However, since power laws in folksonomies have already been treated thoroughly in prior work, we will not discuss our respective results in detail in this thesis.

Description	DMOZ100k06 Number	DMOZ100k06 Note	CABS120k08 Number	CABS120k08 Note	Folksonomy Symbol
Total documents	97,574	100.0%	117,434	100.0%	
Total users	165,192		388,963		$\mathcal{U}$
Total bookmarks	282,529		1,289,563		$P(\mathcal{F})$
Total tags	63,594		889,879		$\mathcal{T}$
Total tag assignments	758,242		3,383,571		$\mathcal{Y}$
Total search queries	-	-	2,617,326		
Total anchor texts	-	-	2,242,321		
Total categories	84,663		84,663		
Total category assignments	115,458		144,850		
*Categorized documents**	*97,574*	*100.0%*	*117,434*	*100.0%*	
*Searched documents**	-	-	*117,434*	*100.0%*	
Anchored documents	-	-	95,230	81.1%	
Bookmarked documents	18,220	18.7%	59,126	50.3%	
Tagged documents	17,342	17.8%	56,457	48.1%	$\mathcal{R}$

**100% due to the creation processes of the data sets (see Section 3.2)*

Table 4.4: **Comparison of the DMOZ100k06 and CABS120k08 data sets.**

[6] A data import, for example, includes a user importing his bookmark collection from applications such as the Web browsers Microsoft Internet Explorer or Apple Safari.

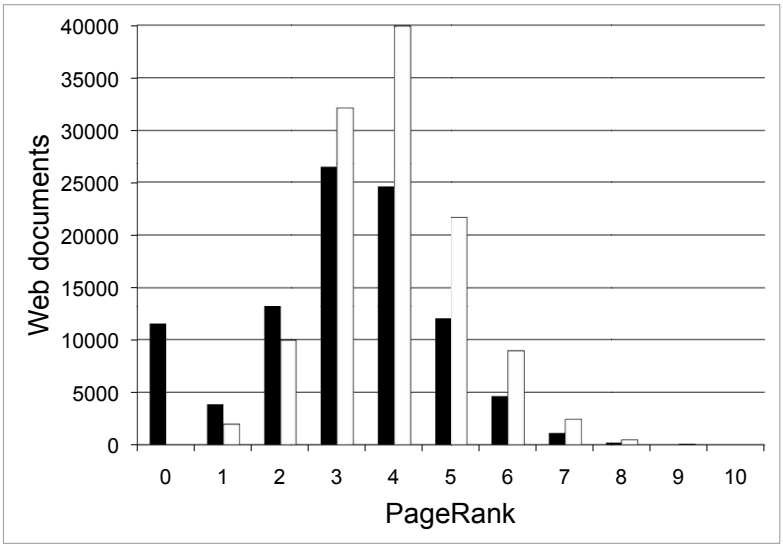


Figure 4.1: **PageRank distributions of DMOZ100k06 and CABS120k08.** The black and white bars denote the number of Web documents per PageRank for DMOZ100k06 and CABS120k08, respectively. DMOZ100k06 does not contain any documents with a PageRank of 10, whereas CABS120k08 does not contain any documents with a PageRank of 0.

	DMOZ100k06		CABS120k08	
Position	**Tag**	**Count**	**Tag**	**Count**
1	reference	16,282	news	77,661
2	software	14,263	reference	54,467
3	news	12,340	software	46,061
4	blog	10,921	shopping	41582
5	web	8,971	travel	40,795
6	tools	8,389	blog	39,172
7	design	8,233	design	38,252
8	search	7,843	music	37,704
9	opensource	7,347	tools	30,350
10	programming	7,075	art	29,140
11	css	6,733	web	25,005
12	webdesign	6,328	photography	23,575
13	video	5,583	politics	22,528
14	music	5,475	technology	21,746
15	art	5,439	business	20,388

Table 4.5: **Top tags of DMOZ100k06 and CABS120k08 by tag count.** The overlap of the Top tags is 60%, i.e. 9 out tags appear in both Top 15 lists.

Statistics per document	DMOZ100k06 Mean	DMOZ100k06 Std. dev.	CABS120k08 Mean	CABS120k08 Std. dev.
Bookmarks	2.90	71.35	10.98	69.98
Tags	1.90	17.28	7.07	28.90
Tag assignments	7.77	190.96	16.01	265.20
Categories	1.18	0.47	1.23	0.54
Category depth	6.66	1.90	5.82	1.91
Anchor texts	-	-	19.10	41.81
Search queries	-	-	22.29	476.28
PageRank	3.13	1.66	3.93	2.45

Table 4.6: **Per-document statistics of DMOZ100k06 and CABS120k08.**

Firstly, we can observe a significant difference in the total numbers between the two data sets with regard to folksonomy-related data. On the one hand, this can be explained by the different data sampling processes as described in Section 4.2 – the data sampling of CABS120k08 might favor documents that are more likely to be tagged by users. We have already noted that the mean popularity of documents in CABS120k08 is higher than in DMOZ100k06, which is also illustrated in Figure 4.1. Our experiments show that the more popular a Web document is with regard to its PageRank, the more likely the document is to be bookmarked and tagged (see Table 4.7 and the discussion further below), thereby supporting this argumentation. On the other hand, the difference in total numbers can also be attributed partly to the difference in the data sets' creation times. We constructed CABS120k08 several months after DMOZ100k06, so users in Delicious "had more time" to contribute tagging data to documents in the data set. We already made a similar observation between the initial version of DMOZ100k06 [NM07a] and its final version [NM08b] that we study in this thesis: Here, the number of social bookmarks – for the same sample of Web documents – increased by +32.6% during the course of three months. But even though there are some differences between DMOZ100k06 and CABS120k08 as described above, we will also see that both data sets exhibit similar characteristics in other data dimensions, for example the length of social bookmarks discussed in Section 4.3.4.

Secondly, we found that the amount of Web documents that have been tagged by users was surprisingly large for both DMOZ100k06 and CABS120k08, with relative frequencies of 17.8% and 48.1%, respectively. As such, collaborative tagging seems to cover already a considerable fraction of the Web. This is particularly interesting since the collaborative tagging system of Delicious had been in operation for only five years at the time we built our experimental data sets. Furthermore, if we follow the assumption that incoming hyperlinks and anchor texts are indications that a Web document is perceived as "important" or "interesting" by the referring party [BP98, ABC98], collaborative tagging even covered $P(tagged \mid anchor\ text) = 55.2\%$ of "relevant" Web documents in the CABS120k08 data set. This result is very encouraging with regard

to leveraging folksonomies for Web information retrieval, because it suggests that the user-driven folksonomies are indeed rich sources of information about the Web. Similarly, we observed that users strongly prefer to add tags to their bookmarks[7]. In other words, *if* users bookmarked a Web document, they also tagged it in almost all cases: 95.2% and 95.5% of bookmarks in DMOZ100k06 and CABS120k08, respectively, include tagging information. This is another promising result because it indicates that users do indeed see a benefit in the provisioning of metadata such as tags for annotating resources, and that tagging is not perceived as an additional burden on top of the cost of bookmarking (cf. [ABC98]). We have described possible explanations for this finding in Section 2.4.3, where we have discussed the user motivation and functions of tags in folksonomies.

Thirdly, there were strong positive correlations of a document's volume of tagging data with its popularity on the Web (as indicated by its Google PageRank), i.e. the more popular a document, the more likely it was to be tagged: Spearman-r [Coo06] were +0.99 and +1.00 for DMOZ100k06 and CABS120k08, respectively[8]. Table 4.7 lists the detailed results. Additionally, tagging activities were shifted towards lower PageRanks than bookmarking activities: In DMOZ100k06, the mean PageRank was 5.71 for tagged documents and 6.36 for bookmarked documents (CABS120k08 supports this finding with means of 5.58 and 6.09, respectively).

	DMOZ100k06		CABS120k08	
PageRank	**Bookmarked**	**Tagged**	**Bookmarked**	**Tagged**
0	0.031	0.029	-	-
1	0.021	0.019	0.100	0.090
2	0.034	0.030	0.157	0.138
3	0.088	0.080	0.314	0.289
4	0.228	0.215	0.525	0.498
5	0.441	0.425	0.731	0.710
6	0.646	0.633	0.870	0.857
7	0.809	0.804	0.926	0.918
8	0.906	0.886	0.965	0.958
9	0.955	0.955	0.981	0.981
10	-	-	1.000	1.000

Table 4.7: **Relative frequencies of bookmarked and tagged documents in DMOZ100k06 and CABS120k08 by PageRank.** For instance, 52.5% of documents with a PageRank of 4 were tagged in CABS120k08.

A popular argument put forth in favor of the "power" of the Social Web in the context of information retrieval is that user collaboration and contribution are supposed to help

[7] As we have noted previously, the provisioning of tags for bookmarks on Delicious is optional and by no means required for the user.

[8] Kendall-τ [Coo06] were +0.95 and +1.0 for DMOZ100k06 and CABS120k08, respectively.

with regard to retrieving "rare gems" from the masses of documents on the Web that search engines miss to identify as such. For example, even though an interesting Web document might not be indexed or ranked highly enough by search engines, word-of-mouth propaganda among users through email, social bookmarking or other means could eventually direct visitors to it (a scenario commonly quoted for the blogosphere). Our findings however suggest that users tend to focus their tagging activities on Web documents that are already popular, and less on unpopular ones. For example, the majority of Web documents in the DMOZ100k06 data set, 52.3%, has a PageRank of 3 or 4 but receives only 17.6% of all tags. We therefore argue that folksonomies are less suited for the scenario described above – at least from a global point of view[9].

The results described above could also be an indication that the ranking models and algorithms of search engines such as Google or Yahoo! are quite capable of identifying interesting and relevant resources for Web readers even though their algorithms such as PageRank [BP98] or HITS [Kle98] are based on information provided by the *authors* of Web documents (e.g. by regarding hyperlinks to other resources as an indication of their importance). On the other hand, it is also possible that the PageRank of a highly tagged document will eventually increase over time because it is a center of attention in a folksonomy. Unfortunately, our experimental data does not include historical information of a document's PageRank. Hence, we cannot verify this claim for the work described in this thesis and have to leave it up to future research.

In summary, we found evidence in this section that folksonomies provide large volumes of data about Web resources and already cover a considerable fraction of the Web. Users seem to be willing to contribute such metadata readily via collaborative tagging. Interestingly, we observed strong correlations of user activity and resource popularity, i.e. users focus their tagging activities on documents that are popular on the Web. In the following sections, we will therefore pay closer attention to the impact of resource popularity on experimental results.

4.3.2 HTML Metadata and Tagging

In this section, we compare the availability of metadata about Web documents as provided by their authors through HTML elements with metadata provided by their readers through tagging. We define *availability* as the share of Web documents with metadata (or tagging data, respectively) in the total set of documents. Figure 4.2 shows the results of our analysis for the DMOZ100k06 data set. A first observation is the relatively stable availability of HTML metadata across all PageRanks. The most frequently used element was the `TITLE` element with a micro-average of 97.14%[10]. `META KEYWORDS` slightly outperformed `META DESCRIPTION`, which agrees with the results of [Hic05]. Both `META KEYWORDS` and `META DESCRIPTION` occurred with a frequency of around

[9] Locally, users may form networks or users groups on collaborative tagging systems and collectively share references to interesting resources, i.e. the "rare gems" described in the text, with their network of friends.

[10] This result correlates with the findings of Hickson [Hic05]. While exact numbers are not given in the study, "the overwhelming majority of pages specify [the title element]".

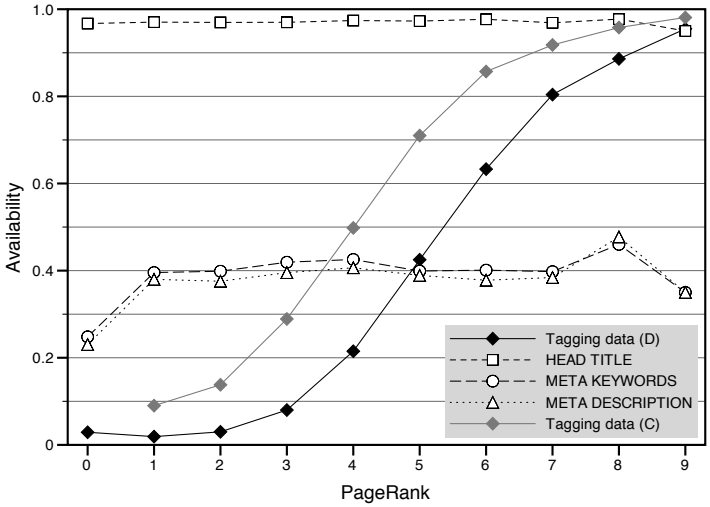


Figure 4.2: **Web authors versus Web readers.** Availability of HTML metadata about Web documents provided by their authors compared with tagging data provided by folksonomy users in DMOZ100k06 (D). The gray line denotes the tagging results of CABS120k08 (C) for comparison across our two experimental data sets. Of all PageRank 5 documents in DMOZ100k06, for example, 39.9% contained `META KEYWORDS` information and 42.5% were tagged.

40% in the data set, with two exceptions: Web documents with a PageRank of 0 (*PR0*) had a lower frequency of around 25%, and Web documents with PageRank 8 (*PR8*) had a higher frequency of around 47%.

A possible explanation for the lower availability of `META DESCRIPTION` and `META KEYWORDS` for *PR0* documents is that these documents might have been assigned a PageRank of 0 intentionally by Google for syntactic deficiencies such as improper composition, faulty document structure or erroneous markup. In other words, the lack of `META KEYWORDS` and `META DESCRIPTION` might be cause *and* effect. The frequency peak at PageRank 8 is harder to explain, and we are unsure yet how to interpret it correctly. We conducted a second, independent test[11] for verification, which confirmed this peak at *PR8*. We still have to find out what might cause this peak.

In summary, the availability of metadata that is provided by the authors of Web documents is relatively stable across all PageRanks. In folksonomies, however, the availability of tagging data increases with a document's popularity on the Web. This suggests that author-supplied HTML metadata is – on a quantitative level – a richer source of data for less popular documents, whereas folksonomies contain more data about pop-

[11] We built a different sample of 341 *PR8* documents. An analysis showed a similar trend with regard to the availability of HTML metadata. Here, the result was 49.9% for both `META KEYWORDS` and description, where each document with `META KEYWORDS` also had a `META DESCRIPTION` and vice versa. Additionally, we tested 105 documents with a PageRank of 9, which resulted in 46.7% for both `META KEYWORDS` and `META DESCRIPTION`. Again, we found the same strong correlation between availability of `META KEYWORDS` and `META DESCRIPTION` as for *PR8*.

ular ones. We will follow up this analysis in Section 4.3.5, where we will investigate how much *new* data about Web documents is provided by folksonomies by comparing tagging data with the HTML sources of documents, which include HTML metadata.

4.3.3 Spatial Granularity of Tagging

In this section, we examine whether users in folksonomies tend to tag documents higher up or deeply within a Web site's document hierarchy. Here, we analyze the URL[12] of tagged Web documents. URL schemes such as HTTP for Web documents contain names that can be considered hierarchical, and the components of the hierarchy are separated by a "/" delimiter character. We therefore based the calculation of a URL's depth on the "/" separator so that a top-level URL such as

http://www.example.com/

would be assigned a depth of 0 (zero), whereas a URL such as

http://www.example.com/**path**/**file.html**

would be assigned a depth of 2, and so on. Since the CABS120k08 data set contains by definition (see Section 4.2) only top level Web documents with a depth 0, we used the DMOZ100k06 data set for this experiment. The results are shown in Table 4.8.

URL depth	Mean	Std. dev.
All Web documents	1.06	1.74
Bookmarked Web documents	0.48	1.06
Tagged Web documents	0.48	1.05

Table 4.8: **Spatial granularity of folksonomies.** Mean URL depths of Web documents including standard deviations in DMOZ100k06.

We observed that users tend to tag top-level Web documents rather than those documents located deeply within a Web site's hierarchy. This result is interesting because intuitively one might think that users would be more likely to add those Web documents to their personal collections that are harder to (re-)find or access. Documents with deeper URLs, however, are often more complicated to navigate to, for example because it takes longer to traverse a Web site's hierarchy or to manually enter their full URL. Our results suggest that tagging data in folksonomies gravitates towards the entry or top-level pages of Web sites. For example, this outcome means that an application that needs to work on deeper Web documents might require additional data sources,

[12] RFC 3986 "Uniform Resource Identifier (URI): Generic Syntax", available at http://www.ietf.org/rfc/rfc3986.txt, last retrieved on March 01, 2010.

or employ techniques that infer information from their "parent" documents, for which a larger volume of tagging data is likely available.

A comparison of our two data sets DMOZ100k06 and CABS120k08 supports this finding on spatial granularity of folksonomies. The CABS120k08 data set comprises only top-level Web documents (depth 0). While it contains only +20% more Web documents than DMOZ100k06, the total number of tag assignments in CABS120k08 is $|\mathcal{Y}_C| = 3,383,571$ compared to $|\mathcal{Y}_D| = 758,242$ for DMOZ100k06, i.e. a relative difference of about +350%. Similarly, the percentage of tagged documents in CABS120k08 and DMOZ100k06 are 48.1% and 17.8%, respectively.

In summary, we found in this section that users in folksonomies focus their tagging activities on resources higher up in a Web site's hierarchy, particularly the entry pages and homepages. The majority of tagging data will therefore be available for these top-level documents. However, information retrieval techniques that want to target deeper documents could augment the data of these documents with folksonomy-derived information about the parent documents higher up in the hierarchy. For example, a deep Web document describing a multimedia playback device could be identified as iPod[13] product information if parent documents were primarily tagged with `apple`, `music`, `itunes`.

4.3.4 Cardinality

The length of a search query, i.e. the number of search keywords per query, has been studied in the past and reported as being rather short with 2.x keywords on average [SWJS01]. Similar results have been reported for the number of words in anchor text [EM03]. We were interested in comparing the length of social bookmarks in folksonomies with search queries and anchor texts. We define the length of a bookmark to be the number of its tags, and the length of an anchor text to be the number of its anchor words. These lengths can be considered as the per-item *cardinality* of each metadata type, which indicates how much data is provided by a single "data unit" of each metadata type.

We found that the micro-averages of the length of searches, bookmarks and anchor texts in the CABS120k08 data set were $\mu_S = 2.89$, $\mu_B = 2.49$ and $\mu_A = 2.43$, respectively[14]. As comparison, the length of bookmarks in DMOZ100k06 was $\mu_B{}^* = 2.51$.

While it may seem at first glance that the length of bookmarks and anchor texts are almost equal, we found that the lengths varied significantly by the popularity of Web documents (indicated by their Google PageRank) as shown in Figure 4.3. There were strong negative correlations with document popularity for search queries and anchor texts: Spearman-r are -0.82 and -0.81, respectively. On the other hard, there was a positive correlation with document popularity for bookmarks: Spearman-r was +0.67[15]. This result suggests that anchor texts provide a larger amount of data for less popular Web documents whereas social bookmarks do so for more popular documents, with

[13] The iPod is a brand of portable media players made by Apple Inc.

[14] Wetzker et al. report an average length of 3.16 for bookmarks in [WZB08].

[15] Kendall-τ for search queries, anchor texts and social bookmarks were -0.64, -0.73 and +0.47, respectively.

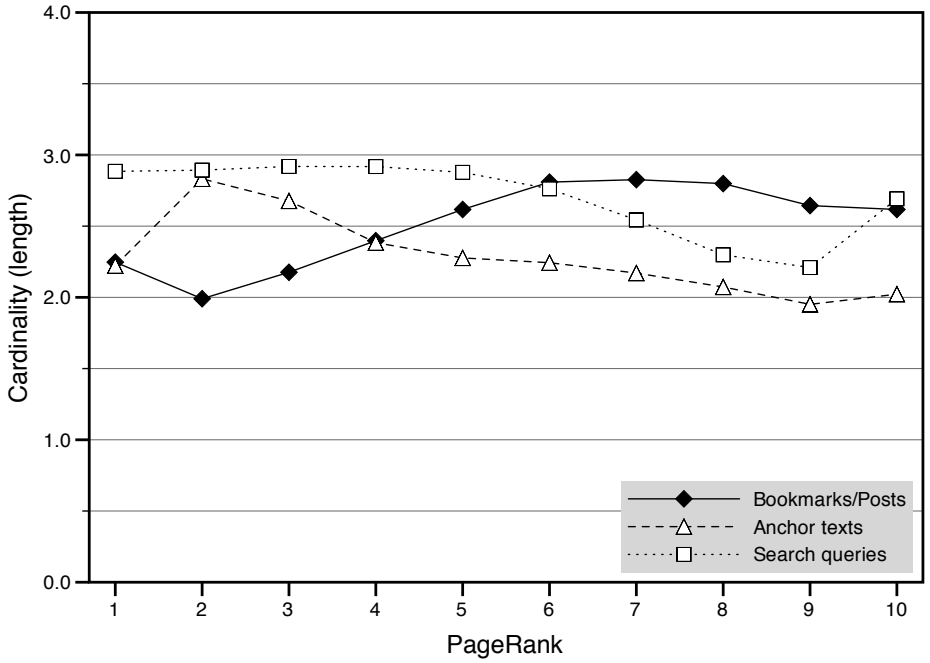


Figure 4.3: **Cardinality of metadata.** Average length of bookmarks, anchor texts and search queries in CABS120k08.

the break-even point being at PageRank 4. In the CABS120k08 data set, the amount of data provided by the average anchor text is larger for 37% of documents (*PR1–PR3*) compared to 29% of documents in the case of social bookmarks (*PR5–PR10*). So in direct comparison, anchor texts "win" in the first third of the cases, draw in the second third, and lose in last third; vice versa for bookmarks.

Looking at searches, the average search query dominated anchor text across all PageRanks. Compared to social bookmarks, search query length has a break-even point with bookmarks at *PR6*, and a second at *PR10*[16]. Here, the average search query provided more data for 90% of documents compared to only 3% of documents for social bookmarks. On the other hand, the data provided by social bookmarks and anchor texts has a stronger connection to the targeted Web documents than search query data, which might offset the quantitative advantage of the latter. As we have noted previously, a user may click on a search result but he may not know in advance whether the search result really satisfies his information needs, i.e. the association of search keywords and clicked result documents are rather weak. Tagging a document or hyperlinking a document (including anchor text), however, represent actions that generally happen after having read or otherwise "processed" the document [HRS07].

Finally, we observed that the average lengths of all metadata types stayed between

[16]The interpretation of the break-even point at *PR10* should be treated with care since CABS120k08 contains only five Web documents with *PR10*.

two and three terms – even when taking variations due to document popularity into account. While we find it difficult to explain – we hypothesize that it might have a cognitive explanation – humans seem to prefer using only two or three terms per action even across different problem domains (collaborative tagging, hyperlink creation, searching the Web).

4.3.5 Novelty

A lot of tasks in information retrieval employ techniques to extract data from Web documents, for example for indexing or classification purposes. On the other hand, not all information is captured by the terms in a document's HTML source, which includes its textual content and HTML metadata. Without further techniques such as anchor text analysis or latent semantic indexing [DDL+90], a Web search for "biology" would not turn up any documents where the term "biology" does not appear in the document content [Kle98].

In this section, we analyze how much new data is provided by social bookmarks, anchor texts and search queries – in other words, data that is *external* to a Web document. We are interested in finding out how much each metadata type is suited to add new information to Web documents, and thus how much it could help to improve information retrieval tasks in the context described above.

We define *novelty* as the percentage of unique terms of a document that are not already present in its content. The terms for social bookmarks are represented by the set of unique tags aggregated over all bookmarks of a document, i.e. if multiple users add the tag `research`, it is counted only once. The terms for anchor texts (unique anchor words) and search queries (unique search keywords) are defined similarly. The corresponding document is represented by the set of unique terms in its textual content (i.e. text within its `BODY` element) and HTML metadata (`TITLE`, `META KEYWORDS` and `META DESCRIPTION`). The results are shown in Figure 4.4.

Firstly, we observed that the amount of new information provided by any metadata type stayed below 7% of novelty for about 90% of documents in the CABS120k08 data set (*PR0–PR5*). Considering that we introduced a uniqueness requirement for terms, which effectively reduced the total number of terms per metadata type and terms per document in our experiment, this result is actually promising – about 1 of 20 unique terms is new to a document (see the discussion on the study of Bischoff et al. [BFNP08] below). Search keywords dominated tags which in turn dominated anchor words. Interestingly, the curves of search keywords and tags showed similar behavior: Both increased with a document's popularity (indicated by its Google PageRank), with larger increases starting at *PR6*. Novelty for words in anchor text basically stayed below the 5% threshold with a small peak at *PR10*.

Secondly, we found that tags provided more new data than anchor words. This indicates that tags are a better source for new data, particularly for popular Web documents. However, we will report in Section 4.3.7 that the similarity of tags and anchor texts was relatively low in the CABS120k08 data set, which indicates that they provide different kinds of information. We therefore argue that if one is interested in collecting new data,

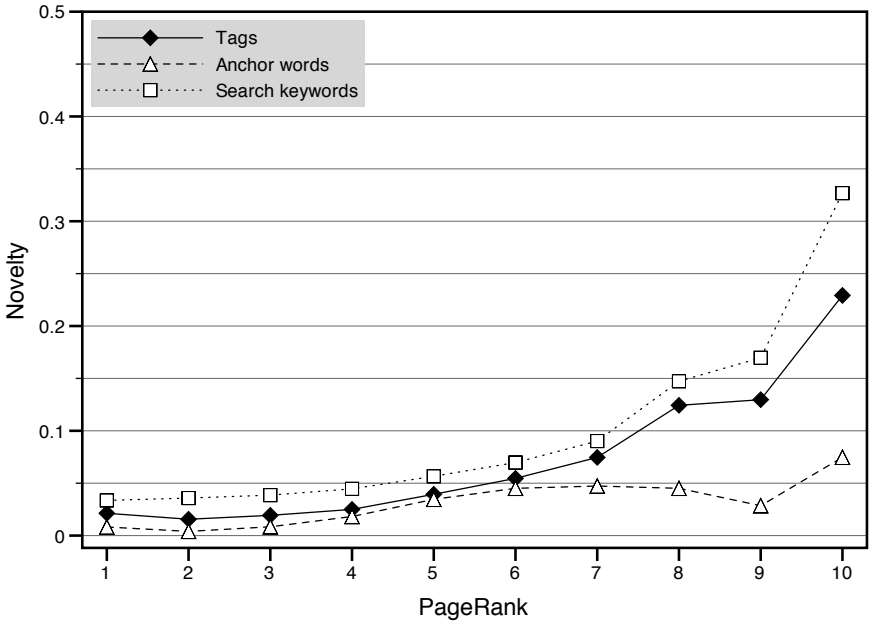


Figure 4.4: **Novelty** Mean percentage of new data provided by a document's tags, anchor words and search keywords in CABS120k08. On average, for example, 7.5% of tags of a PageRank 7 document are not present in the document.

one should in fact consider evaluating both metadata types.

We conclude that the metadata provided by social bookmarks, anchor texts and search queries provide new information about Web documents. This observation is similar to the results of Heymann et al. [HKGM08] and Bischoff et al. [BFNP08]. However, both of these studies report significantly higher numbers of about 45-50% for novelty of information contributed by tags. Unfortunately, the exact processes of matching tags with a document's source were not described in detail, which makes it difficult to explain the differences in experimental results. We could reproduce such high numbers for novelty, however, when a) we restricted tagging information to the most popular tags of Web documents, i.e. when the long-tail of rarely used tags was excluded from analysis, or b) we relaxed the uniqueness requirement for tags during the matching process, i.e. counting multiple matches of the same tag multiple times. We therefore argue that our results agree with theirs in principle.

4.3.6 Diversity

In this section, we study the inherent diversity of information provided by folksonomies, anchor texts and search queries. This analysis is related to the research of the consensus of users in folksonomies as described in Section 2.4.4. As we have noted in that section, when we talk about emergent consensus and stabilization of folksonomies, we are mainly referring to such tags that have managed to "escape" the long tail of tag

distributions for both users and resources. In this section, however, we conduct an indiscriminate analysis of tagging data, i.e. including the long tail.

Generally, we can assume that users do not collaborate when searching the Internet, or when creating Web documents with hyperlinks and anchor texts to other pages. While there is a collaboration aspect for social bookmarking and tagging in folksonomies, it is only one facet of many (see Section 2.4.3).

For our analysis, we have used the measure of *entropy* [Sha48] from information theory to determine the diversity of information with regard to Web documents. In the context of folksonomies, a document's tags and their tag counts can be considered as a "tag histogram", and the entropy E of such an histogram can be computed by

$$E(d) = -\sum_{t_i \in \mathcal{T}(d)} p(t_i|d) \log p(t_i|d) \tag{4.1}$$

where $\mathcal{T}(d)$ is the set of tags with which document d has been annotated, and $p(t_i|d)$ is the probability of d being annotated with tag t_i. We used the observed tag counts in the CABS120k08 data set to estimate the probabilities $p(t_i|d)$. We defined similar entropies for anchor texts (anchor words and their counts) and search queries (search keywords and their counts). Finally, we normalized entropy values so that zero entropy was represented by 0 and maximum entropy by 1. The results of our analysis are shown in Figure 4.5.

Firstly, we found strong negative correlations with document popularity (indicated by Google PageRank) for all metadata types: Spearman-r for tags, anchor words and search keywords were -0.96, -0.87 and -0.99, respectively[17]. With increasing document popularity, the diversity of information decreased, thus becoming more uniform. For tags, this finding also supports the notion that a tagging "consensus" emerges the more people are tagging a resource (see Section 2.4.4) because, as we have noted previously, the volume of tagging data increases with a document's popularity on the Web.

Secondly, we observed that search queries showed the highest diversity. The reason could be that searching the Internet is arguably the most "volatile" user action in our study. In contrast, users create bookmarks or hyperlinks with anchor text only *after* reading a document *and* perceiving it as useful. This process seems to serve as a kind of "noise filter" which search queries are lacking, supporting the results of studies such as [HRS07]. Similarly, users do not only have problems with finding relevant information on the Web per se, they also have problems with formulating good search queries [SJWS02]. Additional effects such as users becoming accustomed to automatic spell correction by search engines might further increase the diversity of search queries.

Thirdly, we found that tags were generally more diverse than anchor texts. On the one hand, this result suggests that tags are noisier than anchor texts and therefore potentially less useful. On the other hand, studies such as [BXW+07] report that tags provide multi-faceted summaries of Web documents. Seen this way, the diversity of tags

[17] Kendall-τ for tags, anchor words and search keywords were -0.91, -0.78 and -0.96, respectively.

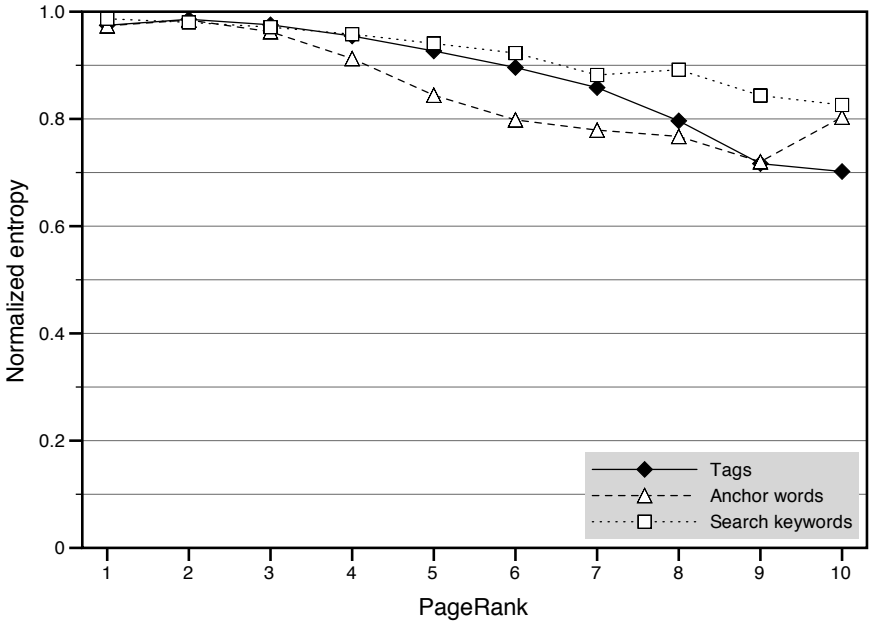


Figure 4.5: **Diversity.** Normalized entropy of tags, anchor words and search keywords in CABS120k08. A value of 0 denotes zero entropy (uniformity), a value of 1 maximum entropy (high diversity).

could be an advantage since it might capture information and meanings that anchor texts miss.

In summary, these results suggest that folksonomies and collaborative tagging do provide valuable information but it is very important to separate signal from noise. A simple way to do so would be applying thresholding techniques (e.g. excluding such tags that are not exceeding a minimum count) or considering only the Top n tags [LBY+07, RGMM07, Sim08] – thereby exploiting the power-law patterns observed in folksonomies to limit the potentially negative effect of the long tail (see Section 2.4.4). A more sophisticated approach would be, for example, to study the structure and dynamics of folksonomies for identifying expert users, thus adding a trust layer on top of folksonomies and collaborative tagging.[18] In Chapter 5, we will investigate such a notion of expertise or "trustworthiness" of users in folksonomies and propose an algorithm, *SPEAR*, for ranking users by their expertise.

4.3.7 Similarity

In this section, we study the pairwise relatedness of social bookmarks, anchor texts and search queries, i.e. how similar each metadata type is *to the others*, by analyzing the CABS120k08 data set. We also use categorization information from the Open Directory

[18] Interestingly, this discussion is related to the analysis of the Web graph for identifying link farms of spam Web pages [WD05].

Project as ground truth to investigate how much each metadata type is suited for classification tasks, thereby extending the related studies in [BXW+07, WZY06, XBCY07].

For similarity analysis, we transformed each metadata record into vector space and then computed their *cosine similarity* [DHS01], which is a similarity measure often used in information retrieval [MRS08]. Cosine similarity measures the similarity of two vectors $\vec{i}$ and $\vec{j}$ by finding the cosine of the angle between them. It is formally defined as

$$similarity(\vec{i},\vec{j}) := \frac{\vec{i}\cdot\vec{j}}{\left\|\vec{i}\right\| * \left\|\vec{j}\right\|} \tag{4.2}$$

where "$\cdot$" denotes the dot product of the two vectors, and the denominator is the product of their Euclidean lengths. Firstly, we preprocessed our experimental data – tags, anchor words, search keywords and ODP categorizations – as has been done in related studies such as [PCT06, TG06, XBF+08], namely by splitting terms at special characters such as "_" or ":", and by stemming the resulting terms based on Porter's stemming algorithm [Por80]. We also removed common English stop words such as "the" or "of" from the data. For example, the tag `new_york` would be preprocessed to the words `new` and `york`. After these steps, each word was treated as one dimension in the vector space for similarity computation. The final results are shown in Table 4.9.

	T	**A**	**S**	**C**
T	x	0.126	0.126	**0.189**
A	0.126	x	**0.193**	0.103
S	0.126	**0.193**	x	0.102
C	**0.189**	0.103	0.102	x

Table 4.9: **Similarity.** Pairwise similarities of tags (T), anchor words (A), search keywords (S) and categories (C) in CABS120k08. The maximum values for each column are in bold font.

We observed the highest similarities between tags and categories (0.189) as well as between anchor words and search keywords (0.193)[19]. This direct comparison suggests that tags are better suited for classification tasks whereas anchor words are better for augmenting Web search[20]. We will therefore analyze tagging information with regard to classification tasks in the next section. On the other hand, this result does not necessarily mean that tagging information of folksonomies can principally not be leveraged to improve Web search. For example, Au Yeung et al. [AGS08a] present an approach

[19] A statistical test revealed that the similarity means for (A, C) and (S, C) were significantly different for $P<0.05$. For (A, T) and (S, T) however, the null hypothesis of having equal means could not be rejected.

[20] These results also agree with the studies of Heymann et al., who analyzed the similarity between tags and search keywords in 2008 [HKGM08].

that exploits folksonomies for Web search disambiguation by classifying Web documents in search results into different thematic categories. Similarly, we propose a new approach to personalization of Web search in Chapter 6 that re-ranks search results lists based on the similarity of user and document profiles derived from folksonomies. We therefore argue that folksonomies provide information that can still help in the broad area of Web search, but it is important to understand how their characteristics – for instance, a closer relatedness of user-contributed tags to classification information from taxonomy experts than to search queries from other end users – can contribute to solving a given problem.

4.3.8 Classification

We observed in the previous section that tags seem to be better suited for classification of Web documents than anchor words or search keywords. In this section, we extend this analysis and study how each metadata type compares with the Web taxonomy of the Open Directory Project, which is maintained by a global community of expert editors.

For this experiment, we matched tags, anchor words and search keywords of a document in CABS120k08 against its categorization. A document in ODP is categorized by one or more category hierarchies such as *Arts* > *Crafts* > *Textiles* > *Weaving*. We analyzed at which hierarchy depth matches occurred, and normalized the results so that the top category in a hierarchy, e.g. "arts", was represented by 0 and the leaf category by 1, e.g. "weaving". Additionally, we used the Levenshtein distance [Lev66] to relax the matching conditions in order to detect small variations such as singular-plural ("dog" vs. "dogs") or different languages ("music" vs. "músika") to a certain degree. The results are shown in Figure 4.6.

Firstly, there were strong negative correlations of category depth with document popularity for all metadata types: Spearman-r for tags, anchor words and search keywords were -0.99, -0.84 and -0.99, respectively[21]. With increasing document popularity, broader classification scores were achieved. This seems to indicate that popular websites cover rather broad topics whereas less popular websites are rather focused[22].

Secondly, tags were used for broader classification than anchor words and search keywords across all PageRanks: The global average for matches of tags was 0.27 compared to 0.41 and 0.43 for anchor words and search keywords, respectively. Under relaxed matching conditions, tags score 0.38 compared to 0.47 for both anchor words and search keywords. This outcome supports the conclusion of the previous section that tags are better suited for classification purposes than anchor words or search keywords in the sense that they can better catch the "aboutness" of documents (cf. [GH06, BXW+07, EM03]). Similarly, our results suggest that for Web information retrieval in general, tags may help more with broad classification or grouping of documents rather than finding the specific "needle in the haystack".

[21] Kendall-τ for tags, anchor words and search keywords were -0.96, -0.73 and -0.96, respectively.

[22] For example, among the *PR10* Web documents in CABS120k08 were WhiteHouse.gov and NASA.gov, compared to Web documents such as WomencareShelter.org or LakeGeorgeRestaurants.com for *PR3*.

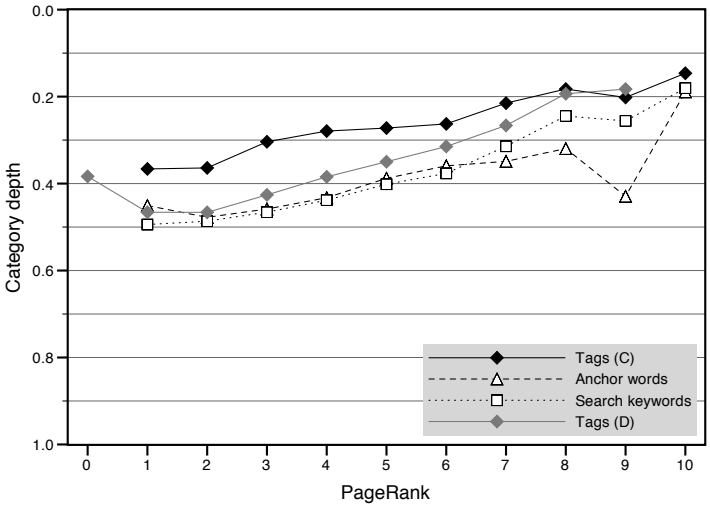


Figure 4.6: **Classification.** Normalized category depth for matches of tags, anchor words and search keywords with categories in CABS120k08 (C). A value of 0 denotes a root category ("broad"), a value of 1 a leaf category ("specific"). The gray line denotes the tag results of DMOZ100k06 (D) for comparison across our two experimental data sets. Please note that the y-axis is plotted in reverse so that root categories are at the top in the figure.

Lastly, we conducted a second experiment for analyzing whether tags that are used for the purpose of classification tend to be particularly popular among users when annotating resources on the Web. As such, we contribute to the studies of user motivation and the functions of tags described in Section 2.4.3. For this experiment, we matched the tags of a Web document with its ODP categories assigned by expert editors. We measured the popularity of a tag by its tag count, i.e. how many times the tag was used to annotate a given document, and normalized the tag count so that the least popular tag of a document was represented by 0 and the most popular tag by 1. As result of our experiment, we found that the mean popularities of tags matching a document's categories were 0.71 and 0.74 in DMOZ100k06 and CABS120k08, respectively. This outcome indicates that the use case of classifying Web resources via tags is particularly popular among users, and that an analysis of tag popularity can help with identifying those tags that provide the most relevant classification information. It also supports the notion that the usage patterns of collaborative tagging (see Section 2.4.3) lead to the emerging classification scheme of folksonomies.

4.4 Discussion and Summary

In this chapter, we have presented our empirical and explorative studies of folksonomies in the context of Web information retrieval. Our experiments and analyses suggest that user-contributed data in folksonomies exhibits several unique characteristics when

compared to other types of data and metadata about Web resources, providing strong support to our hypothesis that we presented at the beginning of this chapter. Where a direct comparison is available, our results agree with studies that analyzed the various data and metadata types in isolation, which we believe is an indication of the soundness of our experimental setup. Our main findings can be summarized as follows.

- Folksonomies provide large volumes of (meta)data about Web resources and already cover a considerable fraction of the Web. Users seem to be willing to contribute such metadata readily via collaborative tagging.
- Folksonomies provide new data that is not available through content inspection or link analysis of Web resources, which includes textual content and metadata written by their authors. We observed similar results for search queries (by Web searchers) and anchor texts (by other Web authors). Additionally, our similarity analyses found that the information in folksonomies is different from other metadata types, which suggests that folksonomies are indeed providing complimentary data for Web information retrieval.
- In general, the data about individual resources in folksonomies is rather diverse. Techniques for separating "signal from noise" are therefore helpful when leveraging folksonomies for Web information retrieval. For starters, even simple techniques such as thresholding may be effective according to our experiments. However, more sophisticated approaches will yield more satisfying results (cf. Chapter 5).
- The cardinality of posts in folksonomies, search queries and anchor texts are similar: the average lengths of all three metadata types stayed between two and three terms. However, we also found that posts in folksonomies are the only metadata type that shows a positive correlation with resource popularity. In direct comparison, for example, the amount of data per anchor text is larger for less popular resources, whereas posts in folksonomies provide more data for popular ones.
- Folksonomies seem to be well-suited for classification tasks in Web information retrieval. Firstly, using tags for classification purposes was very popular among users. Secondly, tags in folksonomies show a higher similarity with classification metadata than other metadata types.

With this chapter, we have come to the end of the first part of the thesis, where we have focused our studies on understanding folksonomies for Web information retrieval. In the second part of this thesis, we will turn our attention to leveraging folksonomies and our knowledge of them for enhancing and improving techniques in the domain of Web information retrieval.

Part II

Leveraging Folksonomies for Information Retrieval

The value of an idea lies in the using of it.

Thomas Edison (1847–1931)

5 Expertise Ranking in Folksonomies

The act of *sharing* information with others is an essential aspect of collaborative tagging and folksonomies, and also one of the defining characteristics of the Social Web in general. On the one hand, tagging allows a user to organize his favorite Web resources. On the other hand, the collaborative dimension also means that he can share with others what he has found interesting on the Web, resulting in a new way for users of discovering useful or otherwise interesting resources. We have seen in previous chapters that tags are helpful for describing what a Web resource is about, and as such they are also helpful in retrieving relevant resources at a later time.

The two basic ways to navigate and discover resources on collaborative tagging systems are browsing by *tag* and by *user*. For example, a Flickr user might browse the list of recently posted photos with the tag `sunset`. He can refine his browsing patterns by adding further tags to his query for building conjunctions of resource sets such as "`sunset` AND `portrait`". Similarly, he might follow the list of recently posted resources of a specific user who is known to be a source of high quality resources on a particular topic. Here, the benefit lies in leveraging other users as human filters for useful resources on the Web.

With the increasing popularity of folksonomies on the Web, however, the numbers of users, tags, and documents within collaborative tagging systems increase as well. The result is that it becomes harder and harder to navigate folksonomies through these simple means [CM08]. Given a list of resources that have been assigned a particular tag, it thus becomes desirable to have a reasonable ranking mechanism that allows for more efficient retrieval of relevant, high quality resources. Similarly, we also need a ranking mechanism for *users* – ideally in a topic-sensitive manner – in order to identify reliable users who we can use as social filters for the kind of information we are interested in. In other words, such a ranking would allow us to find out which users are best to follow in a folksonomy and which are best to avoid (spammers), and thereby help to improve the flow of information within the user community and strengthen social ties.

Developing such ranking schemes for folksonomies is not a trivial task. Firstly, the differences between folksonomies and the Web graph (see Chapter 2.2) complicate the adaptation and extension of traditional ranking schemes in Web information retrieval to folksonomies [MCM+09b]. Secondly, many collaborative tagging systems – particularly those that enjoy high popularity among users – demand only a minimal set of requirements from new users joining the system, and often allow anonymous user ac-

counts where users can act under a pseudonymous username, i.e. the user identity is neither asked for nor verified. Additionally, the popularity of collaborative tagging systems makes them an attractive target for spammers who want to promote their own content on these social platforms (see Section 2.6). For these reasons, we cannot rely on the user himself for self-assessing his "expertise" or "trustworthiness". Lastly, focusing solely on the quantity of user activity is also not recommended [ZD07]. While we have seen that power-law patterns in folksonomies lead to stabilization, a large number of highly active users of collaborative tagging systems are in fact spammers [WZB08]. A ranking approach therefore must favor the "quality" of user activity over its quantity and be able to differentiate between legitimate users and malicious spammers.

We argue that the assignment of the same tag to a resource at a later time than a previous user can be considered as an implicit endorsement or consent to the latter's tagging activity, particularly the judgement that the tagged resource is indeed interesting or useful with regard to the assigned tag. In this chapter, we describe in detail this notion of implicit endorsement, investigate how it can be used for developing a scheme for ranking users, and test our hypothesis regarding user expertise in folksonomies:

> **Hypothesis 2 (User Expertise)**:
> The expertise or trustworthiness of users in a folksonomy can be derived from an analysis of their activity and implicit interactions within the folksonomy.

5.1 Resource Discovery in Folksonomies

The discovery of Web resources – in particular, those of high quality – is one aspect of the broad area of Web information retrieval. In the context of folksonomies, we can differentiate between *external* and *internal* resource discovery. In the former case, a user discovers a resource through means that are external to the folksonomy he participates in, for example through a search engine or a recommendation sent via email from a friend. In the latter case, a user discovers a resource from within the folksonomy. This implies that the resource must have been discovered already by other users in the community. In general, resources must be externally discovered first before they can be subsequently discovered internally.

While there is a wide variety of means for external resource discovery, there are only two basic ways for internal resource discovery in folksonomies as we have described in the beginning of this chapter: browsing by ***tag*** and by ***user***.

Tag-based resource discovery involves obtaining a list of resources that have been assigned a particular tag t. Here, the folksonomy $\mathcal{F}$ is filtered according to a particular tag $t \in \mathcal{T}$ to return a set of resources $\mathcal{R}_t = \{r \in \mathcal{R} \mid (u, \mathbf{t}, r) \in \mathcal{Y}\}$. For example, users can navigate resources by tag via the user interface of collaborative tagging systems (see Figure 3.4 in Section 3.1.1). Most of these systems also provide other means such as news feeds[1] which users can subscribe to for receiving a regularly updated list

[1] There are various technical formats for news feeds such as RSS, ATOM and JSON.

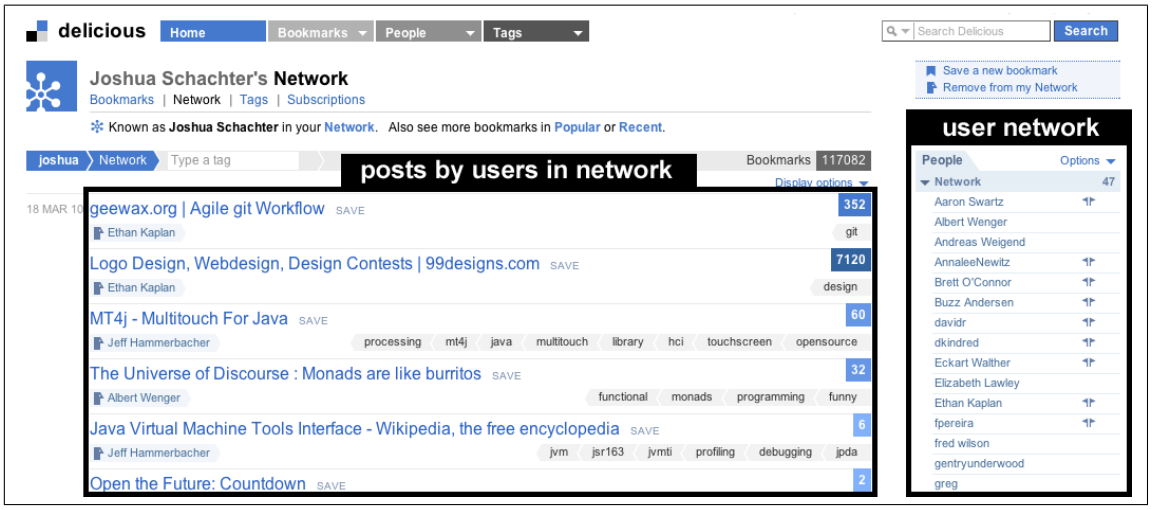


Figure 5.1: **A user's network on Delicious.** The users who are followed by this exemplary user (here: Joshua Schachter, the founder of Delicious) are shown in his *network* on the right side. On the left side, all public posts of the followed users are aggregated into a single list of posts that is sorted descendingly by post creation time, i.e. newest posts are shown first. This allows a user to conveniently track the tagging activities of other people.

of recently posted resources that have been assigned a particular tag.

User-based resource discovery, on the other hand, involves obtaining a list of resources through other users of the folksonomy who have expertise in a particular topic. Here, the folksonomy $\mathcal{F}$ is filtered according to a particular user $u \in \mathcal{U}$ to return a set of resources $\mathcal{R}_u = \{r \in \mathcal{R} \mid (\mathbf{u}, t, r) \in \mathcal{Y}\}$. On Delicious, for example, one can "follow" the tagging activities of other users by adding them to one's personal network of user contacts as illustrated in Figure 5.1. Whenever a user u adds a new resource r through a post p to his personomy $\mathcal{P}_u$, any of his followers is notified about p (and thus r as well) through this network feature. User-based resource discovery is based on the assumption that a user who has tagged high quality resources relevant to a particular topic in the past, he will also tag high quality resources in the future. Thus, if we have identified a user who has expertise in a topic we are interested in, we can simply follow his tagging activities for receiving high quality Web resources that are relevant to the topic.

While both methods have their own advantages depending on the specific context, we believe that user-based resource discovery is more beneficial in general. The main drawback of tag-based resource discovery is that the list of resources $\mathcal{R}_t$ can be very large. Delicious, for example, lists more than three million resources as of March 2010 for the tag `business`, i.e. $|\mathcal{R}_{t=business}| \sim 3,000,000$. Existing collaborative tagging systems usually provide users only two options for sorting and navigating through these resource lists, namely by recency based on post time (newly posted resources are displayed first) or by popularity based on post frequency (frequently posted resources are

displayed first). Neither option guarantees that the resources with the highest quality are presented first, however. User-based resource discovery, on the other hand, leverages expert users as human filters for useful resources on the Web, i.e. users who are good at both internal and external discovery of high quality resources. The main difficulty of user-based resource discovery, however, lies in the finding of such expert users in the first place – the user population $\mathcal{U}$ of Delicious, for example, has more than five million users.

In the following sections, we will therefore discuss the notion of user expertise with regard to a particular topic in the context of folksonomies. We will investigate the characteristics of expert users, i.e. answering the question of what makes a user an expert, and how the level of expertise can be measured.

5.2 Expertise in Folksonomies

Before we can identify experts and rank users according to their expertise, we must first have an idea of the characteristics of an expert. In a general context, an expert is someone with a high level of knowledge or skills in a particular domain. It implies that experts are individuals that we can treat as reliable sources of information and resources that contain such information. This general idea can be readily applied to the context of folksonomies and collaborative tagging. In this section, we describe and justify our two assumptions for experts in folksonomies.

5.2.1 User Expertise and Document Quality

Arguably, the simplest way to assess the *expertise* of a user in a given topic is by the number of documents he has assigned a certain tag (or set of tags) representing the topic. This approach is commonly used by existing collaborative tagging systems. For example, on any page that is dedicated to a particular tag, *LibraryThing* (see Section 2.1.2) presents a list of the Top users of that tag. However, such an approach does not consider the obvious fact that quantity does not imply quality[2]. Similar to the differences in the number and quality between the *submitted* papers and the eventually *accepted* papers of an academic conference, knowing a lot of Web documents about `photography` is not the same as knowing high quality documents about `photography`. Additionally, an approach that relies solely on the frequency of tagging activities is susceptible to spamming (see Section 2.6) because spammers who indiscriminately tag a large number of documents may be mistaken as experts. This vulnerability has been confirmed in both simulated experiments [KEG+07] and empirical studies of folksonomies [WZB08].

Studies in psychology explain that expertise involves the ability to select the most relevant information for achieving a goal [FPE06]. Experts also have the ability to process

[2] Anecdotally, the bookmark count of resources posted in the user's network in Figure 5.1 indeed varies widely: the lowest post count is 2, the highest is 7120. Assuming that a user is most likely to add such users to his network that are good at identifying high quality resources, this observation indicates that lack of quantity does not imply lack of quality.

and apply new information faster than non-experts [Sal91]. In the context of collaborative tagging, users assign tags to resources so as to facilitate retrieval in case the resources are useful to their information needs. A link between studies in psychology and collaborative tagging can thus be drawn. We believe that an expert should be someone who not only has a large collection of documents annotated with a particular tag, but should also be someone who tends to add *high quality* documents to their collections. Similarly, the quality of documents should in turn be determined by the number and by the expertise of the users who have added these documents to their collections. In other words, there is a relationship of mutual reinforcement between the expertise of a user and the quality of a document: a document which is tagged by important users becomes important itself, and vice versa for users.

This approach is similar to the idea of the *HITS (Hypertext Induced Topic Search)* algorithm [Kle98] for the analysis of the hyperlink structure of the Web. HITS is a ranking algorithm that is based on the intuition that hyperlinks can be viewed as topical endorsements: A hyperlink from a Web document d_1 devoted to topic T to another page d_2 is likely to endorse the authority of d_2 *with respect to topic* T. HITS characterizes Web documents with the two attributes of *hubness* and *authority*. A Web document receives higher hub scores if it points to many other documents (i.e. a large number of outgoing hyperlinks), and higher authority scores if many other documents point to it (i.e. a large number of incoming hyperlinks). Here, the relationship of mutual reinforcement is between the hubness and authority of documents, i.e. the hubness of a Web document increases with the authority of documents that it points to, and vice versa.

However, there exists a major difference between HITS and our scenario of user expertise in folksonomies. Collaborative tagging involves two different kinds of interrelated entities, namely human users and Web documents, whereas HITS operates only on Web documents. Additionally, there are only links pointing from users to documents in folksonomies but not vice versa. Thus in our case users will only receive hub scores (expertise) whereas documents will only receive authority scores (quality). This, however, is a very reasonable result: Experts act as hubs because we are likely to find useful resources through them, whereas high quality documents can be considered as authorities because they contain the information we need.

5.2.2 Discoverers and Followers

While it is a very intuitive and reasonable method to use a HITS-like mutual reinforcement approach for the simultaneous measuring of the expertise of users and the quality of documents, we have two concerns about whether it alone is sufficient to yield a good performance in the context of folksonomies[3].

Firstly, in the HITS approach, two users will be considered to have the same level of expertise even though one is the first to tag a set of documents and the other is simply tagging the documents because they are already popular in the community. Because the social aspect of collaborative tagging entices users to actively share information

[3]Similar concerns about simple adaptations of traditional ranking algorithms to the context of folksonomies are raised by Markines et al. [MCM+09b].

with others, they are also more likely to learn from each other, particularly in terms of receiving information from others and spreading it themselves. Mutual reinforcement alone, however, cannot differentiate whether a user is very good at discovering resources or whether he is only following the examples of other members in the folksonomy. Particularly by considering the power laws in folksonomies where a relatively small set of documents is the focus of attention of the user community, we follow the argumentation of Markines et al. [MCM+09b] and argue that a simple adaptation of a traditional Web ranking algorithm such as HITS would therefore not be effective in the specific context of folksonomies. Secondly, a pure mutual reinforcement scheme is very vulnerable to spamming activities. Since the expertise of a user increases with the number of tagged documents, a spammer can exploit this weakness of mutual reinforcement and boost his expertise score by tagging lots of popular documents because these are likely to be of high quality.

Hence, in addition to knowing a lot of high quality documents per se, we believe an expert to be someone who is also able to recognize the usefulness of a document *before* others do [Chi06], thus becoming the first to tag it, and by doing so bringing it to the attention of other users in the folksonomy. This aspect of expertise is similar to a distinguished researcher who not only has profound knowledge of existing publications and prior art in his area of expertise, but who is also able to advance the field by original research of his own. In other words, experts should be the ***discoverers*** of high quality documents, in contrast to the ***followers*** who find these documents at a later time, for example because the documents have already become popular in the folksonomy or because they have been featured in the mass media at some later time. Generally speaking, the earlier a user has tagged a document, the more *credit* he should receive for his tagging activities.

With this assumption, we are introducing *temporal information*, i.e. the time of tagging a document, as an additional dimension for determining the expertise of a user. As such, we infer information from *when* the folksonomy graph changes. While we can never know how a user discovered a document, e.g. through internal or external discovery, the time at which the user posted the document is still a reasonable approximation of how sensitive he is to new information with respect to the topic.

Because the temporal information of tagging activities in a folksonomy cannot be manipulated by its users, there is an added benefit with regard to spam protection. The notion of discoverers and followers with differing credit scores is related to protection mechanisms against Sybil attacks [Dou02, YKGF06] in information security. In a Sybil attack, a malicious user creates multiple user identities in order to boost his reputation or "trust score" within a system such as a peer-to-peer network. However, an attacker can create many identities but only few trust relationships, particularly with participants outside his fake user network. This aspect can be exploited to identify Sybil attacks. Similarly, a spammer that floods a collaborative tagging system for boosting his expertise score will end up being either just a follower (in case he focuses on documents that are already popular within the user community) or a discoverer without any followers (in case he introduces his own spam documents to the community that nobody else cares about). In both cases, he will not benefit much from his malicious

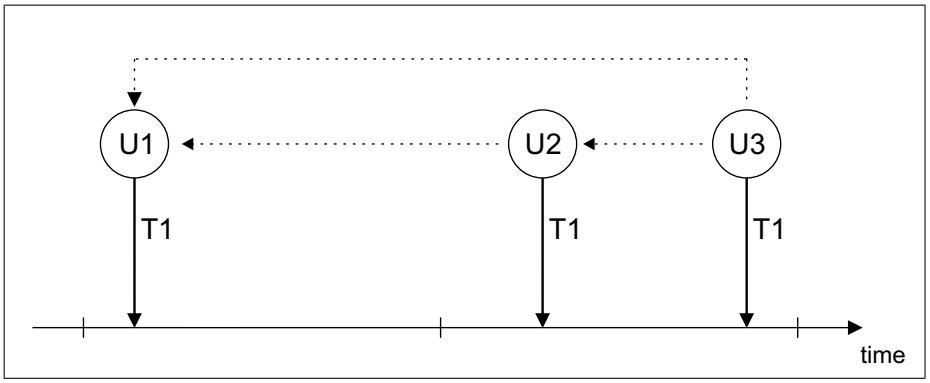


Figure 5.2: **Discoverers and followers: implicit endorsement in folksonomies.** In this example, the tagging timeline of a resource r is shown that has been annotated by three users *U1*, *U2* and *U3* with the same tag *T1*. In such a scenario, the tag assignments of users *U2* and *U3* can be considered as implicit endorsement of the tagging activities of the previous user(s).

activities.

The discoverer-follower scheme can also be considered as an implicit, topical endorsement similar to the intuition behind the HITS algorithm (see discussion above), albeit in an enhanced scenario due to the additional temporal component of the former. As illustration, Figure 5.2 shows the tagging history of a document that has been annotated with the same tag t_i by three users *U1*, *U2* and *U3* at different points in time. When *U2* tags the document with the same tag t_i as *U2* did previously, we argue that this represents an implicit endorsement on two different levels: On the *resource level*, *U2* agrees with *U1* that, firstly, the particular document is a useful resource[4] and, secondly, the document's annotation with tag t_i is reasonable. On the *user level*, *U2* asserts *U1* a certain level of knowledge in the topic that is represented by t_i because *U1* was correct in selecting t_i for describing the document from the viewpoint of *U2*. This notion of implicit endorsement is also transitive, i.e. the tag assignment of *U3* endorses the tagging activities of both *U1* and *U2* with respect to the two levels described above. As more and more users tag the resource in the same way, we are increasingly assured of the tagging activities of the early users in the document's timeline and their expertise with regard to the respective topic. As such, the earlier a user appears in the timeline of a document and the more implicit endorsement he subsequently receives from other users, the more credit he should receive for discovering the document with regard to the topic[5]. On the opposite end, when a user does not receive such implicit endorse-

[4] This notion follows the argumentation and findings of Abrams et al. [ABC98] for the general case of bookmarking a Web document.

[5] This concept of implicit endorsement in the context of folksonomies bears resemblance to the work of Xu et al. [XFMS06]. They propose a measure against tag spam by introducing a reputation score for each user. This score measures how well each user has tagged in the past, which can be modeled as a voting problem: Each time a user votes "correctly", i.e. he tags consistent with the majority of other users, the user receives a higher reputation score.

ment from others, it means that either his selected tag does not describe the document well enough (indicating a lack of expertise of the user with regard to the topic represented by the tag) or that the document is not particularly useful or topically relevant to the rest of the user community.

In summary, we believe that the discoverer-follower assumption is both reasonable and desirable because experts should be the ones who bring good documents to the attention of novices. In addition, this also makes our method of ranking expertise more resistant to spamming activities as mentioned above.

5.3 Spamming-resistant Expertise Analysis and Ranking

Based on the assumptions about user expertise in folksonomies described in the previous sections, we propose *SPEAR (Spamming-resistant Expertise Analysis and Ranking)* as an algorithm to produce a ranking of users with respect to a set of one or more tags. Without loss of generality, we assume that the topic of interest is represented by a tag $t \in \mathcal{T}$. We therefore focus on users who have used tag t for annotations, and documents (resources) to which tag t has been assigned. As we also take into consideration the time at which a tag assignment is created, we extend the notion of tagging by associating a timestamp to each tag assignment (cf. the extended folksonomy definition 2.2-2 in Section 2.2). Hence, every tag assignment $y \in \mathcal{Y}$ becomes a tuple of the form $y = (u, t, d, c)$, where c is the time when user u assigned the tag t to document d, and $c_1 < c_2$ if c_1 refers to an earlier time than c_2. The restriction of $\mathcal{Y}$ to t is then used as the topic-sensitive input data of the algorithm, i.e. $\mathcal{Y}_t = \{y \in \mathcal{Y} \mid t \in (u, t, d, c)\}$.

Since our algorithm is based on the HITS algorithm [Kle98], we first give a brief introduction of this algorithm before describing in detail our proposed SPEAR algorithm.

5.3.1 The HITS Algorithm

The HITS algorithm performs link analysis in order to produce a ranking of Web documents. As we have mentioned above, it is based on the intuition that hyperlinks can be viewed as topical endorsements: A link from a Web document d_1 devoted to topic T to another document d_2 is likely to endorse the authority of d_2 *with respect to topic* T [NZT07]. HITS measures two characteristics of documents, namely authority and hubness. Authoritative documents are those that provide good information with respect to a chosen topic, while hubs are documents that point to good authorities. A detailed comparative analysis of HITS is given in [BRRT05].

According to the assumptions of the algorithm, these two characteristics have a mutual reinforcement relationship: a document has high authority if many documents pointing to it have high hubness, and a document has high hubness if it points to many documents with high authority. Mathematically, the authority $a(d)$ and hubness $h(d)$

of a document d can be defined as follows:

$$a(d) \leftarrow \sum_{d' \in P(d)} h(d') \tag{5.1}$$

$$h(d) \leftarrow \sum_{d' \in G(d)} a(d') \tag{5.2}$$

where $P(d)$ is the set of documents with a link to d, and $G(d)$ is the set of documents pointed to by d.

The above operations can be represented using linear algebra. Let $\vec{a}$ be an n-dimensional vector of authority weights and $\vec{h}$ be another n-dimensional vector of hubness weights for n documents. In addition, let $\mathbf{A}$ be an $n \times n$ square matrix, where

$$\mathbf{A}_{i,j} := \begin{cases} 1, & \text{if document } d_i \text{ links to document } d_j \\ 0, & \text{else} \end{cases} \tag{5.3}$$

Then, the algorithm at the k-th iteration can be represented by the following equations:

$$\vec{a}_k = \alpha_k \mathbf{A}^T \vec{h}_{k-1} \tag{5.4}$$

$$\vec{h}_k = \beta_k \mathbf{A} \vec{a}_{k-1} \tag{5.5}$$

where α_k and β_k are normalization constants.

The authority and hubness vectors can be proved to converge. By solving the above two equations, we have the following equations after k iterations:

$$\vec{a}_k = \theta_k (\mathbf{A}^T \mathbf{A})^{k-1} \mathbf{A}^T \mathbf{1} \tag{5.6}$$

$$\vec{h}_k = \psi_k (\mathbf{A}\mathbf{A}^T)^k \mathbf{1} \tag{5.7}$$

where θ_k and ψ_k are normalization constants. Since $(\mathbf{A}^T\mathbf{A})$ and $(\mathbf{A}\mathbf{A}^T)$ are symmetric, we can obtain for each of the matrices a set of eigenvalues with full eigenspaces. According to theories in linear algebra, $\vec{h}$ would converge to the principle eigenvector (corresponding to the largest eigenvalue) of the matrix $(\mathbf{A}\mathbf{A}^T)$, and a similar case applies to $\vec{a}$. It is found that these two vectors converge quite rapidly in practice.

5.3.2 The SPEAR Algorithm

We now describe our proposed algorithm, SPEAR, for ranking users in a collaborative tagging system by taking into account the two assumptions of experts mentioned in Section 5.2.

Our first assumption of experts involves the level of expertise of the users and the quality of the documents mutually reinforcing each other. We define $\vec{E}$ as a vector of *expertise scores* of users: $\vec{E} = (e_1, e_2, ..., e_M)$, where $M = |\mathcal{U}_t|$ is the number of unique users in $\mathcal{Y}_t$. In addition, we define $\vec{Q}$ as a vector of *quality scores* of documents: $\vec{Q} = (q_1, q_2, ..., q_N)$, where $N = |\mathcal{R}_t|$ is the number of unique documents in $\mathcal{Y}_t$. $\vec{E}$ and $\vec{Q}$ are

initialized by setting every element to 1. Basically, the exact value of the elements can be arbitrary as long as they are all equal, because the vectors will be normalized in later operations.

Mutual reinforcement refers to the idea that the expertise score of a user depends on the quality scores of the documents which he annotated with tag t, and the quality score of a document depends on the expertise score of the users who assign tag t to it. We prepare an adjacency matrix $\mathbf{A}$ of size $M \times N$, where:

$$\mathbf{A}_{i,j} := \begin{cases} 1, & \text{if user } i \text{ has assigned tag } t \text{ to document } j \\ 0, & \text{else} \end{cases} \tag{5.8}$$

Based on this matrix, the calculation of expertise and quality scores is an iterative process similar to that of the HITS algorithm:

$$\vec{E}_k = \alpha_k \mathbf{A}^T \vec{Q}_{k-1} \tag{5.9}$$

$$\vec{Q}_k = \beta_k \mathbf{A} \vec{E}_{k-1} \tag{5.10}$$

To implement the idea of discoverers and followers, we prepare the adjacency matrix $\mathbf{A}$ in a way different from the above method of assigning either 0 or 1 to its components. Before the iterative process we use the following equation to populate the adjacency matrix $\mathbf{A}$:

$$\mathbf{A}_{i,j} = |\{u | (u, t, d_j, c), (u_i, t, d_j, c_i) \in R_t \wedge c_i < c\}| + 1 \tag{5.11}$$

According to equation 5.11, the component $\mathbf{A}_{i,j}$ is equal to 1 plus the number of users who have assigned tag t to document d_j after user u_i. Hence, if u_i is the first to assign t to d_j, $\mathbf{A}_{i,j}$ will be equal to the total number of users who have assigned t to d_j. If u_i is the most recent user to assign t to d_j, $\mathbf{A}_{i,j}$ will be equal to 1. The effect of such an initialization of matrix A is that we have a sorted timeline of any users who tagged a given document d_j.

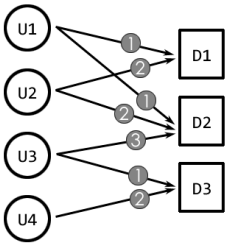
The last step is to assign proper credit scores to users by applying a *credit scoring function* C to $\mathbf{A}$:

$$\mathbf{A}_{i,j} = C(\mathbf{A}_{i,j}) \tag{5.12}$$

A first idea would be a linear credit score assignment such as $C(x) := x$. In this way, when the expertise scores are calculated by the iterative algorithm, users who tagged a document earlier will claim more of its quality score than those who tagged the document at a later time. One concern of such a linear credit score assignment is that the discoverers of a popular document will receive a comparatively higher expertise score even though they might have not contributed any other documents thereafter.

We believe that one criterion of a proper credit scoring function C is that it should be an increasing function with a decreasing first derivative: $C'(x) > 0$ and $C''(x) \leq 0$. In other words, the function should retain the ordering of the scores in $\mathbf{A}$ so that discoverers still score higher than followers but it should reduce the differences between

(a)



(b)

	D1	D2	D3
U1	1.4	1.7	0.0
U2	1.0	1.4	0.0
U3	0.0	1.0	1.4
U4	0.0	0.0	1.0

(c)

	Rank	Score
U1	1	0.422
U2	2	0.328
U3	3	0.212
U4	4	0.038

Table 5.1: **A simple example of using SPEAR to rank users in a folksonomy.** (a) shows the bipartite graph of four users and three documents. An arrow from a user to a document represents the fact that the user has assigned the tag concerned to the document. The numbers in circles represent the order of assigning the tag to the document. (b) shows the adjacency matrix after the credit score function is applied. Finally, (c) shows the final ranking of the users. In this example, `U1` is the discoverer of two popular documents (`D1` and `D2`), therefore `U1` is ranked first. `U4` is a mere follower of a single document (`D3`), and so `U4` is ranked last.

scores which are too high. This is because it is undesirable to give high expertise scores to users who happened to be the first few to tag a very popular document but have not contributed any other high quality documents thereafter. Here, we conduct our experiments with $C(x) := x^{0.5} = \sqrt{x}$. Overall, the above procedures of generating an adjacency matrix for the operation of SPEAR from the tagging data given a certain credit score function can be represented by the following function:

$$\mathbf{A} = GenerateAdjacencyMatrix(R_t, C) \tag{5.13}$$

The final SPEAR algorithm is shown in pseudocode in Algorithm 1, while Table 5.1 presents an example of running SPEAR on a simple case.

The SPEAR algorithm is different from the HITS algorithm in two aspects. Firstly, the adjacency matrix is not a square matrix. This is because, instead of considering a single set of documents, we consider a set of users and a set of documents, and the number of users is not necessarily equal to the number of documents under consideration. Secondly, instead of having only the values 0 or 1 for the components in the adjacency matrix $\mathbf{A}$, we initialize the matrix with different values depending on when the documents were tagged by the users. However, SPEAR can be proved to converge in the same way as HITS. This is because the proof involves the eigenvectors of the matrices $(\mathbf{A}^T\mathbf{A})$ and $(\mathbf{A}\mathbf{A}^T)$, instead of $\mathbf{A}$ [FLM+06]. Also, the proof is independent of the values in the components of $\mathbf{A}$, as long as $\mathbf{A}$ is non-negative, which is also true in the case of SPEAR. Hence, SPEAR is guaranteed to converge[6] under the same conditions as HITS.

[6] In our experiments, the values in the vectors stabilized on average after 160 iterations.

Algorithm 1 SPEAR: Spamming-resistant Expertise Analysis and Ranking

Input: Number of Users M
Input: Number of Documents N
Input: A set of tag assignments $\mathcal{Y}_t = \{(u, t, d, c)\}$
Input: Credit scoring function C
Input: Number of iterations k
Output: A ranked list L of users.

Set $\vec{E}$ to be the vector $(1, 1, ..., 1) \in \mathbb{Q}^M$
Set $\vec{Q}$ to be the vector $(1, 1, ..., 1) \in \mathbb{Q}^N$
$A \leftarrow GenerateAdjacencyMatrix(R_t, C)$
for $i = 1$ to k **do**
    $\vec{E} \leftarrow \vec{Q} \times \mathbf{A}^T$
    $\vec{Q} \leftarrow \vec{E} \times \mathbf{A}$
    Normalize $\vec{E}$
    Normalize $\vec{Q}$
end for
$L \leftarrow$ Sort users by their expertise score in $\vec{E}$
return L

5.4 Experimental Setup

5.4.1 Methodology

It is not a trivial task to investigate and evaluate the performance of SPEAR. The reason is the lack of a proper ground truth for user expertise in folksonomies to compare experimental results with. Firstly, there exists no standard data set for the evaluation user ranking on Delicious at the time of writing. Secondly, a manual examination of user accounts would only be possible for a limited volume of experimental data, i.e. small sets of users and documents, and thus might result in an evaluation that is not representative or objective. For this reason, we adopt the experimental setup of studies such as [CLW08] and [KEG+07]: We combine real-world data and simulated data to evaluate and compare the behavior and performance of SPEAR with baseline algorithms. Additionally, we augment these experiments with some qualitative studies to verify our results.

Our methodology can be described as follows. Firstly, real-world data is used as the base input for our experiments. Here, it is important to realize that with regard to user data, "real users" means "user accounts derived from real-world data", which may include real human users as well as real automated spam bots and other phenomena found in the wild. We then insert controlled, simulated data into the original real-world data at the proper places. The behavior of simulated users – and thus the places where their activities are inserted to – is determined by a) the results of recent studies of collaborative tagging systems such as [KEG+07, KFG+07, HKGM07, WZB08, KSHS08] and b) the characteristics of our real-world data sets. The former ensures that simulated users

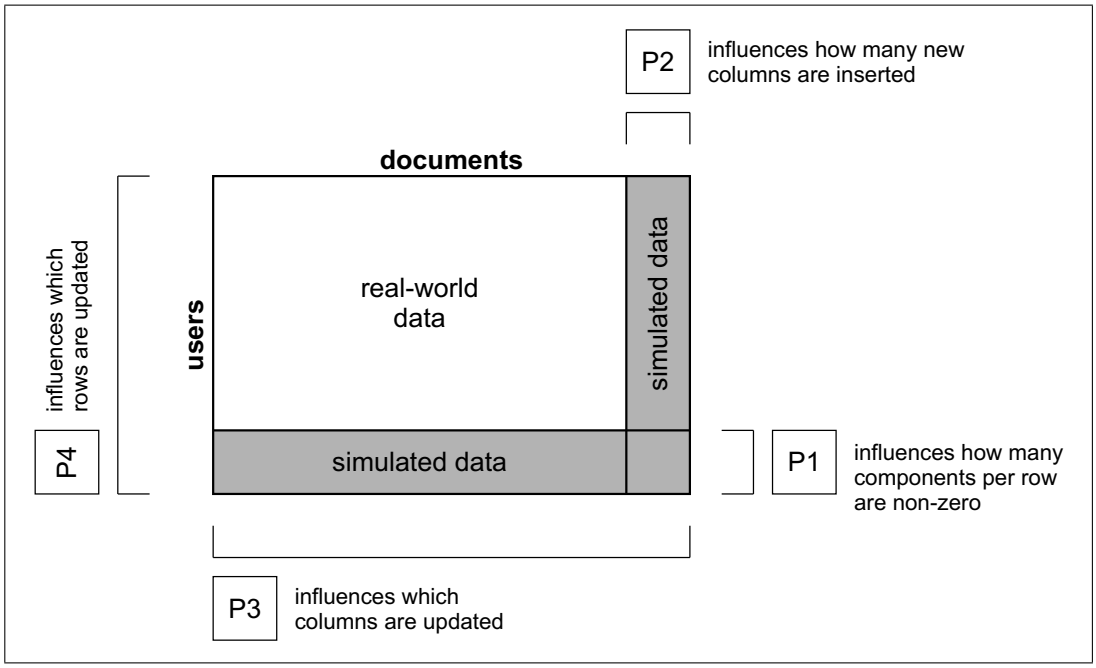


Figure 5.3: **Effects of simulation parameters P1-P4.** The simulation parameters control where and how the adjacency matrix **A**, which is initialized with real-world data, is updated and augmented through simulated data. In **A**, users and documents are represented as rows and columns, respectively.

show the expected *type* of behavior, whereas the latter ensures that we can properly setup the *intensity* or *magnitude* of their behavior according to our real-world data sets. This approach of combing real-world and simulated data allows us to mitigate the lack of a proper ground truth by embedding controlled data into a real-world scenario, and analyze how the expected results compare to the experimental outcomes.

With regard to real-world data, we created and used the large *SPEAR collection* of folksonomy data described in Section 3.2.3. To recall, this collection consists of 110 tag-based data sets that comprise a wide variety of topics derived from 110 seed tags such as `film`, `geography`, `history` and `opera`. These data sets contain a total of more than 1 million users, 15 million bookmarks, 50 million tag assignments and 130,000 Web documents. As such, the collection allows us to study how consistent the performance of SPEAR is across different documents and users in a folksonomy.

With regard to the simulation, it is performed by manipulating and updating the adjacency matrix **A**, which is initially set up from real-world data. The simulation is controlled with four parameters that are described in Section 5.4.4, and allows for a probabilistic setup to make the experiments more realistic. These parameters control where and how the adjacency matrix **A** is updated and augmented through simulated data. The big picture is shown in Figure 5.3. The insertion of simulated data is mainly carried out by injecting virtual bookmarks into the real-world data at the proper places.

User Type	Variants
Experts	Geek Veteran Newcomer
Spammers	Flooder Promoter Trojan

Table 5.2: **The simulated user profiles created for the evaluation of SPEAR.**

For example, a discoverer-type user would be simulated by inserting a virtual bookmark early in the timeline of document's "real" bookmarking history, i.e. updating the respective column $\mathbf{A}_{*j}$. All users with a later bookmark would automatically become followers of the simulated user for this document. Similarly, we would have to insert virtual bookmarks to popular documents in order to simulate experts because these users tend to tag only relevant information.

In our experiments, we construct two different types of simulated user profiles: *expert*-like users and *spammer*-like users. For each type of these users, we model three variants in order to better match real-world scenarios and to improve the evaluation setup. An overview of simulated user profiles is shown in Table 5.2. They are described in detail in Sections 5.4.2 and 5.4.3. It should be noted that due to the probabilistic setup of our experiments even identical user profiles would produce variations in simulated data (see Section 5.4.4 below). On the one hand, this means that even two users with the same profile would behave differently up to a certain extent (there can be some geeks who are "better" geeks than the others). On the other hand, we can expect overlaps in user behavior and experimental results between different user variants (a "good" newcomer might receive a higher expertise score than a "bad" veteran).

5.4.2 Simulated Experts

Simulated expert profiles are subdivided into geeks, veterans, and newcomers. They represent expert users with different levels of expertise with regard to a particular topic.

A *veteran* is a user who bookmarks significantly more documents than the average user, following the reports of user behavior on Delicious described in [HKGM08, NM07a]. He tends to be among the first users to tag documents which usually become quite popular within the community. Hence, he is a discoverer with many followers. In the real-world, a veteran could be compared to an experienced researcher who has profound knowledge of his area of expertise, and advances the field by publications of his own.

A *newcomer* is an upcoming expert who is only sometimes among the first to "discover" a document. Most of the time, the documents are already quite well-known within the community at the time he tags them. In the real-world, a newcomer could be compared to a PhD student who already has knowledge about the state of the art in his area of expertise, but has yet to gain his reputation within the scientific community.

He has just started with his own original research, so the number of publications is still low.

A *geek* is similar to a veteran but has significantly more bookmarks than a veteran. In the real-world, he could be a very distinguished researcher with the best knowledge of his area of expertise and a significant number of own publications. We can consider the geek profile as the "best" expert within our simulation.

In the experiments, geeks should generally be ranked higher than veterans, and the latter should in turn rank higher than newcomers. It should be noted that the differences – or the "gap" – between geeks and veterans are more subtle compared to those between veterans and newcomers. While geeks generally have higher chances to tag high quality documents, the probabilistic setup of our experiments plus the notion of document quality make it possible that some veterans may achieve a higher expertise score than some geeks.

5.4.3 Simulated Spammers

Simulated spammer profiles are subdivided into flooders, promoters, and trojans. They represent different types of spammers found in the wild. While these spammers employ different strategies to achieve their goals, their common objective is to artificially boost their reputation in the folksonomy and promote their own documents, which are not likely to be of interest to legitimate users.

A *flooder* tags a huge number of documents which already exist in the system, most likely in an automated way. This spammer variant can often be found in the wild [WZB08, KEG+07, KFG+07]. He tends to be one of the last users in the bookmarking timeline[7]. Additionally, he tends to tag documents already known to the community rather than tagging new documents because he aims at gaining "reputation" through lots of bookmarks of existing, popular content.

A *promoter* is a spammer who focuses on tagging his own documents to promote their popularity, and does not care much for other documents. He tends to be the first to bookmark documents which attract few followers if any. This spammer type is quite common and we could find several on Delicious during our experiments. There were cooperating groups of them who had sequentially named user accounts of the form *iSpamYou001*, *iSpamYou002*, etc. who were possibly trying to perform a Sybil-type attack as discussed in Section 5.2.2. Such promoter-type spammers have recently been reported: Wetzker et al. [WZB08] found that 19 of the top 20 most active Delicious users in their experimental data set were spammers who bookmarked ten thousands of URLs pointing to only few Web domains. In total, these 19 spammers alone accounted for 1.3 million bookmarks or around 1% of their data corpus. Likewise, Krause et al. [KSHS08] observed spammers registering several accounts and publishing the same

[7] This spammer behavior is not only caused by specific spamming strategies that try to boost expertise/reputation scores by spamming popular documents. In practice, such behavior can also be the result of the spam bot being created by its masters long after the Delicious service went online in 2003, so regular users have had a head start. Back in 2003, the eventual success of Delicious was not foreseeable, meaning that spamming it right away was not worth the risk and effort.

bookmark several times in a coordinated "attack". Similar to our anecdotal findings, [KSHS08] also observed that the number of digits in a username is an indication of "spamminess", i.e. the more digits, the more likely the user is a spammer.

A *trojan* is a more sophisticated spammer in the way that his strategy is to mimic regular users in the majority of his tagging activities, thus sharing some traits with so-called *shilling attacks* in recommender systems [SZC05, CNZ05] and *slow-poisoning attacks* [YSR+06, HPS08]. A trojan disguises his malicious intents by tagging already popular pages, but at some point he adds links to his own documents which can be malware-infected or phishing Web pages. In other words, this spammer follows the "majority" opinion in the folksonomy most of the time to avoid detection. He tries to trick users into believing he is a knowledgeable, benevolent member of the community and then lures them into a trap – like a wolf in sheep's clothing. A recent study by [MC08] discusses trojan-like spammers in the context of collaborative systems for reporting phishing Web sites.

As flooders and promoters can already be observed in existing collaborative tagging systems, an algorithm for telling experts from spammers should therefore be able to handle such spammer types. Trojan-type spammers could be seen as the next step in the evolution of malicious spamming techniques. For this reason, we are interested in finding out how well SPEAR performs on these sneaky and potentially more harmful spammers.

5.4.4 Simulation Parameters

We control our simulation with four parameters that we use to model the simulated users and their tagging behavior.

- **P1**: *Number of a user's bookmarks*. For example, geeks and flooders would have a greater number of bookmarks than veterans or promoters, respectively.
- **P2**: *Newness* – Percentage of bookmarks of such documents that are not in the original real-world data. To make our experiments more realistic, we needed a feature which allowed simulated users to bookmark completely new documents, i.e. documents that hadn't been bookmarked by any real-world user yet. For example, trojans and promoters create links to their own Web documents. The actual URLs of such "new" documents were irrelevant in our experiments as long as they were unique.
- **P3**: *Document rank preferences* – A probability mass function (PMF) which specifies whether rather popular or rather unpopular documents tend to be selected when inserting simulated bookmarks. For example, the PMFs of veterans and trojans tend to select popular documents whereas the PMFs of flooders are more evenly distributed.
- **P4**: *Time preferences* – A probability mass function (PMF) which specifies where in the original timeline a simulated bookmark tends to be inserted into a given

Type	P1	P2	P3	P4
Geek	$2 * P1_{Veteran}$	0.10	*See figure 5.4(a)*	*See figure 5.4(b)*
Veteran	$\{0.01, 0.02, ..., 0.05\} * n_d$	0.10	*See figure 5.4(a)*	*See figure 5.4(b)*
Newcomer	$P1_{Veteran}$	0.10	*See figure 5.4(a)*	EQUAL()
Flooder	$\{0.02, 0.04, ..., 0.20\} * n_d$	0.05	EQUAL()	*See figure 5.4(b)*
Promoter	$\{10, 20, ..., 100\}$	0.95	EQUAL()	*See figure 5.4(b)*
Trojan	$\{10, 20, ..., 100\}$	0.10	*See figure 5.4(a)*	*See figure 5.4(b)*

Table 5.3: **Configuration of parameters P1-P4 for simulated user profiles.** n_d is the total number of bookmarked documents in the relevant data set. $EQUAL()$ means that each document rank or time is selected with equal probability. The sequence of numbers in curly brackets denote multiple experiment runs with varying parameters as indicated.

document's bookmarking history. For example, the PMFs of veterans tend to focus on the early stages of the bookmarking history, newcomers are rather evenly distributed, and flooders tend to be very late.

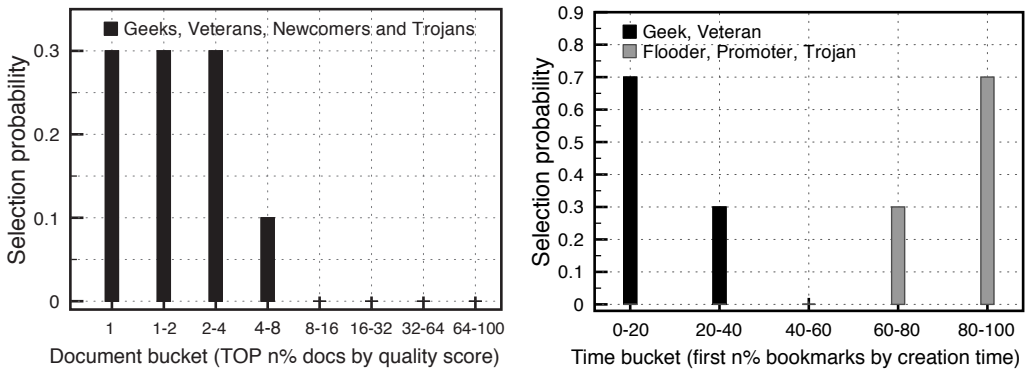
The actual configurations of the simulation parameters for each user type are shown in Table 5.3 (see also Figure 5.4 for the probability mass functions for **P3** and **P4**). Note that the number of bookmarks for promoters and trojans is set to absolute values (from 10 to 100), unlike that for flooders. Our reason for this decision is that promoters and trojans should exhibit behavior similar to that of real users (flooders are more likely to be bots that generate bookmarks automatically). The mean number of bookmarks of real users in our data set was $\mu_{max} = 69$, therefore our chosen values covered a similar range.

5.4.5 Evaluation Baselines

In order to have a baseline for our evaluation, we compare SPEAR with two related algorithms. The first algorithm, *FREQ*, is a simple frequency count ranking algorithm, and as such it relies solely on a quantitative analysis of tagging activities for ranking users. We use it for two main reasons: On the one hand, it represents the naive, intuitive approach to rank users in folksonomies – the more a user engages in tagging activities, the higher his expertise. On the other hand, this approach is commonly used by existing collaborative tagging systems. Hence, we consider the FREQ algorithm as a general baseline for our experiments[8], and also as the "empirical" baseline with regard to folksonomies in practice.

The second algorithm is the original HITS algorithm, albeit slightly adapted to operate on the data model of folksonomies. We choose HITS because it employs a mutual

[8] Another popular baseline variant in such experiments is the random algorithm, which would simply rank users at random. However, we argue that a comparison of SPEAR with the random algorithm would not yield a lot of insights into its performance.



(a) PMF for document rank preferences (P3)

(b) PMF for time preferences (P4)

Figure 5.4: **PMF for document rank preferences (P3) and time preferences (P4).**
(a) PMF of document rank preferences (P3) for geeks, veterans, newcomers and trojans. Flooders and promoters choose document ranks randomly. Lower bucket numbers refer to higher quality documents. We chose exponentially increasing bucket sizes here to account for power law patterns in folksonomies.
(b) PMF of time preferences (P4) for geeks, veterans (black) and flooders, promoters, trojans (gray). Lower bucket numbers refer to earlier timestamps, e.g. the first bucket represents the first 20% of bookmarks in a URL's history. Newcomers choose timestamps randomly.

reinforcement scheme as SPEAR does, although it lacks the latter's notion of discoverers and followers. More precisely, it differs from SPEAR in the initialization of the adjacency matrix: HITS is a special case of SPEAR when the credit scoring function $C(x)$ is constant, namely $C(x) = 1$. Another reason for using HITS as the second baseline of our experiments is that mutual reinforcement is a characteristic of several other ranking approaches in Web information retrieval. Notable examples are the PageRank algorithm [BP98] and its folksonomy-adaptations *FolkRank* [HJSS06c, HJSS06d], *ExpertRank* [JS06] and *SocialPageRank* [BXW+07]. Similarly, a direct adaptation of HITS to folksonomies, *SocialHITS*[9], is proposed by Abel [ABB+09]. Hence, using the HITS algorithm as a baseline also allows us to compare SPEAR with this class of ranking algorithms whose essential element is a mutual reinforcement scheme.

A conceptual comparison of the three algorithms SPEAR, HITS and FREQ is shown in Table 5.4. In particular, we can see that SPEAR has a greater level of detail compared to the baseline algorithms: While FREQ is a simple global aggregation ("How *much*

[9]The SocialHITS algorithm was published several weeks after we proposed SPEAR in [ANG+09]. Apart from the lack of a folksonomy-specific component such as the discoverer-follower scheme we use in SPEAR, the evaluation of SocialHITS described in [ABB+09] was conducted on a significantly smaller experimental data set. Abel's data set contained only $|\mathcal{U}| = 450$ users ("mainly from the research community in computer science") who bookmarked a total of $|\mathcal{R}| = 2,189$ Web documents and provided $|\mathcal{Y}| = 3,190$ tag assignments.

does a user engage in tagging activities?") and HITS analyzes the folksonomy graph only down to the resource level ("On *which resources* does a user focus his tagging activities?"), SPEAR also analyzes the tagging activities of users for individual resources ("How does a user *compare to others* who tag the same resources?"). This improved granularity should help SPEAR with differentiating the expertise of users, most notably preventing SPEAR from assigning multiple users the same expertise score (which we particularly expect to happen for the simple frequency count ranking of FREQ).

	FREQ	**HITS**	**SPEAR**
Quantity analysis	yes	yes	yes
Quality analysis	no	yes*	yes*
Level of detail	**global**	global ↓ **resource**	global ↓ resource ↓ **user activity per resource**

**via mutual reinforcement*

Table 5.4: **Conceptual comparison of FREQ, HITS and SPEAR.**

5.5 Experimental Results

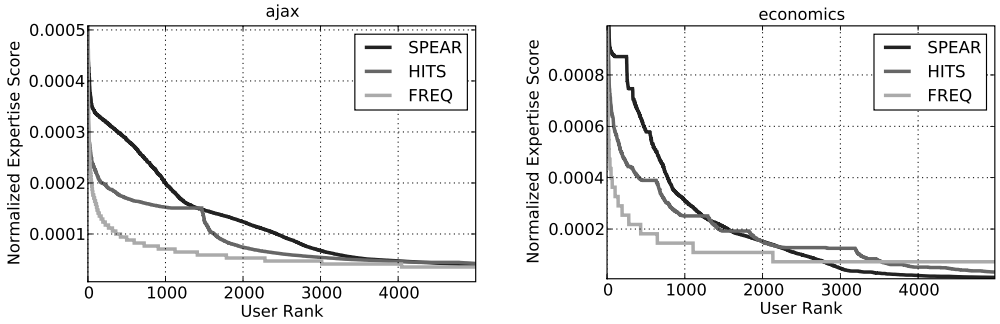
We start our discussion of the experimental results with a description and comparison of the general behavior of the three algorithms SPEAR, HITS and FREQ. We continue with a detailed description of how different types of expert users and spammers are ranked by these algorithms, including both quantitative and qualitative analyses.

5.5.1 General Behavior

Figure 5.5 shows the normalized expertise score distributions of SPEAR, HITS and FREQ for two exemplary data sets, namely `ajax` and `economics`. We observed that SPEAR generally produced more differentiated values than HITS and FREQ for top users, i.e. the difference in expertise scores between two ranks for SPEAR was generally larger than for HITS and FREQ, where the curves were flatter. We will see how SPEAR benefits from this characteristic in Section 5.5.3.

Another observation was the staircase-like shape of FREQ caused by the integer frequency counts on which it is based. This means FREQ tends to group users into buckets of equal expertise score instead of assigning an individual rank to each user, i.e. many users share the same rank. While SPEAR and HITS also show occasional staircase steps, they result from different reasons.

In HITS, users who have assigned the same tag to the same set of documents will be assigned the same rank. This is because their expertise score is derived from the quality



(a) Results for `ajax`

(b) Results for `economics`

Figure 5.5: Normalized expertise scores of the Top 5000 users as returned by SPEAR, HITS and FREQ for the two exemplary data sets `ajax` and `economics`.

of the documents they tag, and if they happen to tag the same documents they will also receive the same expertise score. In other words, this behavior is caused by the level of detail (see Table 5.4) of the algorithm: Since HITS can only analyze folksonomy data down to the resource level, it cannot differentiate users who tag the same document. Our experiments show that this lack of detail is actually relevant in practice because we can observe many such cases in our data.

In SPEAR, the observed staircase steps are the result of a limitation of our real-world data sets. We can only retrieve the date of a social bookmark from Delicious but not its time of day (cf. Section 3.1.1), resulting in what we call "time collisions" of users how tag the same document at the same day. Even though SPEAR in theory would be able to differentiate these users properly, the limited granularity of our data sets with regard to the timelines of documents causes SPEAR to assign the same expertise score to users affected by such time collisions. For these reasons, it is not a deficiency of our proposed algorithm, and the problem can be readily solved with more fine-grained data sets.

In summary, SPEAR was able to differentiate the expertise scores of users better than the baseline algorithms HITS and FREQ. It was better at spreading the expertise scores across a wider range, and was less likely to assign the same score to two users. We argue that this is the result of the improved level of details of SPEAR's analysis of folksonomy data compared to the baselines.

5.5.2 Promoting Experts

To study how different variants of experts are ranked by SPEAR, we simulated, for each of the 110 real-world data sets, 20 experts of each type (60 total per data set) and added them together with their simulated tagging activities to the corresponding data set. We then applied SPEAR, the original HITS algorithm and FREQ to these data sets comprising both real-world and simulated users. The results are shown in Figure 5.6. For a better comparison across data sets, we normalize the rank of simulated users so

that a user who is assigned the highest expertise score (with integer rank 0 being the highest absolute rank, i.e. prior to normalization) will receive a normalized rank of 1.0. The normalized rank of 0.0 is assigned to users with the lowest expertise score. It is expected to observe some overlapping between the three expert variants due to the PMF-based simulation setup as described in Section 5.4.1.

The plots show some major differences between SPEAR and the baseline algorithms. In SPEAR, geeks were generally ranked higher than veterans, which in turn were ranked higher than newcomers. We also observed that geeks and experts did compete for the top ranks even though geeks won in general. This means that some veterans, although having had fewer tagged documents than geeks in general, were ranked higher by SPEAR because they had some documents of higher quality in their personomies $\mathcal{P}_u$. Another observation was that veterans were ranked higher than newcomers. Similarly, we could see again that some newcomers were assigned higher ranks than some veterans due to the reasons mentioned above.

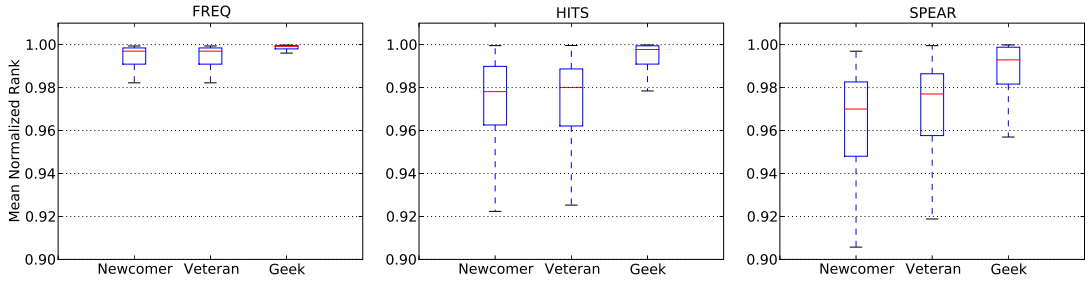
On the other hand, the baseline algorithms HITS and FREQ performed not as good as SPEAR. They did rank geeks higher than veterans and newcomers, but geeks were also the "easiest" expert variant to be ranked correctly because they have a very large number of high quality documents in their personomies. This means even the naive FREQ algorithm should and did perform reasonably for this user variant. However, both HITS and FREQ failed to differentiate between veterans and newcomers, which ended up being mixed with each other. This result suggests that only SPEAR succeeded in distinguishing veterans and newcomers by implementing the notion of discoverers and followers. In contrast, HITS still tended to return results which were heavily influenced and biased by the number of documents in a user's collection, even though it is also an implementation of a mutual reinforcement scheme. A zoomed view of the rankings produced by the three algorithms is shown in Figure 5.7 for two selected tags `economics` and `iphone`. We can see that the three expert variants were clearly separated by SPEAR, whereas the baseline algorithms intermixed veterans with newcomers.

In summary, we can conclude that in usage scenarios where quantity does not guarantee quality — and we believe collaborative tagging is one such scenario — SPEAR is expected to produce better rankings of users. We argue that this is the result of SPEAR being better at detecting the subtle differences between different types of users.

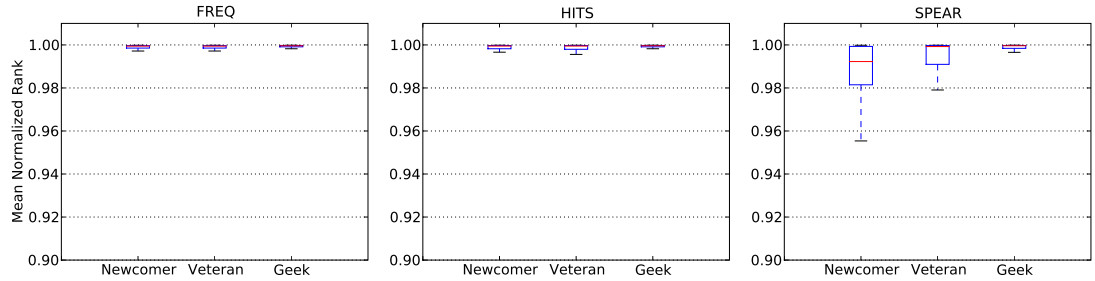
5.5.3 Demoting Spammers

We conducted the spammer-related experiments similarly to our expert-related experiments described in the previous section. Here, we generated and added 20 flooders, promoters and trojans, respectively, for each of the real-world data sets. Additionally, we also varied the number of documents tagged by each spammer type for evaluating the algorithms with regard to their sensitivity to the quantity of users' tagging activities. Again, we normalized user ranks as described in the previous section for the presentation of experimental results, which are shown in Figure 5.8.

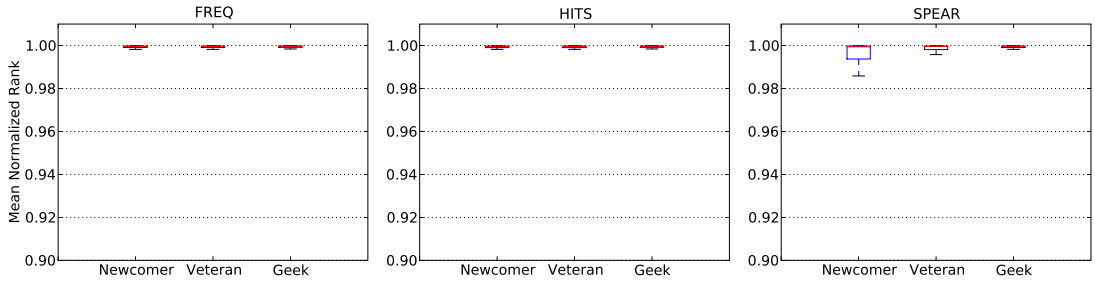
FREQ showed the weakest performance among the three algorithms. All spammers



(a) Results for $P1_{Veteran} = 0.01 * n_d$



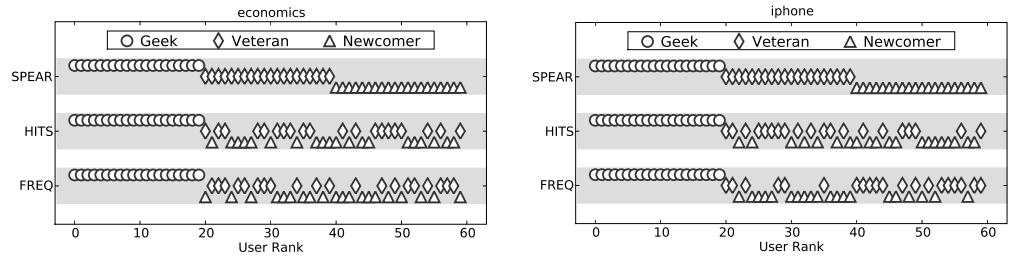
(b) Results for $P1_{Veteran} = 0.03 * n_d$



(c) Results for $P1_{Veteran} = 0.05 * n_d$

Figure 5.6: Boxplots of mean normalized ranks of simulated experts—**newcomers, veterans, geeks**—in direct comparison across all data sets for the three algorithms. Rank values of 1.0 and 0.0 represent the top-ranked user (highest expertise) and the bottom-ranked user (lowest expertise), respectively. The plots (a), (b) and (c) show the results for $P1_{Veteran} = 0.01 * n_d$, $P1_{Veteran} = 0.03 * n_d$ and $P1_{Veteran} = 0.05 * n_d$, respectively. The $P1$ parameters of geeks and newcomers change in relation to $P1_{Veteran}$ as shown in Table 5.3.
With regard to the results, some overlapping of simulated experts is expected due to the experimental setup as described in the text.

were assigned top ranks simply because they tagged large numbers of documents. This result shows that a simple frequency count ranking algorithm is very vulnerable to



(a) Simulated experts in `economics`

(b) Simulated experts in `iphone`

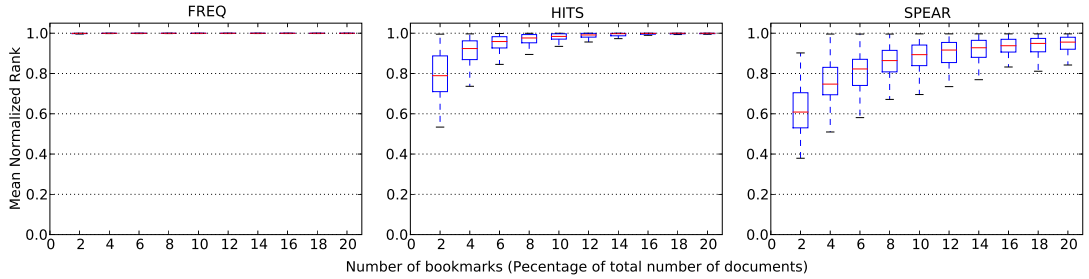
Figure 5.7: **Ranks of simulated experts for two selected tags `economics` and `iphone`.** In this figure, the absolute rank value of 0 represents the top-ranked user (highest expertise), and larger absolute rank values denote lower expertise. SPEAR clearly distinguishes between the three types of expert users, while HITS and FREQ tend to mix up veterans and newcomers.

spamming activities in folksonomies. This is true particularly for flooder-type spammers, which unfortunately are often found in today's collaborative tagging systems [WZB08]. HITS, on the other hand, performed better than FREQ but was dominated in all experiments by SPEAR. While HITS was good at demoting promoters, it had problems to demote flooders with increasing numbers of spam bookmarks, and was weak in general for handling trojans.

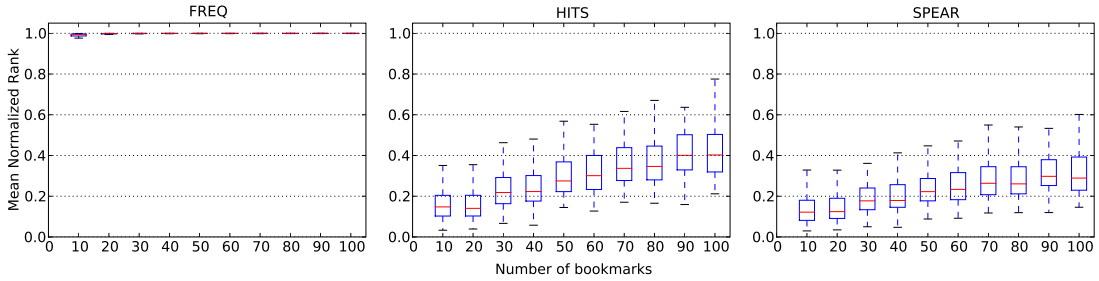
SPEAR showed the best performance among the three algorithms. Firstly, it correctly demoted both flooders and promoters by assigning them significantly lower ranks than HITS and FREQ. This result is very encouraging because flooders in particular have a strong negative impact on collaborative tagging systems by inserting large volumes of junk data into the technical infrastructure, and because they also pollute the folksonomy itself.

Secondly, SPEAR was also able to demote trojans, which use a much more sophisticated spamming scheme than flooders and promoters. While trojans were still ranked higher than the other two spammer variants, they were rarely ranked higher than rank #100 by SPEAR across our experimental runs. This positive result is particularly illustrated in Figure 5.9, which shows the detailed result for the two selected tags `economics` and `iphone`. Compared to HITS and FREQ, SPEAR demoted all trojans from the TOP 200 ranks. Given that in practice the TOP 10 to the TOP 50 experts should be the ones we are most interested in, SPEAR in its current form already performed reasonably well in getting rid of all trojans in the relevant rank range.

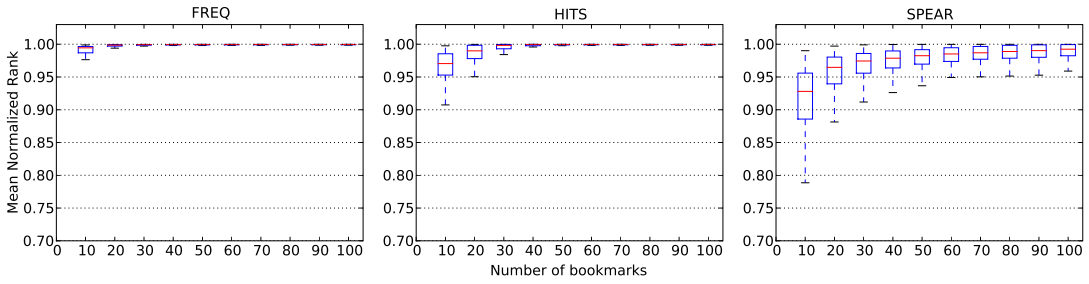
That being said, the problem with trojans is that it is tricky to demote them without demoting good users at the same time, because from a pragmatic point of view a trojan is still a rather good hub of resources. One possible way to tackle trojans would be to verify whether documents are really legitimate and useful resources prior to visiting them, and SPEAR can actually support users in this assessment by computing a quality score of documents.



(a) Results for flooders



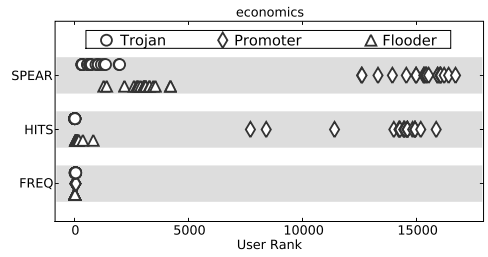
(b) Results for promoters



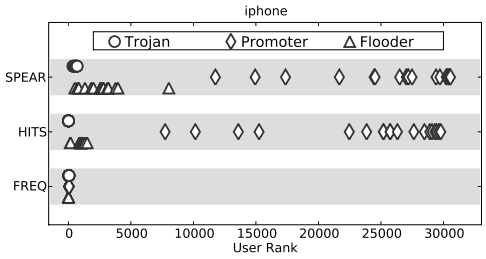
(c) Results for trojans

Figure 5.8: Boxplots of mean normalized ranks of simulated spammers – **flooders**, **promoters**, **trojans** – across all data sets for the three algorithms in relation to the number of bookmarks generated per flooder (x-axis). Rank values of 1.0 and 0.0 represent the top-ranked user (highest expertise) and the bottom-ranked user (lowest expertise), respectively. Lower values are better.

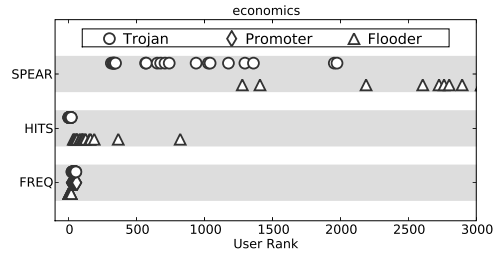
Lastly, we observed that SPEAR was the only algorithm that did not tend to "clump" spammers together in one spot in our experiments, i.e. it was better at differentiating and detecting nuances in spammer behavior compared to HITS and FREQ. We argue that this is a direct result of the different expertise score curves as described in Section 5.5.1.



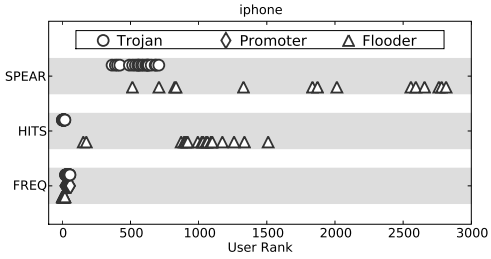
(a) Simulated spammers in `economics`



(b) Simulated spammers in `iphone`



(c) Simulated spammers in `economics` (zoomed)



(d) Simulated spammers in `iphone` (zoomed)

Figure 5.9: **Ranks of simulated spammers for two selected tags `economics` and `iphone`.** In this figure, the absolute rank value of 0 represents the top-ranked user (highest expertise), and larger absolute rank values denote lower expertise. (a) and (b) show the full range of user ranks, while (c) and (d) focus on the Top 2500 user ranks. The figures show that SPEAR could demote trojans from the Top 200 ranks.

5.5.4 Simultaneous Ranking of Experts and Spammers

In the above experiments, we injected each type of simulated users separately into the real-world data sets. As an overall evaluation, we conducted a combined evaluation of expert users and spammers by injecting both types – and thus all six variants – simultaneously into the real-world data sets to compare the performance of different algorithms.

Similar to the experiments described above, we first generated the six different variants of simulated users using different parameters, and injected their profiles and tagging activities into the real-world data sets. We then used the three algorithms SPEAR, HITS and FREQ to rank the users. Due to the large number of possible combinations of parameters, we only report a typical result in detail: Figure 5.10 shows the results of this experiment with the settings $P1_{Veteran} = P1_{Flooder} = 0.03 * n_d$ and $P1_{Promoter} = P1_{Trojan} = 100$. With these parameters, the spammers always had larger numbers of bookmarks than newcomers and veterans, but were comparable to those of the geeks. Table 5.5 shows the mean normalized rank of each of the different types of users pro-

duced by the three algorithms.

From Figure 5.10 and Table 5.5, we can see that a combined simulation produces similar results as the separated simulations described in the previous sections. FREQ ranked all spammers at the top due to their large collection of bookmarks. HITS was able to demote the flooders and promoters to a certain extent, but still ranked the trojans among the Top users. SPEAR showed good performance by demoting the flooders and promoters more significantly than FREQ and HITS, and by removing the trojans from the top of the list. It was also the only algorithm which could rank all three expert types at the top *and* retain the expected correct order, i.e. geeks before veterans before newcomers.

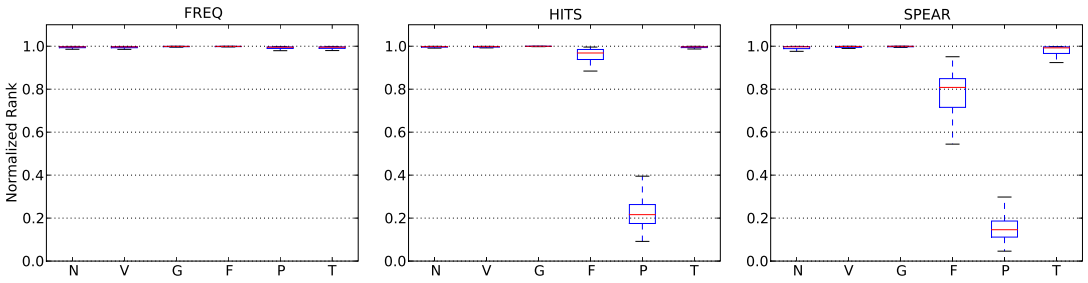


Figure 5.10: **Boxplots of mean normalized ranks of all different types of simulated users being ranked at the same time.** In the figures, N=Newcomers, V=Veterans, G=Geeks, F=Flooders, P=Promoters, and T=Trojans. Due to the large number of users in the data sets and the zoom level of the plots, it is difficult to judge the detailed results of expert users in this figure.

	G	V	N	F	P	T	Ranking Order
SPEAR	0.9914	0.9821	0.9774	**0.7687**	**0.1656**	**0.9707**	$G > V > N > T > F > P$
HITS	0.9943	0.9838	0.9842	0.9322	0.2286	0.9874	$G > T > N > V > F > P$
FREQ	0.9873	0.9731	0.9747	0.9888	0.9797	0.9827	$F > G > T > P > N > V$

Table 5.5: **Summary of the result of overall evaluation with all different types of simulated users being ranked at the same time.** For spammers, the best (lowest) result is shown in bold font. As can be seen, only SPEAR was able to rank all three expert types at the top and retain the expected correct order. G=Geeks, V=Veterans, N=Newcomers, F=Flooders, P=Promoters, and T=Trojans.

5.5.5 Qualitative Analysis

In addition to the quantitative analysis of the simulation results, it is worthwhile to take a look at the ranking of real users produced by SPEAR in a qualitative way so as to gain more insight into its effectiveness.

We ran SPEAR on the data sets of four arbitrarily selected tags, namely `photography`, `semanticweb`, `javascript` and `programming`, where the last two are combined to form a conjunction as an example of running SPEAR on a more specific topic. We examined the Top users who are given high ranks by SPEAR in each of these data sets. While it would be difficult to provide an objective evaluation of the expertise of these users, we discovered that there were several things that were indicative of their expertise.

Firstly, many of these Top users were more likely to provide optional personal information in their Delicious account, including for example their real names, address of personal Websites, links to their photos on Flickr.com, and links to their Twitter.com microblogging account. This implied that they were more involved in using Delicious. Secondly, many of them have a lot of other tags used together with the corresponding tag in which they attain high expertise scores. For example, a Top user in `photography` has used 359 other tags together with `photography`, suggesting that he has an extensive collection of documents about the topic. Finally, we identified some "real" experts among the Top users. For example, two users who were ranked in the Top 10 in `semanticweb` turned out to be two researchers of Semantic Web technologies, while a third was found to be an active blogger of the same subject. The Top two experts ranked by SPEAR in `javascript` $\cap$ `programming` were two professional software developers. In contrast, all the users mentioned above were ranked lower by FREQ and HITS, sometimes even outside the Top 200.

As for spammers, we singled out the obviously heavily spammed tag in Delicious, `mortgage`, collected the bookmarking histories of the documents that were annotated with the tag[10], and run SPEAR, HITS and FREQ on it to rank the users. We wanted to find out whether spammers were really demoted by SPEAR and whether FREQ was vulnerable to spammers in this real setting. While we did not have a labeled list of the spammers as ground truth, we identified them manually by looking for several characteristics common to spammers. Spammers are usually automated bots. Hence, they either tend to extract words from the documents themselves (especially the title) and use them as tags, or use the same set of tags on a large number of documents even though the tags are not semantically related to the document content [MCM09a]. Also, some spammers aim at promoting their own content, and therefore many of their bookmarks are likely to be documents from the same domain (which can usually be classified as spam at first glance).

By looking for these characteristics of users who used the tag `mortgage`, we successfully identified 30 spammers in the 50 most active users. Obviously, this meant that out of the Top 50 users ranked by FREQ, 30 of them were found to be spammers. It is interesting that we even discovered a group of spammers whose usernames had the same prefix and were only different from each other in the numbers in the suffixes, suggesting that there exist spammers who submit spams in a more sophisticated way than merely flooding the system. As for the rankings produced by SPEAR and HITS, we observed similar results as we did in our simulations. All these 30 spammers were significantly demoted to below the 3000th rank by SPEAR and HITS, with ranks of these

[10]The data set of the tag `mortgage` was not among the 110 data sets we had collected in the beginning.

spammers in SPEAR much lower than those in HITS. We also observed that there were no spammers in the Top 50 ranks returned by SPEAR and HITS.

In addition, we also run FREQ and SPEAR on arbitrarily selected tags and examined the differences between the Top rank users. We found that very often users ranked at the Top by FREQ were quite the opposite of experts, not to mention that many of them were spammers. For example, for the tag `bridge`, a user was ranked first by FREQ because he had a large number of bookmarks with the tag. However, a closer look at his collection of documents in Delicious revealed that the majority of them were not related to any conventional meanings of the word 'bridge'. In contrast, SPEAR ranked this user much lower, at 2,088th out of the 3,144 users being ranked. The fact that this user was ranked low by SPEAR was that, despite the number of times he had used this tag, there were very few if any other users who would do the same thing as he did. In other words, although he was not necessarily a spammer, this user had few followers due to his idiosyncratic use of the tag. Arguably, SPEAR gave a more sensible result because other users were quite unlikely to benefit from this user with respect to the topic in question.

By this qualitative study, we showed that SPEAR also works reasonably well in a real setting. On the one hand, it is able to identify real experts. On the other hand, it is able to solve problems in day-to-day operation of collaborative tagging systems by demoting real spammers.

5.5.6 Analysis of Credit Score Functions

One important element of SPEAR is the credit score function $C(x)$ by which we assign higher scores to users who have tagged a document earlier and lower scores to users who have tagged the document at a later time. This credit score function actually directly affects the performance of SPEAR. If we do not apply the credit score function, SPEAR will be no different from the original HITS algorithm, in which every component in the adjacency matrix will either be 1 or 0.

Intuitively, with a credit function of larger second derivative[11] SPEAR should be more resistant to spammers. This is because the number of followers of a user is an important piece of information that allows us to distinguish between spammers from legitimate users. However, there is also a drawback when such an aggressive credit score function is used.

To give higher scores to users who have tagged a document at an earlier time will increase the chance of mistaking an inactive user as an expert. Consider a very popular document with 5,000 users, a certain user may happen to be the 100th user to tag this document, and therefore he has 4,900 followers with respect to this document. As a result, he will be assigned a an initial score of $x = 4,900$. Consider two credit score functions $C_1(x) = x^{0.2}$ and $C_2(x) = x^{0.8}$: $C_1(x)$ will return 5.47, while $C_2(x)$ will return 895.69. If C_2 is used, this user will receive an exceedingly high expertise score given this high credit score coupled with the probably very high quality scores of this popular

[11] In this case, credit scores for a user increase faster and faster when he has more and more followers.

document. Other expert users who have tagged many more high quality documents will find themselves ranked lower than this user only because they are followers of him in this particular document. This will be a problem because this inactive user is very unlikely to benefit other users.

To investigate how the credit score function affects the ranks of these inactive users, we conducted experiments on some selected data sets with different credit score functions. Firstly, we randomly picked three tags from our data sets: `film`, `history` and `iphone`. For each of these data sets, we run SPEAR to obtain a ranking of the users involved by using different credit score functions of the form $C(x) = x^y$, where y ranged from 0 to 1.0 (in the case of y=0, the SPEAR algorithm effectively becomes HITS). While it is true that there are many other types of functions that can be considered here, this class of functions should be sufficient in allowing us to have a better understanding of the behavior of SPEAR, as it provides us with functions with different second derivatives, in which we are most interested. We then examined for each of the tags the ranks of the users who were found to have only tagged the most popular document in the respective data set.

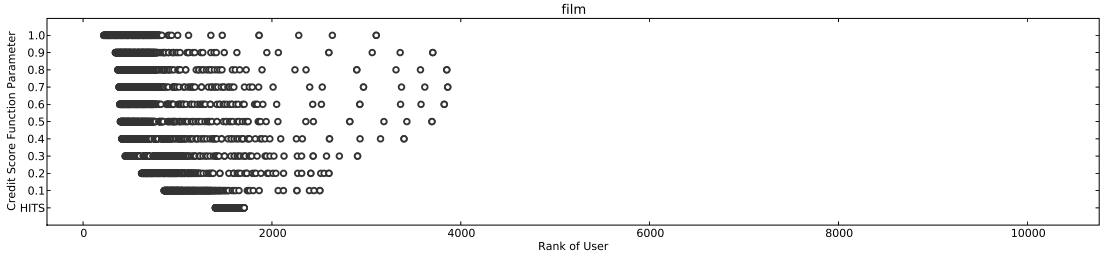
Figure 5.11 shows the ranks of users who have only tagged the most popular document in each of the three data sets, with SPEAR operating under different settings of credit score function. We can see that the differences between credit score functions show similar effects on the ranking of these inactive users. Credit score functions with greater values of y tend to spread the users across a wider range. This is due to the fact that these credit score functions assign scores that spread a wider range of values. However, these functions also tend to rank some inactive users quite high, especially when they tagged the most popular document at a very early time.

On the other hand, credit score functions with smaller values of y tend to clump users in small range of ranks. At the extreme end where $y = 0$, all of the users under consideration are assigned the same expertise score. A merit of these functions is that they tend to give lower range to these users on average. Therefore they also have a smaller chance of mistaking these users as expert users. However, as we have shown in our simulations, HITS, which is SPEAR with $y = 0$, performed relatively poorer than SPEAR where we set $y = 0.5$. In other words, smaller values of $y = 0$ would also make SPEAR more vulnerable to spammers.

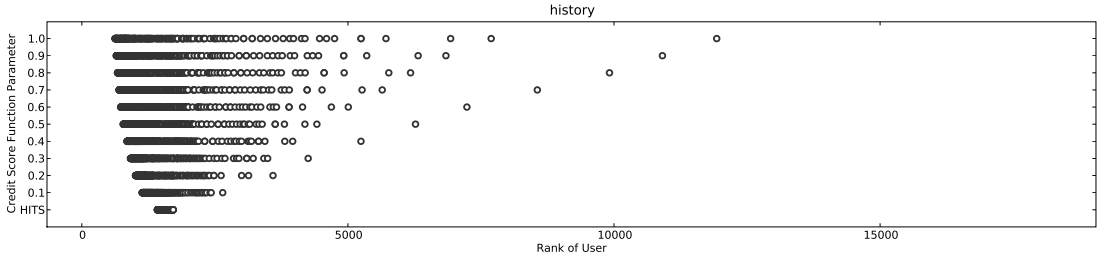
Different credit score functions have different merits and weaknesses. Therefore there is no single correct choice of credit score function for SPEAR. In settings where spamming activities are commonly observed, functions with greater values of y or other functions with similar characteristics should be used. On the other hand, in settings where there are few spammers, one may consider to use functions with smaller values of y or other functions with similar characteristics.

5.5.7 Excursus: Document Quality in Folksonomies and in the Web

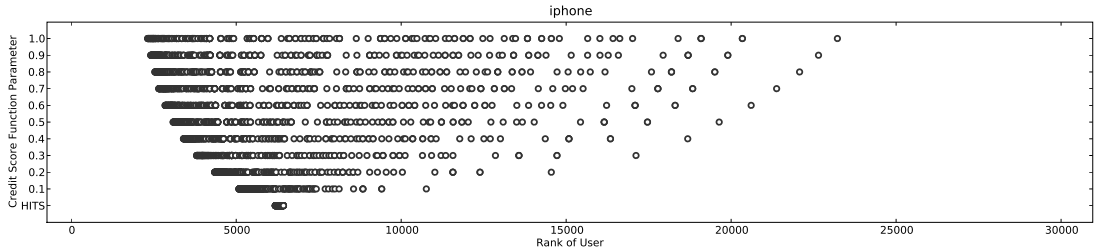
Collaborative tagging systems and folksonomies are not isolated from the rest of the Web. While we focus our discussion and evaluation of SPEAR on its ability to rank users according to their expertise, the algorithm also computes a quality score for doc-



(a) Tag `film`



(b) Tag `history`



(c) Tag `iphone`

Figure 5.11: **Ranks of users who have only tagged the most popular document for each of the three selected tags `film`, `history` and `iphone`.** Only these users are represented by the circular symbols. Other users in the data sets are not shown.

uments. This quality score can be very useful for providing a ranking of documents in folksonomies and also for Web information retrieval in general. As an illustrating example, Table 5.6 shows the Top 5 documents returned by SPEAR for the `photography` data set. Even at first glance, the list contains documents which appear very relevant to photography in general, including quite a number of online tutorials on different aspects of photography. For instance, the first document is a very detailed technical tutorial of photography describing basic concepts and introducing different shooting techniques. The fifth document provides a legal summary of photographers's rights when stopped or confronted for photography.

We were therefore interested in finding out whether there is a relationship between

the folksonomy-derived quality scores of documents as returned by SPEAR and the popularity of these documents on the Web as indicated by their Google PageRank, which is based on a hyperlink analysis of the Web graph [BP98]. A major difference between these two scenarios is that rankings derived from folksonomies are based on the activities of Web readers whereas the rankings derived from the Web's link structure is based on the activities of Web authors.

A problem, however, is that the data required for such an analysis is difficult to obtain in practice. While the volume of our experimental data sets collected from Delicious is rather large, it is still only a snapshot of its full folksonomy. Similarly, the PageRank information that may be queried from Google is comparatively coarse. Still, we argue that our experiments described below can show a general direction of the relationship between a document's value in folksonomies and its value in the Web graph.

Top Rank	Web document
1	http://www.berniecode.com/writing/photography/beginners/
2	http://www.diyphotography.net/
3	http://strobist.blogspot.com/2006/07/how-to-diy-10-macro-photo-studio.html
4	http://digital-photography-school.com/blog/
5	http://www.krages.com/phoright.htm

Table 5.6: **Top 5 documents returned by SPEAR for the `photography` data set.**

Firstly, we investigated how the subsets of the highest quality and lowest quality documents in our data sets compare with the total set of all documents in terms of Pagerank information. We created two subsets of Web documents for this experiment, namely *SPEAR-TOP* and *SPEAR-BOTTOM*, which contained the Top 100 documents and Bottom 100 documents, respectively, from each of our 110 real-world data sets. We discarded 6 out of the 110 data sets because they were comprised of less than 200 documents. This step yielded a total of $104 \times 100 = 10,400$ documents for each of *SPEAR-TOP* and *SPEAR-BOTTOM*. Next, we queried the Google search engine for PageRank information of all Web documents in our real-world data sets, and compared the PageRanks of all documents with those in *SPEAR-TOP* and *SPEAR-BOTTOM*. The results are shown in Figure 5.12 and Table 5.7.

We observed that high quality documents in SPEAR tend to have higher PageRanks (*PR*) than a random selection of documents. Similarly, low quality documents in SPEAR tend to have lower PageRanks. This indicates that there is a kind of correlation between the folksonomy-based rankings of SPEAR and the hyperlink-based rankings of PageRank. For verification, we computed the Pearson-*r* correlation coefficient [Ric95] of the complete (i.e. not only Top and Bottom) document rankings as returned by SPEAR and PageRank for each real-world data set. The mean Pearson-*r* correlation coefficient across all 110 data sets was $\bar{r}_{arithm} = +0.324$ ($\sigma = 0.146$), i.e. a weak positive correlation. The *p*-values were ≤ 0.05 for all but eight data sets; most of the latter had

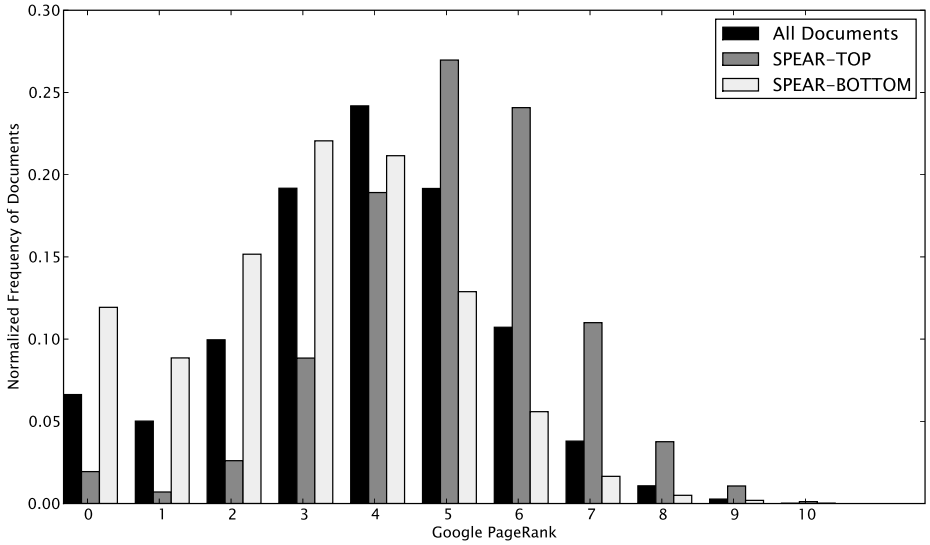


Figure 5.12: Google PageRank distribution for all documents, *SPEAR-TOP* and *SPEAR-BOTTOM*. The plot shows the shifts of high quality documents towards higher PageRanks, vice versa for low quality documents.

Documents	**Mean** *PR*	**Std. Dev.**	**Median** *PR*
All	3.71	1.81	4
SPEAR-TOP	5.05	1.61	5
SPEAR-BOTTOM	3.05	1.81	3

Table 5.7: **Google PageRank (*PR*) statistics for all documents and those in *SPEAR-TOP* and *SPEAR-BOTTOM*, respectively.** We observed clear shifts towards higher PageRanks for documents in *SPEAR-TOP* and towards lower PageRanks for documents in *SPEAR-BOTTOM*.

less than 100 documents in total, i.e. the sample size was comparatively small. Under the assumption that SPEAR is reasonably able to measure the quality of a document within a folksonomy, this result suggests that there is a correlation between the "value" of document within a folksonomy – driven by Web readers – and its value within the hyperlink graph of the Web – driven by Web authors. It is also an indication that the algorithmic outcome of SPEAR is reasonable in principle.

On the other hand, the rankings of SPEAR are still quite different from PageRank as is exemplarily shown in Figure 5.13 for the data set `entertainment`. Here, the Top #1 document for PageRank with Google PageRank value of 10 (*PR10*) was the well-known news site *CNN.com*. However, CNN.com was only ranked #250 by SPEAR, which is even lower than the highest-ranked *PR0* document for SPEAR at #194. Interestingly,

the latter *PR0* document automatically redirected via an HTTP header `301 Moved Permantently` to the home page of *The View*, a popular ABC talk show, which itself has a high PageRank value of *PR8*. This might be an indication that SPEAR could identify the value of the document while PageRank failed. However, we must also consider that the *PR0* document in question did not display any content of its own but rather redirected to another Web document – which might be the reason why Google's PageRank implementation assigned it a low *PR0* value in the first place.

Overall, only two documents in the Top 20 list of PageRank were present in the Top 20 list of SPEAR. For the record, the Top #1 document for SPEAR was *eOnline.com*, a *PR7* Web site on entertainment news and celebrity gossip.

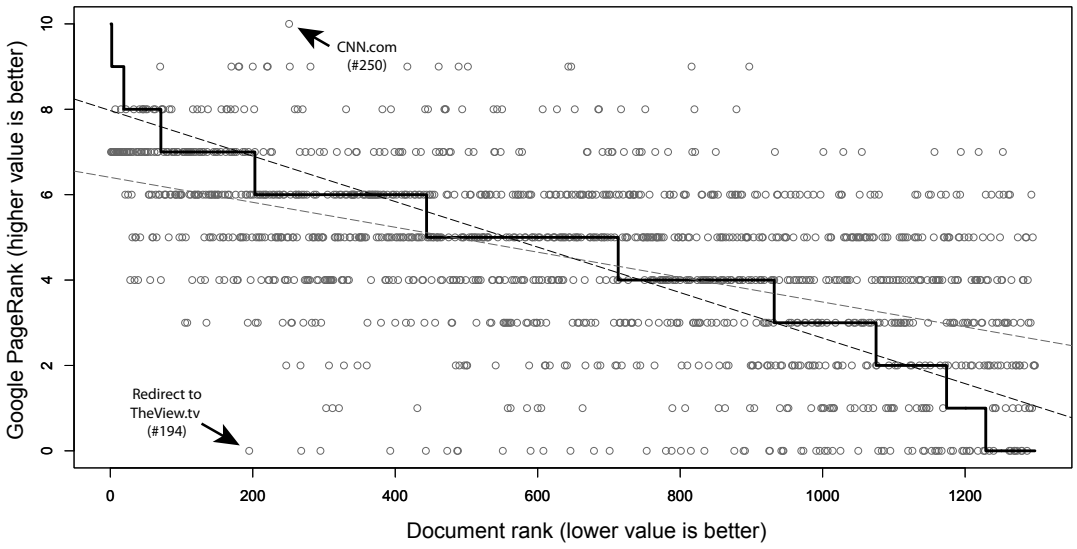


Figure 5.13: **Google PageRank distribution from *PR0* (lowest) to *PR10* (highest) for the data set `entertainment`.** The solid, staircase-shaped line shows the PageRank distribution of documents when ranked by their *PR* value; the gray circles denote the PageRank distribution of documents when ranked by their SPEAR quality score. The dashed lines in black and gray show the least squares regression lines for ranking by PageRank and SPEAR, respectively.

In summary, our findings suggest that there is a correlation between the "value" of Web documents in folksonomies (driven by Web readers) and its value in the Web graph (driven by Web authors). However, our results also indicate that folksonomies can yield information about Web documents through algorithms such as SPEAR that traditional ranking algorithms cannot derive from an analysis of the Web's hyperlink structure.

5.6 Discussion

Our experiments described in the previous sections have shown that our proposed algorithm, SPEAR (Spamming-resistant Expertise Analysis and Ranking), produced better rankings than the evaluation baselines, i.e. the original HITS algorithm and a simple frequency counting algorithm. It distinguished reasonably well between different types of experts, and it consistently demoted different types of spammers and removed them from the top of the rankings[12]. In other words, SPEAR was able to detect the subtle differences between good and bad users, and to demote spammers while still keeping the experts at the top of the ranking. We note that SPEAR measures expertise mainly based on a user's ability to discover (new) high quality content, which is but one aspect of an expert's skill set in the real world. However, a primary goal of collaborative tagging systems is to identify high quality resources, so the expertise aspect analyzed by SPEAR is very relevant in these systems.

There are a number of reasons of why an expert ranking algorithm is needed in folksonomies. Firstly, with increasing number of documents for a given tag, it becomes increasingly difficult to retrieve documents which are useful and of good quality. One way to solve this problem is to first identify the experts, and then browse their collection which should contain good documents. On the other hand, by keeping an eye on the collection of an expert, we are able to benefit from notification when he adds new and useful documents to his collection.[13]

Our experiments have also shown that a simple technical adaptation of traditional ranking algorithms to the data model of folksonomies does not yield satisfying results, even if they employ a mutual reinforcement scheme to go beyond simple frequency measures of expertise. While SPEAR achieved encouraging results for ranking users, HITS and FREQ particularly failed with regard to their resistance to spamming activities. We argue that this is caused by a fundamental difference between scenarios of analyzing the Web's hyperlink structure and analyzing the usage patterns in folksonomies. In the former case, a Web author can indiscriminately increase the number of *outgoing* hyperlinks of one of his Web documents to other documents, but it is much more difficult to manipulate the number of *incoming* hyperlinks from non-affiliated documents to his Web document because those documents are not under his control. Traditional ranking algorithms in Web information retrieval such as HITS or PageRank benefit from this characteristic that results in a higher resistance to spamming activities on the Web. In a folksonomy, however, the only source of information about a user's expertise is his personomy $\mathcal{P}_u$, i.e. his collection of bookmarks and tags. Because documents in a folksonomy cannot link back to users, this means that measuring the expertise of a user is mainly based on data that is completely under his own control. Hence, we ar-

[12] Following the taxonomy of anti-spam approaches in folksonomies presented by Heymann [HKGM07], SPEAR is a combination of a detection-based approach (analysis of user behavior) and a demotion-based approach (spam-hardened rankings).

[13] Currently, Delicious allows users to subscribe to a particular tag or to become a follower/fan of another user. However, there is neither a measure of a user's expertise nor a recommendation of related experts in your areas of interest given your own user profile.

gue that the reason of the better performance of SPEAR is its higher level of detail in terms of data analysis and its notion of implicit endorsement, i.e. the differentiation between discoverers and followers based on temporal information. Particularly, because temporal information of user activity in a folksonomy is managed by a trusted entity – the collaborative tagging system itself – it cannot be manipulated by users to game the system and artificially boost their expertise. This increases the difficulty for spammers to gain high user ranks in SPEAR whereas it is comparatively easy in the case of traditional ranking schemes such as HITS.

Although we only discuss expert ranking in the context of folksonomies and collaborative tagging, SPEAR is in fact applicable to many different situations because it assumes a very general model of user-resource interactions. For example, it can also be applied to online services such as the social news site Digg[14] or the microblogging service Twitter[15], which are very popular among Web users nowadays, to rank users by their expertise in a given topic. Another area in which SPEAR could be used is in measuring the expertise of authors of scientific papers, similar to studies such as [MR08] that apply PageRank to citation networks.

Possible Improvements of SPEAR

While we have seen that SPEAR shows good performance for ranking of users according to their expertise in a particular topic, we have identified some opportunities for improvement.

Firstly, SPEAR may mistake inactive users as expert users, especially when these users were once the discoverers of very high quality documents, as we have shown in our analysis of the credit scoring function. A related idea is that of "recency of information", i.e. how recent and up-to-date user-contributed information is. It is reasonable in our scenario that a user who has been more active recently should be given more credit than a user who only discovered several popular documents in the distant past and ceased contributing thereafter (scenario of a "retired researcher"). Hence, it would be desirable to incorporate certain measures for reducing the weight and impact of old user activities into SPEAR. This will make it easier for new users to rise to the top of the expert ranks and prevent older users to have an undue influence. On the other hand, it would also make SPEAR's user and document ranking scheme more trend-aware, for instance to the benefit of document recommendation schemes.

Secondly, SPEAR focuses on user activity in a document's timeline. A tag-based analysis is only performed in a pre-processing stage for filtering documents and users by topic (where a topic is represented by a tag or a combination of tags) to produce the topic-sensitive input data of SPEAR. This leads to two limitations of our approach. The first limitation is that it overlooks users who have used related tags, such as synonyms, of the tag chosen for analysis. For example, when ranking users for the topic `javascript`, should we also consider users who are ranked high in `programming`? While one can currently specify a conjunction or disjunction of related tags for creating

[14] Digg, http://www.digg.com/.
[15] Twitter, http://www.twitter.com/.

the topic-sensitive input data of SPEAR, it is usually difficult to know all related tags of a particular tag beforehand. Hence, it could be desirable to integrate some kind of tag co-occurrence analysis into SPEAR to produce a more comprehensive user ranking.

Lastly, SPEAR could benefit from an analysis of the tagging vocabulary of users for increasing its robustness against spammers who assign incorrect tags to documents. As discussed in Section 2.6, there are other anti-spam approaches that tackle spammers by focusing on an analysis of the tag usage of users [KEG+07, KFG+07, MCM09a, NO09], i.e. the tags that users selected to annotate documents. We believe that these approaches and SPEAR are complimentary to each other, and that a combination could result in an even better user ranking algorithm.

5.7 Summary

In this chapter, we have presented our study of ranking users in folksonomies according to their expertise in a particular topic. We have proposed the SPEAR ranking algorithm, which is based on the relationship of mutual reinforcement between users and documents as well as the notion of implicit endorsement via an analysis of the temporal dimension of tagging activity. Our experiments have shown that the algorithm is effective at promoting expert users and demoting spammers at the same time. As such, we have shown that an appropriate method such as SPEAR is able to gain a better understanding of the characteristics of users – in our scenario, information about their expertise or trustworthiness – by analyzing their collective behavior in folksonomies. Hence, our results have also shown that the activities and implicit interactions of users can be exploited to derive information from folksonomies that is not explicitly expressed anywhere, thus supporting our hypothesis regarding user expertise in folksonomies.

In the next chapter, we will present our approach to personalization of Web search by exploiting folksonomies for profiling of users and Web documents, and demonstrate how the approach can be implemented in practice.

A question that sometimes drives me hazy: am I or are the others crazy?

Albert Einstein (1879—1955)

6

Web Search Personalization with Folksonomies

We have seen in the previous chapters that there are differences in the perception of resources on the Web between their authors and their readers, and differences in the resources' popularity or importance when measured by an analysis of the Web's hyperlink structure (driven by Web authors) and an analysis of the activity and implicit interactions of users in folksonomies (driven by Web readers). The domain of Web search is arguably the most prominent example where the analysis of the Web graph is exploited to create applications and services for Web users. In this chapter, we propose a method that combines both perspectives for the personalization of Web search. Namely, we describe how we can understand the characteristics of users and resources through an analysis of folksonomies and show, at the example of Google, how the search results of a traditional search engine can be re-ranked according to this folksonomy-derived information. As such, our approach can be considered as an integration of the collective behavior of Web users into traditional Web search[1].

In the following sections, we describe our proposed Web search personalization approach and demonstrate how it can be implemented in practice. We present our analysis of experimental results and test our hypothesis with regard to information about users and Web resources in folksonomies:

> **Hypothesis 3 (Web Search Personalization)**:
> Folksonomies provide sufficiently rich information about users and Web resources to allow for the personalization of Web search, i.e. an individualized search for resources on the Web.

[1] Seen this way, our proposed personalization technique is related to the notion of third generation search engines described by Broder [Bro02]. In his Web search taxonomy, he characterizes the third generation of search engines as those approaches which attempt to blend data from multiple sources in order to answer "the need behind the query".

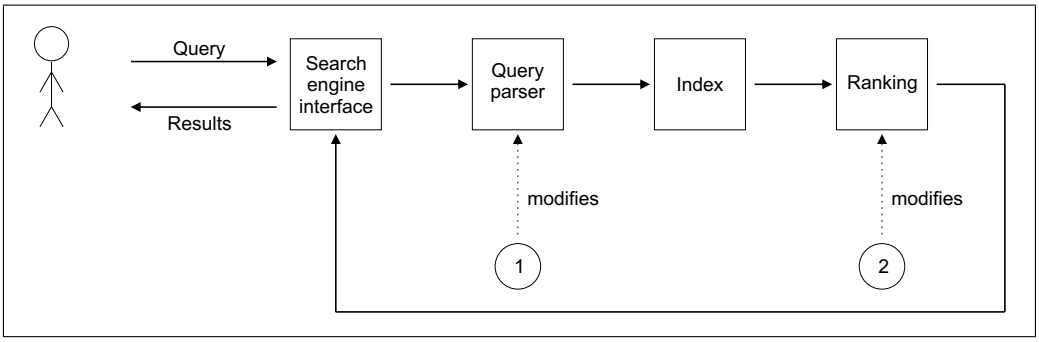


Figure 6.1: **Basic process of Web search.** Search query refinement (1) influences query parsing, whereas search result processing (2) influences the ranking.

6.1 Web Search and Personalization

6.1.1 Web Search

The domain of Web search is one aspect of the broad area of Web information retrieval. While a comprehensive introduction to Web search is beyond the scope of this thesis, we provide a brief summary of the basic search process and its most important aspects in this section.

The search process starts with a user entering a query into a search engine, typically by using search keywords such as "`landscape photography switzerland`". The search engine parses the query and performs a lookup at its index to provide a listing of search results, i.e. an ordered list of the best-matching Web documents. The usefulness of search results depends on the relevancy of the returned Web documents. While the set of documents on the Web that include a particular word may be very large – a query for "`computer science`" on the search engine Google returns 32 million results – some of these documents may be more relevant, important or authoritative than others. For this reason, search engines perform a ranking of search results in order to present the best documents to users first (cf. Section 2.7). Typical examples of such ranking algorithms are HITS [Kle98] and PageRank [BP98]. In the final step, the ranked list of search results is presented to the user through the search engine interface. Figure 6.1 illustrates this process.

6.1.2 Personalization of Web Search

The drawback of Web search as described in the previous section is that the returned search results are the same for any user given the same query. As such, this traditional approach to Web search has been rather impersonal due to lack of adaptability to a user's individual preferences and topics of interests. The goal of Web search personalization is to address this need for a more individualized Web search experience, and it has been shown to yield considerable improvements compared to non-personalized search [SHY04, TDH05a, TDH07]. For achieving this goal, it integrates user-specific

data into the process of finding the best-matching documents to a search query, thus increasing the amount of input information available to search algorithms.

We can differentiate two general methods of Web search personalization [PSC+02, XBF+08]: The first approach is *search query refinement*, which modifies or augments a user's original query, thus influencing the query parsing component in Figure 6.1. For instance, a query for "`IR`", when issued by a researcher in the domain of information retrieval, might be translated to "`information retrieval`" [FD97, JRMG06]. Similarly, the original weight of each query term could be changed. There exist several techniques to implement search query refinement such as [RD06, CFN07]. Liu et al., for example, examine the search history of users in order to assign each user to a set of categories [LYM02]. When a user submits a query, the search engine will add his categories to the query so that the search results may be personalized to the user's topics of interest.

The second approach is *search result processing*. Here, the user's query is left as is and instead the original order of search results is re-ranked based on information about the individual user, thus influencing the ranking component in Figure 6.1. Again, there exist several techniques to implement search result processing [Hav02, Hav03, SHY04]. Teevan et al., for example, examine the files on a user's personal computer in order to derive user preferences for re-ranking of search results [TDH05b].

In the following sections, we will describe how we can derive information about users and documents from folksonomies and present our proposed method to leverage this information for personalizing Web search through re-ranking of search results.

6.2 Folksonomies and Web Search

If we want to leverage folksonomies for Web search personalization, we must first have an idea whether both usage scenarios – *searching* for Web resources and *tagging* them after they have been discovered and deemed useful – are sufficiently related so that positive results can be expected from such a personalization approach.

Several recent studies – including our research work described in Chapter 4 – have investigated and compared the user activities in the domains of collaborative tagging and Web search. Similar to the stabilization of folksonomies described in Section 2.4.4, Wedig and Madani [WM06] found that users' topical interest distributions in the context of Web search become distinct from the population, and converge to a stable distribution. In their study of the collaborative tagging systems Delicious (Web resources in general), Last.fm (music) and Flickr (images), Bischoff et al. [BFNP08] found that the majority of tags can indeed be used for search, and that in most cases tagging behavior show approximately the same characteristics as searching behavior. The research works of Krause et al. [KJHS08, KHS08] support these findings. They investigated the distributions of tags and search query keywords and found that both exhibit similar characteristics and dynamics. For example, they could observe power-law and small-world patterns in both scenarios (cf. Section 2.4.4). Their results also indicate that tagging and searching behavior are indeed triggered by similar motivations. For

example, they found evidence that popular events (e.g. political campaigns) trigger both search and tagging activities close to the event. Krause et al. also report that, for each search query, both traditional search engines and folksonomies focus on basically the same subset of documents on the Web[2]. Heymann et al. [HKGM08] observed that the users share a common distribution of the employed vocabulary when tagging and searching, i.e. similar tags and search keywords are selected by users. They found that popular search keywords and tags overlap significantly, and argue that folksonomies can help with queries where tags overlap with search keywords. Outside the context of folksonomies, Agrahri et al. [AMR08] investigated user collaboration in general for improving the relevance of Web search results. They have shown that people are biased towards documents at the top of the search result lists (even if the list is randomized). However, they found that the explicit feedback of people – and tagging is a kind of explicit feedback – is not biased. They also observed that people's shared preferences do not always agree with a search engine's result order. They therefore argue that social search techniques might improve the effectiveness of search engines. Li et al. [LGZ08] report that tags are suited for similarity computation of Web documents. They conclude that tag-based topic clustering and similarity computation is not only simple and accurate, but also cost-effective in computation because the dimension of term vector space can be significantly reduced.

All these studies provide strong support for the general applicability of folksonomies to Web search personalization. Nevertheless, we have reported our own comparison of folksonomies with other Web-related data and metadata including search queries in Chapter 4. And while we have found similarities between tags and search keywords, too, we have also observed a closer similarity of tags to classification information of Web resources. In other words, users tend to use tags rather for the classification and categorization of Web documents as in the traditional field of subject indexing (cf. Section 2.3.1). We therefore argue that, instead of basing a personalization method on the query step as in search query refinement (i.e. focusing on the similarity of search keywords and tagging data), it may be even more beneficial to leverage folksonomies for search result processing. As we have seen in Chapters 2 and 4, we may derive a user's personal preferences and topics of interests from his personomy $\mathcal{P}_u$ and, similarly, derive topical information or the "aboutness" of a document from the collective tagging activities of all users in a folksonomy. Hence, we can use folksonomies in a first step to construct appropriate profiles for both users and documents, and in the second step we can compute the similarities between these profiles for the purpose of re-ranking search results.

We believe that such an approach has have several advantages. Firstly, it does not rely on an overlap between tags and search keywords, thereby increasing the likelihood of applying the approach also to non-popular queries [HKGM08]. Secondly, we have found in Chapter 4 that tagging data in folksonomies is less suited for finding

[2]This finding of Krause et al. also supports our notion in Chapter 3 that folksonomies cover a considerable fraction of "relevant" documents on the Web, i.e. such documents that are actually perceived by users as being valuable in one way or the other.

the specific "needle in the haystack" with regard to resource retrieval. However, if we can combine such an approach with existing search engines, we can "outsource" the task of retrieving the base input set of documents for subsequent personalization to these engines. Lastly, the decoupling of our proposed personalization method from a user's query and its search keywords also means that the approach might be extended to other domains in Web information retrieval, for example recommendation of Web documents.

6.3 Folksonomy-driven Personalization

In this section, we describe our proposed approach for Web search personalization through search result processing with information derived from folksonomies.

The basic idea is to use a traditional search engine for processing a user's query in order to return a first list of ordered search results. Then, data from folksonomies is used to re-rank this initial list of Web documents *on the client side*, e.g. the user's Web browser, according to the user's topics of interest. Even though they do not explicitly refer to folksonomies[3], our personalization technique thus falls into the first category of the taxonomy of re-ranking approaches described by Bharat and Mihaila [BM02], because it effectively uses folksonomies to derive human classifications of Web documents. Similarly, our approach is also an implementation of the notion of Markines et al. [MCM+09b], who argue that folksonomies allow us to extend the assessment of what a Web resource is about from content and link analysis algorithms to the collective "wisdom of the crowd"[4].

While we will demonstrate how our personalization approach can be implemented at the example of the Google search engine, the approach itself is independent of the search engine being used. Hence, users are free to use their favorite search engine.

Because the approach makes use of the search results as returned by a search engine, its performance depends on the *size* and the *quality* of the initial search result list. With regard to the size of the search result list, we focus our studies in this chapter on the use case where an ordered list L of Web documents with size $|L| = 10$ is returned as the result of a search query. The reason is that $|L| = 10$ is the default value of most popular search engines such as Google, Yahoo! and Microsoft Bing, and is also often used in the literature. Evaluating the approach with a search result list of ten Web documents will therefore contribute to a better comparison of experimental results with other scientific studies and also yield more insights into its performance in a practical setting. However, our approach itself is not restricted to a specific number or upper limit of search results per query.

With regard to the quality of search results in the context of personalization, our proposed method benefits from the strategy of search engines to distribute their Top

[3] The study of Bharat and Mihaila [BM02] was conducted in 2002, i.e. before collaborative tagging and folksonomies becoming popular to a wider audience.

[4] For example, if many people agree that a Web resource is about programming, then with high probability it is about programming even if its content does not include the word "programming".

Figure 6.2: **Google search results for "`jaguar`"**. As can be seen in this exemplary query, the search engine distributes its Top search results among the various meanings and topics of the query.

search results among the various meanings and topics of a query [WM06]. For instance, a query for "`jaguar`" on Google returns in the Top 10 search results Web documents about the car, the feline and the Mac operating system of the same name in order to increase the chance that at least one of these topics matches the user's intended search (see Figure 6.2). In other words, the results returned by search engines represent a range of intentions that people associate with queries. The study of Teevan et al. [TDH05a] supports this finding. Particularly, they argue that personalized search systems could take current Web search results as a starting point for user-centric refinement via re-ranking [TDH05b]. They report that the original ranking of results by a Web search engine is a useful source of information for a more personalized ranking, and, as they discovered, the first several results are particularly likely to be relevant.

Our proposed personalization method is mainly based on two elements: Firstly, the profiling of both users and documents from folksonomy data, and secondly, the computation of a similarity score between user and documents profiles, which is used for the actual re-ranking step. In the following sections, we will describe in detail the various steps in the personalization process, and demonstrate how the approach can be implemented in practice.

6.3.1 Data Collection

In this section, we describe how the input data for our personalization approach is collected.

The traditional scenario of Web search personalization involves search engines collecting such data from people using their services. In our case, however, input data is contributed by folksonomy users through social bookmarking and tagging activities. Hence, our approach leverages data that is provided by users in a variety of scenarios (e.g. browsing the Web, receiving recommendations from friends), of which Web search is but one. It also means that a user's personal data, i.e. the personal bookmark collection in his personomy $\mathcal{P}_u$, is not stored by search engines but the respective collaborative tagging system. A user thus can benefit from his tagging activities not only through the personalization of a single Web search engine, but theoretically of any search engine or similar service on the Web. In other words, users can update their personal data wherever they happen to be on the Web, and similarly, make use of their personal data wherever they are. Another benefit of a folksonomy-driven approach is that it can also leverage data from users' tagging of such Web documents that are not publicly accessible[5] (e.g. intranets or access-restricted Web sites) [LV03, JKHS08] or newly created, unlinked Web documents [HKGM08, KJHS08]. Generally, search engines cannot index such documents and therefore a) cannot return these Web documents in their search results and b) cannot collect data from the activities of users searching for these documents.

Collaborative Tagging

As we have discussed above, we use the data contributed by users through social bookmarking and collaborative tagging as input information for our personalization technique. This data is available in the folksonomies and accessible through the respective collaborative tagging systems.

The quality and relevance of this input data depend on the following two assumptions about bookmarking and tagging:

1. Users primarily bookmark and tag Web documents that are in one or the other way useful.

2. Users have an incentive to add meaningful tags to their bookmarks, particularly such tags that can be used for classification purposes.

The studies and research work described in Chapters 2 and 4 provide strong support for these assumptions. In other words, the *act* of tagging and the actual *tags* can be treated as explicit positive feedback[6] with regard to the affected documents and the

[5] This class of Web documents that are not publicly accessible is also called "the Deep Web" [LV03], a term coined by the computer scientist Mike Bergman [Wri09].

[6] Of course, it's also true that the lack of a tag assignment doesn't necessarily mean irrelevance [XBF+08]. For example, the user might have simply forgotten to add a specific tag. This problem can be mitigated

topics they are about, and also with regard to the user's topics of interests. We have also shown in Chapter 4 that folksonomies contain data about Web documents that is not directly contained in the documents' contents or in the metadata supplied by their authors, which suggests that integrating tagging information can further help to improve our personalization method.

The research domain of personalization techniques often differentiates between implicit and explicit data collection from users. In the former case, the behavior and activities of users are tracked and subsequently analyzed to understand the characteristics and intentions of users. In the latter case, users are explicitly asked about their interests, intentions and similar information. The primary advantage to using implicit measures is that such techniques remove the cost to the user of providing explicit feedback [CR87], whereas explicit measures are generally thought to be more accurate with regard to the user's real interests and preferences [Nic97, KT03].

From a conceptual point of view, data collection from collaborative tagging and folksonomies is a mixture of explicit and implicit collection techniques. On the one hand, users are not prompted to enter their preferences or topics of interests explicitly in a special configuration step. On the other hand, they are also not monitored or tracked in the background as is the case for search engines and the analysis of search query logs. Instead, personal data is collected rather explicitly because tagging Web documents is a manual user task, and users thus know exactly when such data is collected. However, unlike explicit collection measures, this manual task is not an additional burden for users as we have seen in Chapter 2. Additionally, we can further alleviate this burden by providing tag suggestions and recommendations to users as discussed in Section 2.4.3 in order to close the usability gap to fully implicit techniques. Furthermore, we present an easy way of semi-automated tagging of Web documents called *tagmarking* in the next section that specifically targets the scenario of Web search.

It should also be noted that while users may search and tag within close time intervals when performing searches on the Web, the act of tagging a document is not comparable to providing explicit feedback on their interests to a specific search engine. In the former case, users add interesting Web documents to their personal collections and may share them with other users – thus benefitting from their tagging activities beyond the search experience on a single search engine.

Tagmarking

We have seen in Section 6.2 that there is a similarity between search keywords and tags. The notion of *tagmarking*, as we call it, exploits this similarity. The basic idea of tagmarking is to allow a user to automatically tag a document that was found through an interaction with a search engine by annotating the document with the search keywords extracted from the user's query.

by the integration of techniques such as tag co-occurence analysis, which identifies tags that are related to a particular tag. In Chapter 7, we will also introduce an extension to the normal model of tagging that is able to account for negative feedback of tag assignments with regard to the affected documents.

In our system prototype, which we discuss further below, we have developed a Web browser extension that integrates tagmarking as a simple one-click button into the browser's user interface. While the user evaluates search results, the browser extension keeps track of his interactions with the search engine and stores his most recent search query, e.g. "`gutenberg poe raven`", in memory. Whenever the user finds a Web document that is relevant to his query, he can conveniently annotate the document and add it to his personal collection through a single click on the Tagmark button. The browser extension will automatically translate the search query to tags and add them to the respective bookmark (in our example, the tags would be `gutenberg`, `poe` and `raven`), thus saving the user from providing these tags himself. Of course, the user can still manually specify those tags he deems most appropriate for the document.

Tagmarking blends the scenarios and user experiences of searching for Web documents and tagging those documents[7]. Particularly, it reduces the effort of bookmarking and tagging a document, and thus closes the usability gap to personalization approaches based on fully implicit data collection as described above.

6.3.2 Profiling Users and Documents

In this section, we describe how the input data presented in the previous section is aggregated into profiles of users and Web documents.

A widely used model in the domain of information retrieval in general and Web search in particular is the *vector space model* [MRS08]. In this model, Web documents and search queries are represented as vectors in a common vector space according to the terms extracted from either type of data. For example, a document d containing the phrase "a man who buys a piano, owns a piano"[8] would be mapped to a document vector $\vec{v}(d)$, where the components are set to the respective term frequencies in alphabetical order:

$$\vec{v}(d) = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 2 \\ 1 \\ 1 \end{bmatrix} \Leftarrow \begin{matrix} a \\ buys \\ man \\ piano \\ owns \\ who \end{matrix} \qquad (6.1)$$

The same mapping can be achieved for search keywords in queries. Additionally, several refinements may be used to adapt the vector representation of documents and queries, for example by weighting terms with measures such as *Term Frequency–Inverse*

[7] A related work to our idea of tagmarking is the study of Jaschke et al. [JKHS08], who try to extract folksonomies from search query logs. Here, search keywords extracted from a user's query are considered as implicit tag assignments to such Web documents that the user subsequently visits through clicking on the search results of his query.

[8] Quote by pianist Vladimir Samoylovich Horowitz (1903-1989) about the difference between physically possessing a piano and working hard to master the instrument.

Document Frequency (TF-IDF) [SB88]. A very interesting property of representing documents and queries as vectors is that we can compute their relatedness through similarity measures such as *cosine similarity* (cf. Section 4.3.7). These similarity measures can then be used to retrieve those Web documents from a search engine's index that are most similar to a user's query. Though simple, the vector space model has shown an amazing effectiveness and efficiency in practice.

Inspired by this model, we propose to model the profiles of users and documents using a *topic space*, where each dimension of the topic space represents a topic. As we have seen in the previous chapters, tags are very suited to derive topical information about users and documents. We therefore use tags as reasonable estimates of the topics in the topic space. The profiles of users – i.e. their interests in topics – and the profiles of documents documents – i.e. their topics or aboutness – can thus be directly inferred from tagging data in folksonomies.

Hence, we may derive a user's personal preferences and topics of interests from his personomy $\mathcal{P}_u$ and, similarly, derive topical information or the "aboutness" of a document from the collective tagging activities of all users in a folksonomy.

User Profiles

As described above, we derive a user's interests in particular topics from his tagging activities in a folksonomy $\mathcal{F}$, namely his personomy $\mathcal{P}_u$, which we have defined in Section 2.2. To recall, the personomy $\mathcal{P}_u$ of a user u includes his tag assignments $\mathcal{Y}_u = \{(t, r) \in \mathcal{T}_u \times \mathcal{R}_u\}$. Hence, we can use these tag assignments to derive a user-specific tag-document matrix $\mathbf{M}_u$ of size $|\mathcal{T}| \times |\mathcal{R}|$ in the topic space:

$$\mathbf{M}_u = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix}, c_{i,j} \in \{0, 1\} \tag{6.2}$$

where the components $c_{i,j}$ are initialized as

$$c_{i,j} := \begin{cases} 1, & \text{if the user } u \text{ assigned tag } t_i \text{ to document } d_j \\ 0, & \text{else} \end{cases} \tag{6.3}$$

$\mathbf{M}_u$ represents the user's assignments of $|\mathcal{T}_u|$ tags to $|\mathcal{R}_u|$ documents, which also means that only $|\mathcal{T}_u|$ rows ($\mathcal{T}_u \subseteq \mathcal{T}$) and $|\mathcal{R}_u|$ columns ($\mathcal{R}_u \subseteq \mathcal{R}$) in $\mathbf{M}_u$ are non-empty[9]. A column vector $\vec{b}_j$ in $\mathbf{M}_u$ represents the user's bookmark (post) of document (resource) d_j.

To compute a user's profile $UP(u)$, we multiply his tag-document matrix $\mathbf{M}_u$ with a document weight vector $\vec{\omega}_u$ with $|\mathcal{R}|$ components as follows:

[9] In general, we can assume that a user only employs a small subset $|\mathcal{T}_u| \ll |\mathcal{T}|$ of tags to annotate documents in his personomy $\mathcal{P}_u$, and that he only tags a small subset of all documents within the folksonomy, i.e. $|\mathcal{R}_u| \ll |\mathcal{R}|$. As such, the matrix $\mathbf{M}_u$ is generally sparse.

$$UP(u) := \mathbf{M}_u \cdot \vec{\omega}_u = \begin{bmatrix} c_1^* \\ \vdots \\ c_m^* \end{bmatrix}, c_i^* \in \mathbb{N}_0 \tag{6.4}$$

The user profile $UP(u)$ is thus a vector with $|\mathcal{T}|$ dimensions in the topic space. The values of its components c_i^* denote the *degree of interest* of the user in a particular topic, i.e. how important the topic is to the user as estimated from his tagging activities in the folksonomy.

We assume that frequently used tags are more interesting and relevant to a user than rarely used tags. In the study described in this thesis, we therefore define the document weight vector $\vec{\omega}_u^T := \vec{1}^T = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}$, thus assigning equal importance to all documents in the user's personomy. In this case, the components c_i^* of the user profile denote the total count of tag t_i for the user's bookmark collection. Of course, it is possible to refine the computation of the user profile, for example by adapting the weight of documents specified in $\vec{\omega}_u$ according to their quality (see Chapter 5) or the recency of their addition to the user's personomy. Table 6.1 shows an exemplary user profile derived from the folksonomy of Delicious.

User profile	
Topic	**Degree of Interest**
programming	157
software	147
python	120
research	98
photography	97
opensource	95
astronomy	89
...	...

Table 6.1: **Exemplary profile of Delicious user *Enibevoli* derived from his/her public tagging activities in the Delicious folksonomy.** Here, only the Top 7 interests of the user are shown (rest omitted).

Constructing user profiles as described above implies that they can be updated incrementally whenever a user adds a new bookmark to his collection, or modifies or deletes an existing one. This means that our personalization technique can adapt to shifts of user interests over time. Additionally, it allows for a more efficient computation of user profiles compared to techniques that require full rebuilds of profiles on changes [MCM+09b].

Document Profiles

We construct profiles of Web documents similar to user profiles as described in the previous section. In contrast to user profiles, which are derived from a user's individual personomy $\mathcal{P}_u$, document profiles are the result of the collective tagging activities of all users in a folksonomy $\mathcal{F}$. Whenever a user u_i creates or modifies a bookmark of a Web document, this information is shared with the community, and the document's profile is updated accordingly.

In the first step, we filter the folksonomy $\mathcal{F}$ for any information about a particular document d by restricting $\mathcal{F}$ to d and derive a document-specific tag-user matrix $\mathbf{M}_d$ of size $|\mathcal{T}| \times |\mathcal{U}|$ in the topic space:

$$\mathbf{M}_d = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix}, c_{i,j} \in \{0,1\} \tag{6.5}$$

where the components $c_{i,j}$ are initialized as

$$c_{i,j} := \begin{cases} 1, & \text{if user } u_j \text{ assigned tag } t_i \text{ to the document } d \\ 0, & \text{else} \end{cases} \tag{6.6}$$

$\mathbf{M}_d$ represents the assignments of $|\mathcal{T}_d|$ tags to the document by $|\mathcal{U}_d|$ users[10]. This also means that only $|\mathcal{T}_d|$ rows ($\mathcal{T}_d \subseteq \mathcal{T}$) and $|\mathcal{U}_d|$ columns ($\mathcal{U}_d \subseteq \mathcal{U}$) in $\mathbf{M}_u$ are non-empty[11]. A column vector $\vec{b}_j$ in $\mathbf{M}_d$ represents a bookmark (post) of the document by user u_j.

To compute a document's profile $DP(d)$, we multiply its tag-user matrix $\mathbf{M}_d$ with a user weight vector $\vec{\omega}_d$ with $|\mathcal{U}|$ components as follows:

$$DP(d) := \mathbf{M}_d \cdot \vec{\omega}_d = \begin{bmatrix} c_1^* \\ \vdots \\ c_m^* \end{bmatrix}, c_i^* \in \mathbb{N}_0 \tag{6.7}$$

The document profile $DP(d)$ is thus a vector with $|\mathcal{T}|$ dimensions in the topic space. The values of its components c_i^* denote the *degree of topical aboutness* of the document, i.e. to which extent the document is about a particular topic as estimated from the collective tagging activities of users in the folksonomy.

We assume that frequently assigned tags indicate a stronger relation of the corresponding topic to the document than rarely used tags. In the study described in this

[10] The set of unique tags assigned to a document d is defined as $\mathcal{T}_d := \{ t \in \mathcal{T} \mid (u, \mathbf{t}, \mathbf{d}) \in \mathcal{Y} \}$. Likewise, the set of users who tagged a document d is defined as $\mathcal{U}_d := \{ u \in \mathcal{U} \mid (\mathbf{u}, t, \mathbf{d}) \in \mathcal{Y} \}$.

[11] In general, we can assume that a document is only annotated with a small subset $|\mathcal{T}_d| \ll |\mathcal{T}|$ of tags, and that only a small subset of all users in the folksonomy has tagged the document, i.e. $|\mathcal{U}_d| \ll |\mathcal{U}|$. As such, the matrix $\mathbf{M}_d$ is generally sparse as well.

thesis, we therefore define the user weight vector $\vec{\omega}_d^T := \vec{1}^T = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}$, thus assigning equal importance to all users in the folksonomy who have tagged the document. In this case, the components c_i^* of the document profile denote the number of times the document was annotated with a particular tag t_i by users in the folksonomy. Of course, it is possible to refine the computation of the document profile, for example by adapting the weight of users specified in $\vec{\omega}_d$ according to their expertise (see Chapter 5). Table 6.2 shows an exemplary document profile derived from the folksonomy of Delicious.

Document profile	
Topic	**Degree of Topical Aboutness**
python	704
hadoop	629
mapreduce	528
programming	266
distributed	220
tutorial	165
cluster	158
...	...

Table 6.2: **Exemplary profile of a Web document derived from the collective tagging activities of users in the Delicious folksonomy.** The Web document in question is a tutorial by the author of this thesis about writing an Hadoop MapReduce application in the Python programming language [Nol07c]. Here, only the Top 7 topics of the document are shown (rest omitted).

6.3.3 Profile Similarity

In this section, we describe how to determine the similarity of user and document profiles.

Our notion of user-document similarity is closely related to the measure of cosine similarity. However, we deviate from the original definition of cosine similarity to account for some specific characteristics of folksonomies. We have seen in Section 2.4.4 that tag distributions in folksonomies exhibit power laws. Particularly, it has been found that resources are annotated with a large number of tags that are only used once or twice. These rarely used tags form the long tail of tag distributions. Closely related, the power-law behavior of tag distributions also implicates that a large number of users agree on a small set of tags, which means that users collectively arrive at a consensus on which tags are *the most important* to describe a given resource. In other words, the emergent consensus in folksonomies with regard to tags is mainly referring to those tags that have managed to "escape" the long tail of tag distributions for resources. We have also seen seen in Chapter 4 that even simple techniques such as thresholding may be effective for separating "signal from noise" when leveraging folksonomies for Web information retrieval.

For these reasons, we integrate a filtering function **F** into our user-document similarity measure. The purpose of this filtering function is to remove any tag noise from document profiles (by assigning the respective components a value of 0), i.e. tags that have been assigned only once or twice to the document, thereby discarding most of the long tail of tags from the analysis. Additionally, the filtering function also flattens the remaining non-zero dimensions of document profiles (by assigning the respective components a value of 1) so that a filtered document profile is, basically, a binary representation of the document with regard to the topic space. The idea is to leverage community-derived topical information mainly for identifying commonly agreed topical information of documents, and let the (unfiltered) profile of the particular user be the key factor for the personalization of his search results.

In our case, the profile similarity of a user u and a document d is a dimensionless score that is used for the *relative* weighting and re-ranking of documents within a given ordered list. It is defined as:

$$SIM(u,d) := UP(u) \cdot \mathbf{F}(DP(d)) \tag{6.8}$$

where **F** is a filtering function that updates the document profile $DP(d)$ as follows:

$$c_i^* := \begin{cases} 1, & \text{if } c_i^* > 2 \\ 0, & \text{else} \end{cases} \tag{6.9}$$

For example, Equation 6.10 illustrates the similarity computation of the exemplary user and Web document in Tables 6.1 and 6.2, respectively, which yields a similarity score of 458. As we have mentioned above, the score 458 by itself is not very meaningful – its use lies in the relative comparison with other computed similarity scores.

$$SIM(u,d) = \begin{bmatrix} 157 \\ 147 \\ 120 \\ 98 \\ 97 \\ 95 \\ 89 \\ \vdots \end{bmatrix} \cdot \mathbf{F}(\begin{bmatrix} 14 \\ 2 \\ 36 \\ 1 \\ 0 \\ 3 \\ 0 \\ \vdots \end{bmatrix}) \Leftarrow \begin{matrix} programming \\ software \\ python \\ research \\ photography \\ opensource \\ astronomy \\ \vdots \end{matrix} = \begin{bmatrix} 157 \\ 147 \\ 120 \\ 98 \\ 97 \\ 95 \\ 89 \\ \vdots \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix} = 458 \tag{6.10}$$

The profile similarity defined in Equation 6.8 favors documents with tags (topics) that are frequently applied by the user himself and, in combination with the personalization algorithm described in the next section, tends to promote known, similar documents and to demote non-similar or unknown documents. *Known* in this case refers to documents that are already in the folksonomy $\mathcal{F}$, i.e. documents that have been tagged already by some users. Thus, an important factor for the viability of our personalization approach in practice is the availability of folksonomy data for Web documents, which we investigate in Section 6.5.1.

Algorithm 2 *Personalize_List*(u, L)

Input: The user u
Input: A list L of documents (here: search results)
Output: The personalized list L^* of documents for user u

```
1: Set S to an empty list for storing <document, similarity> tuples
2: Set L* to an empty list
3: for all d in L do
4:     APPEND tuple <d, SIM(u,d)> TO S
5: end for
6: # in-place stable sort, highest to lowest similarity
7: SORT S BY SIMILARITY
8: for all tuples <d, similarity> in S do
9:     APPEND d to L*
10: end for
11: return L*
```

6.3.4 Personalization Algorithm

In the previous sections, we have described the steps of collecting data about users and documents, transforming it into user profiles and document profiles, and defined a similarity measure for these profiles. We can now perform the actual personalization of Web search results by re-ranking the returned documents according to their similarity with the user submitting the query. The personalization procedure is shown in pseudocode in Algorithm 2. The reason for employing a *stable* sort mechanism in the algorithm is to maintain the original order of search results in situations where a) personalization is not possible for some or all documents, or b) two or more documents end up having equal profile similarities with the user. In such cases, the relative order of these documents is preserved.

The left side of Table 6.3 presents an exemplary search query on Google for "security" by one of our test users (cf. Section 6.5.2), who showed a strong interest in information technology and network security in his user profile. After we performed the personalization, the result list looked as shown on the right side of Table 6.3. In general, Web sites related to IT security were promoted to the top, while sites such as the White House's information page about Homeland Security (whitehouse.gov/homeland) were demoted to the bottom. In this example, the algorithm confirmed the top-ranked search result of SecurityFocus for this user, which is a well-known and popular Web site in the IT security community. The homepage of CERT (cert.org), a center of Internet security expertise, was pushed from position 9 to 2. The US Department of Homeland Security (dhs.gov) lost six positions and was ranked at the bottom of the list. One of the Microsoft Web documents (microsoft.com/security/) was demoted because it gives a only a very high-level overview of IT security compared to the higher-ranked documents.

#	URL		#	△#	URL
1	securityfocus.com/		1	•	securityfocus.com/
2	microsoft.com/security/		2	⇑ +7	cert.org/
3	microsoft.com/technet/security/def...		3	•	microsoft.com/technet/security/def...
4	dhs.gov/		4	⇑ +4	w3.org/Security/
5	whitehouse.gov/homeland/		5	⇑ +2	ssa.gov/
6	windowsitpro.com/WindowsSecurity/		6	⇑ +4	nsa.gov/
7	ssa.gov/		7	↓ −5	microsoft.com/security/
8	w3.org/Security/		8	↓ −2	windowsitpro.com/WindowsSecurity/
9	cert.org/		9	↓ −4	whitehouse.gov/homeland/
10	nsa.gov/		10	↓ −6	dhs.gov/

Table 6.3: **Google search results for "`security`" before (left) and after (right) personalization.** For readability, the URL schemes and "www." prefixes have been omitted, and long URLs have been truncated.

6.3.5 Personalization Workflow and Implementation

General Workflow

On a technical level, our implementation of the proposed personalization method is a client-side approach, i.e. the personalization is performed by a software application on the user's computing device. This application communicates with two external Web services, namely a Web search engine (for search results) and a collaborative tagging system (for folksonomy data), to retrieve the input data required for the personalization algorithm. The general process of Web search personalization works as follows:

1. A user u makes a query on a Web search engine of his choice.
2. A ranked list of Web documents L is returned by the search engine as the result of this query.
3. For each result document $d_i \in L$, the corresponding tagging data $\mathcal{F}_{d_i}$[12] is retrieved from the collaborative tagging system.
4. The user's personomy $\mathcal{P}_u$ is retrieved from the collaborative tagging system.
5. For each document $d_i \in L$, the document profile $DP(d_i)$ is computed.
6. The user profile $UP(u)$ is computed.
7. The list of documents L is re-ranked based on the similarity $SIM(u, d_i)$ of these profiles.

In practice, this 7-step process can be computationally expensive because the volume of data transferred in steps 3 and 4 can be rather large for very popular documents

[12] $\mathcal{F}_{d_i}$ represents the restriction of $\mathcal{F}$ to d_i.

(i.e. tagged by a lot of users) and for very active users with large personomies. In such cases, the time required to transfer such data from the collaborative tagging system to the client-side application (i.e. the network latency) may increase to such a level that it might negatively impact the user experience and usability of the personalization. In other words, this general process might take longer than desired.

Optimized Workflow

A lot of existing collaborative tagging systems such as Delicious provide programmatic access to aggregated information about documents and users, and also the possibility to retrieve tagging data from multiple documents at once. Specifically, they allow for convenient retrieval of a) a user's tagging vocabulary $\mathcal{T}_u$ including the number of times these tags have been selected by the user to annotate documents with (effectively, the user's profile $UP(u)$ as defined in Equation 6.4), and b) the tags including frequency counts with which the community of users in the folksonomy have annotated a particular document with (effectively, the document's profile $DP(d)$ as defined in Equation 6.7)[13].

The general process can thus be optimized in practice so that the problematic cases described above are not affecting the technical performance of the personalization anymore. The optimized process works as follows:

1. A user u makes a query on a Web search engine of his choice.
2. A ranked list of Web documents L is returned by the search engine as result of the query.
3. For each result document $d_i \in L$, the corresponding document profile $DP(d_i)$ is retrieved from the collaborative tagging system.
4. The user profile $UP(u)$ is retrieved from the collaborative tagging system.
5. The list of documents L is re-ranked based on the similarity $SIM(u, d_i)$ of these profiles.

Here, only the profile similarities (including filtering of the document profiles by applying **F** to them) and the re-ranking of documents have to be computed on the client-side. Because the user profile $UP(u)$ can also be cached and updated locally by the client application, it is possible to save yet another step – and thus one HTTP request – in practice. The communication flow of the optimized process is shown in Figure 6.3.

[13] The same data is required, for example, to create so-called *tag clouds* of a particular user, i.e. visual depictions of a user's tagging vocabulary. Tag clouds can be similarly created for documents for visualizing the tags with which the community of users in the folksonomy have annotated a particular document. Since tag clouds are a popular feature among users, most collaborative tagging systems readily compile and aggregate the required data for implementing tag clouds, and often also provide programmatic access to this data through their application programming interfaces (API) and feeds (e.g. RSS, JSON).

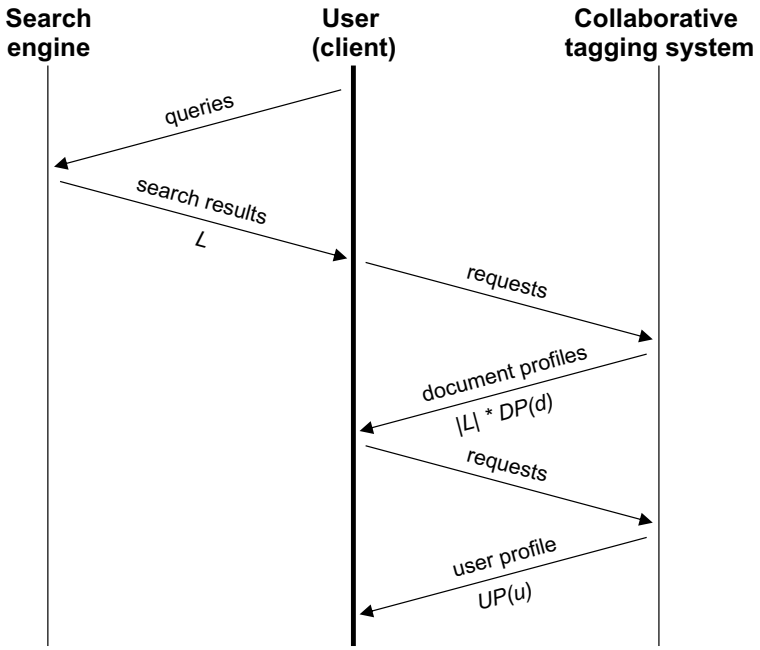


Figure 6.3: **Optimized workflow for Web search personalization.**

Transparent Personalization

In our prototype, the client-side application is a Web browser add-on (plugin), which we designed and implemented for Mozilla Firefox[14]. This browser add-on can detect when a user visits a search engine such as Yahoo! or Google, and hooks itself into the Web search process (cf. Section 6.1.1). Whenever the user submits a search query, the add-on personalizes the returned search results according to the user's topics of interests, particularly by carrying out steps 3-5 of the optimized personalization workflow as described in the previous section.

On the user interface level, the personalization of our prototype is completely transparent to the user and happens instantly even though extra communication with the collaborative tagging system is required: The browser add-on analyzes the HTML code of search result pages on the fly, and re-ranks the search results by modifying the DOM tree[15] of these pages in real-time on the user's computing device (see illustration in Figure 6.4). While the technical implementation of DOM tree manipulation is specific

[14]Mozilla Firefox Web browser, http://www.mozilla.com/firefox/, last retrieved on March 01, 2010.

[15]The *Document Object Model (DOM)* is a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of Web documents. A document can be further processed and the results of that processing can be incorporated back into the presented page. A description of DOM provided by the World Wide Web Consortium (W3C) is available at http://www.w3.org/DOM/, last retrieved on March 01, 2010.

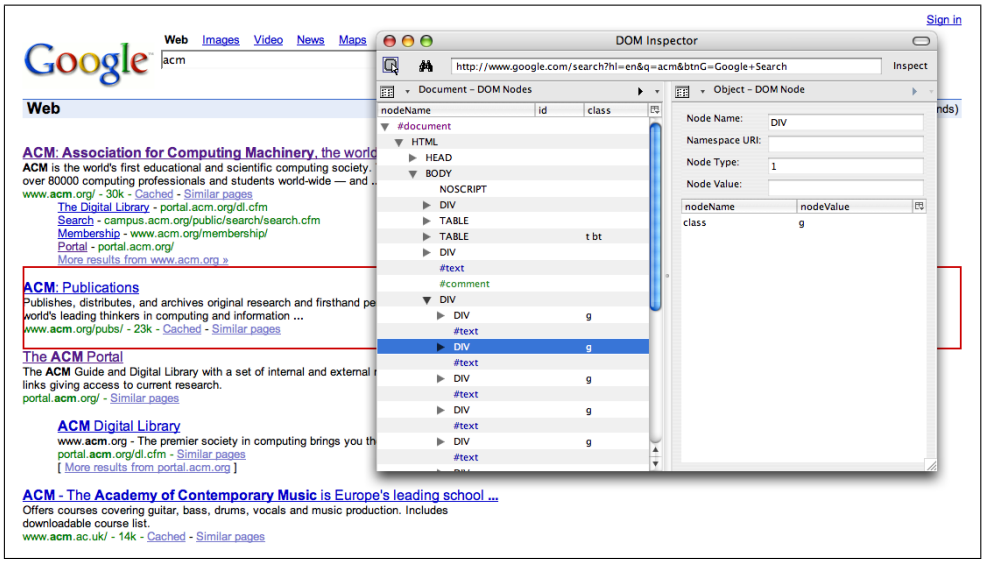


Figure 6.4: **DOM tree of search result pages on Google.** On the user interface level, the personalization in our prototype is completely transparent to the user: Our browser add-on analyzes the HTML code of the search result pages of Web search engines, and updates the search results in real-time on the user's computing device. In the screenshot on the left side, for example, the second search result *"ACM: Publications"* is contained in the `DIV` element that is highlighted in the DOM tree view on the right side (here: DOM Inspector of the Firefox browser).

to a particular search engine[16], the proposed personalization method is conceptually independent of the search engine being used.

Additionally, the browser add-on enhances the browser GUI with a "Tagmarking" button as described in Section 6.3.1. It also highlights such Web documents in search results that have already been tagged by the user for easier visual identification of documents known to the user[17]. As illustration, Figure 6.5 shows the re-ranked search result list described in Table 6.3 at the example of the Google search engine.

[16] DOM tree manipulation is specific to a search engine because the search engine user interfaces, e.g. the design and structure of search result pages, and corresponding HTML code vary between search engines.

[17] We are currently working on a feature that will retrieve potentially relevant Web documents from a user's personomy $\mathcal{P}_u$ based on the entered search query. Such a feature would allow us to present Web documents to the user that are not in the search engine's index and thus cannot be returned through a Web search, e.g. a non-public intranet Web document.

Figure 6.5: **Personalized search results on Google.** Here, the re-ranked search result list described in Table 6.3 is shown. "As you can see, you can see nothing" – with the exception of the green star icon, which denotes that the user already tagged the homepage of Security Focus in the past, the personalized search result list looks exactly like Google's original.

6.4 Experimental Setup

We evaluate the performance of the proposed personalization method with a quantitative analysis and a qualitative user study.

Firstly, the *practical feasibility* of the personalization method depends on the availability of folksonomy data about Web documents in search results. The more documents are expected to be covered by the tagging activities of folksonomy users, the more data may be available for constructing document profiles and, subsequently, for allowing proper personalization of Web search according to our folksonomy-driven approach. On the opposite end, we cannot re-rank documents that haven't been tagged by users, which means that such documents would keep their relative positions according to the initial ranking of the search engine (but would be ranked below any documents for which the profile similarity with the user is greater than zero). In Chapter 4, we have already investigated the availability of the folksonomy data in the general context of Web information retrieval. We found that folksonomies do indeed provide large volumes of data about Web documents and already cover a considerable fraction of the Web. For the specific scenario of Web search personalization, we further extend and augment these studies with an analysis of folksonomy data for Web search results. Another finding described in Chapter 4 is the correlation between folksonomy data and a document's popularity on the Web (indicated by its Google PageRank): the more popular a document, the more likely it is tagged by users. We can thus infer information

about the availability of folksonomy data about search result documents by analyzing their PageRank distributions. Hence, we have combined our previous experimental results with an analysis of the AOL500k corpus (see Section 3.1.4). We randomly sampled ∼1,750,000 queries with 1,000,000 clicked search results from AOL500k, and subsequently retrieved PageRank information for each clicked Web document from Google. This experimental data set allows us to gain insights into the PageRank distributions of Web search results, and thereby also to estimate the availability of folksonomy data for Web search personalization.

Secondly, we evaluate the *quality* of the personalization method by comparing its rankings with the evaluation baseline of the original, non-personalized results of a search engine. While involving a human in an evaluation process is rather cumbersome and expensive, previous studies such as the work of Najork et al. [NZT07] have shown that human judgments are crucial for the evaluation of search engines because no document features have been found yet that can effectively estimate the relevance of a document to a user query. For this reason, it is required to ask a human to evaluate the outcomes of our personalization method in order to compare the quality of (re-)ranked search results. Hence, we conducted a user study with $N = 8$ participants that followed the experimental methodology of search engine evaluation by Haveliwala [Hav02][18].

With regard to the human participants, all our testers were computer literate, familiar with Web search and users of Delicious. Their job functions included researchers, system administrators, webmasters and software developers. With regard to the technical setup of the experiment, we selected Google as the search engine and Delicious as the collaborative tagging system (cf. Section 6.3.5). We constructed the user profiles of our participants from their personomies $\mathcal{P}_u$[19] on Delicious; document profiles were created similarly. The actual user study was conducted as follows: For each participant, we randomly built a set of 13 search queries from her or his search history and tagging activities, thus totaling 104 queries. For each search query, a participant was presented two result rankings of $|L| = 10$ Web documents, i.e. the first result page of the query: 1) the original, non-personalized list from Google Search, and 2) the personalized version according to our proposed approach. The experiment was conducted as a *blind test*, i.e. the result lists themselves were presented in random order so as not to bias the participants. Similarly, they were not told anything about how either of the rankings was generated. Participants were then asked to determine which of the two results lists of a query was "better" overall, in their opinion. Here, *better* was defined as ranking highly relevant results at the top of the list and ranking irrelevant results at the bottom, i.e. promoting "good" results and demoting "bad" results. Participants could also vote for a draw if they could not decide which list was better.

[18] It should be noted that Xu et al. [XBF+08] propose an evaluation framework for personalized search using folksonomies that tries to circumvent the need for direct user studies. However, their framework bootstraps a ground truth for evaluation by assuming a strong correlation between the tags and search query keywords. While we have seen in Chapter 4 that there is a correlation between tags and search keywords, we argue that the correlation is not strong enough to rely on it for evaluation purposes in the context of our work described in this chapter.

[19] The average number of posts (bookmarks) in a participant's personomy $\mathcal{P}_u$ in our study was 153.

6.5 Experimental Results

We start our discussion of the experimental results with the quantitative study of folksonomy data in the scenario of Web search, and continue with the results of our qualitative user study.

6.5.1 Quantitative Analysis

Firstly, we looked at Web documents and analyzed the average PageRank of *displayed* – but not yet clicked – documents for each search result position. We have seen that the volume of folksonomy data about a document increases with its popularity on the Web. As such, the PageRank distribution of search results is an indication of the availability of tagging data in practice.

We observed that the Top 10 positions had an average PageRank of 5.2 or higher as shown by the black line in Figure 6.6. The dashed red line denotes the click frequency of users per search result position. The first five positions accounted for 73.7% of all clicked search results, most of which was contributed by the Top 1 position. The drop between positions 10 and 11 is most probably caused by the default configuration of AOL/Google search to show only $|L| = 10$ Web documents per search result page (a setting similar to other popular search engines such as Yahoo! Search and Microsoft Bing), which means that users are very unlikely to look for search results beyond the first page. These results are encouraging for our personalization method. On the one hand, they indicate that proper re-ranking is very relevant and useful in practice because it is important to promote the best documents according to a user's personal preferences to the prominent positions at the very top of result lists. On the other hand, Web documents in search results are likely to be tagged due to the expected high PageRank. For example, we have seen in Section 4.3.1 that about 73.1% of Web documents with a PageRank of 5 were tagged in our experimental data set CABS120k08. In this context, we also observed that the difference in data sampling between CABS120k08 and DMOZ100k06 data sets had a visible effect in the scenario of Web search: The estimated probabilities were much higher for CABS120k08 than for DMOZ100k06. Because the Web documents contained in CABS12k08 are derived from an intersection of AOL500k with the Open Directory Project (cf. Section 3.2.2), we argue that it provides better estimates of the true probabilities of Web documents being tagged in search results than the DMOZ100k06 data set.

Secondly, we looked at users and averaged the PageRank of *clicked* search results for each user in the experimental data, i.e. we derived individual click preferences for Web documents regardless of their position in the search results. This allows us to investigate the PageRank distributions of Web documents in search results without a potential bias due to the positioning in the result lists, and also account for variations of individual user click behavior to a certain extent.

We observed that 80.1% of users had an average clicked PageRank of 5 or higher, and 32.9% had a PageRank of 6 or higher. The details are shown in Figure 6.7, where the black line denotes the percentage of users with an average clicked PageRank of x

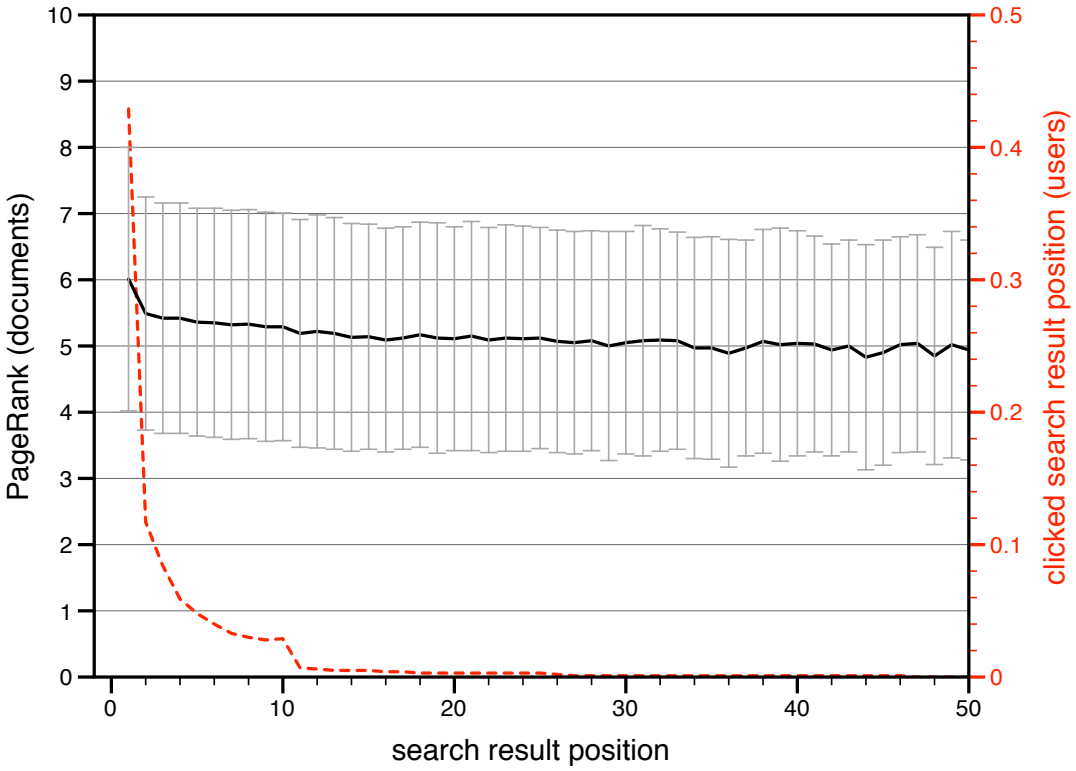


Figure 6.6: **PageRank distribution of displayed search results by position.** Mean PageRank of Web documents (black) including standard deviations (gray bars) per search result position. The click frequency of users is shown by the dashed red line.

or higher. The solid and dashed diamond lines show the estimated probabilities of a document to be tagged in our data sets CABS120k08 and DMOZ100k06, respectively, based on our findings in Chapter 4. Hence, for most clicked Web documents in search results, the probabilities of having been tagged are quite high – in CABS120k08, for example, 73.1% and 87.0% for $PR5$ and $PR6$ documents, respectively. These experimental results therefore suggest that most users would indeed benefit from the proposed personalization approach in practice.

While these observations are promising already, it should also be noted that for being effective the personalization method does not require every single document being tagged. It is acceptable if some $k < |L|$ documents in search results have not been tagged yet, because the remaining $|L| - k$ documents may still allow for reasonable personalization quality as we will in the next section. Additionally, the usage of collaborative tagging systems such as Delicious is still increasing in the Web today [CM08], and thus the availability of folksonomy data about Web documents will further increase over time as well.

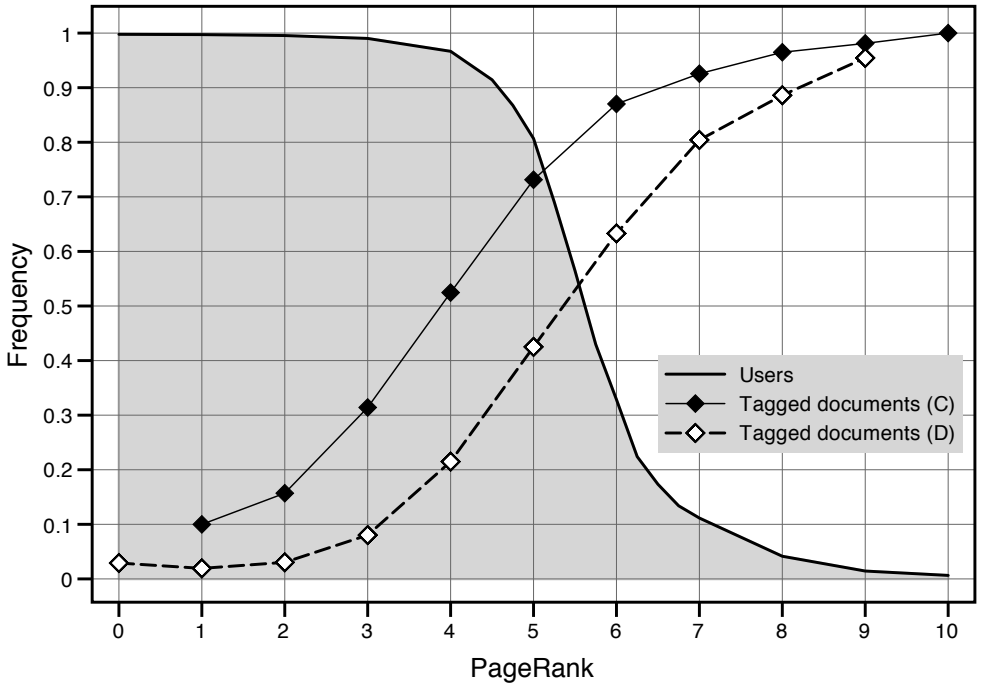


Figure 6.7: **PageRank distribution of clicked search results regardless of position.** The frequency of users with an average PageRank for clicked search results equal to or higher than x, i.e. $P_{user}(PageRank \geq x)$, is shown by the black line enclosing the gray plot area. The solid and dashed diamond lines denote the frequencies of tagged documents for a particular PageRank x in the experimental data sets CABS120k08 and DMOZ100k06, respectively, i.e. $P_{doc}(tagged|PageRank = x)$.

In a last experiment, we randomly sampled 140 "seed" tags ($\mathcal{T}_0$) from the so-called "popular tags" reported by Delicious[20]. We subsequently run search queries for each seed tag on Google, yielding a total of $|\mathcal{T}_0| * |L| = 1,400$ search result documents ($\mathcal{R}_{query}$). For each document, we retrieved the document's most popular tags from Delicious, thereby discarding tag noise from our analysis. Hence, we effectively performed a similar procedure as carrying out steps 1-3 of the optimized personalization workflow described in Section 6.3.5. The final data set consisted of a total of 981,989 user bookmarks ($P(\mathcal{F}_{query})$) with 20,498 assignments ($\mathcal{Y}_{query}$) of 2,300 tags ($\mathcal{T}_{query}$). The details are shown in Table 6.4 and Figure 6.8. We observed that about 9 out of 10 search results were bookmarked and 8.5 out of 10 search results were tagged. Again, this is a promising outcome with regard to the practical feasibility of our personalization approach. Our findings indicate that the majority of search results – at least for popular tags/queries – can be personalized in practice.

[20]Delicious Popular Tags: http://delicious.com/tag/.

Pos	Bookmarks	Tag Assignments		Pos	Bookmarks	Tag assignments
1	1450	19.8		6	456	13.7
2	627	16.4		7	495	13.4
3	1199	15.5		8	574	13.7
4	451	14.2		9	404	14.0
5	610	12.5		10	784	13.3

Table 6.4: **Mean number of bookmarks and tag assignments (here: only with regard to popular tags) of Web documents per search result position.** The peak of 784 for position 10 was caused by two extreme data points in our sample; it drops to 519 when these two data points are removed.

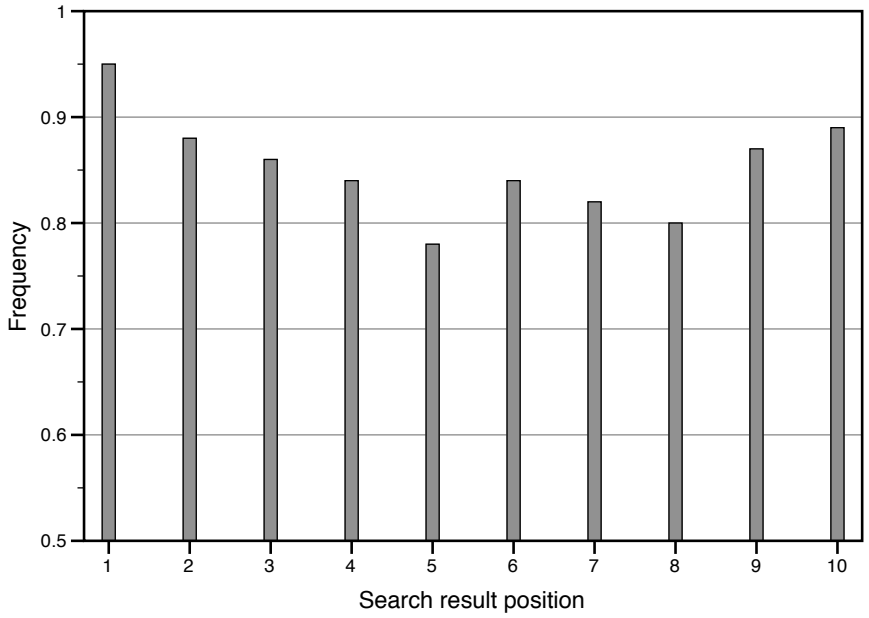


Figure 6.8: **Percentage of Web documents per search result position that had at least one associated popular tag.**

6.5.2 Qualitative User Study

In our user study, the participants considered the personalized list to be better than the original result list in 63.5% of the queries as illustrated in Figure 6.9. The evaluation baseline, i.e. the unmodified result list as returned by the Google search engine, was preferred in 29.8% of the queries. An interesting observation was the low frequency (6.7%) of the cases where users could not prefer one list over the other.

Previous studies such as [JP01] and our experiments described in Chapter 4 have shown that most search queries are rather short, with the average search query consisting of only one or two words. We found that the personalization method was particu-

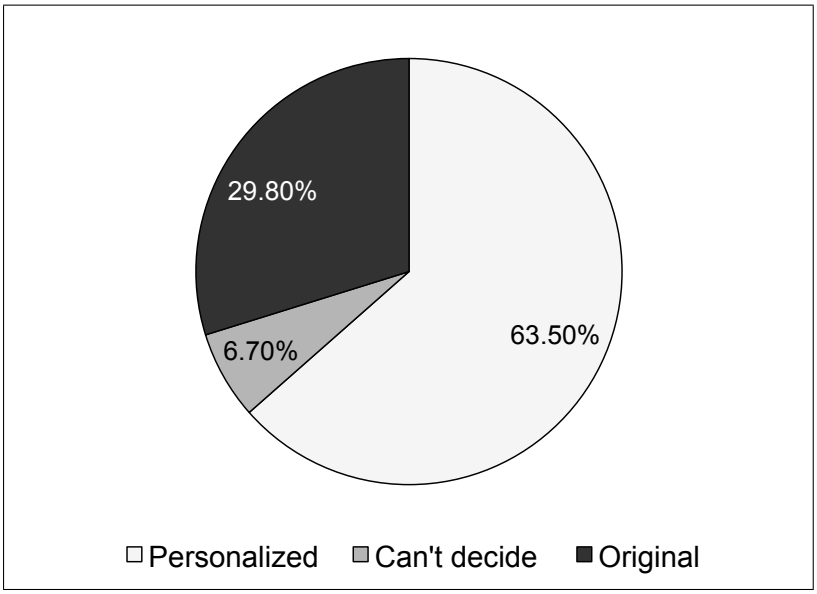


Figure 6.9: **Results of the qualitative user study.** In our study, participants preferred the personalized list in 63.5% of the queries. The original, non-personalized ranking of Google was deemed better in 29.8% of queries. For 6.7% of queries, users could not prefer one list over the other.

larly helpful for disambiguation of words and contexts for queries such as `jaguar` or `golf` (see also the `security` example in Section 6.3.4). It also showed its strength for queries that used abbreviations or acronyms such as `sf` (e.g. "Science-Fiction" or "San Francisco"). As such, these results indicate that the proposed personalization approach is especially helpful for short queries by properly understanding and matching the topics of documents with the individual topical preferences of users. On the other hand, we found that the personalization method had difficulties to improve search results for those users who were only broadly interested (or knowledgeable) in a particular topic, and thus were not providing a sufficient level of detail with regard to tags. For instance, a user with a large number of bookmarks that are tagged just with `design` will not benefit as much from this personalization approach as a user who tags his bookmarks about design more granularly. This means that the more a user is able and willing to provide adequate tag assignments for Web documents, the better we can understand his topics of interests based on an analysis of his personomy $\mathcal{P}_u$ and, as a consequence, the better will be the performance of Web search personalization. Lastly, another finding was that the personalization helped to demote spam documents, e.g. "fake" reviews of products, from the top ranks in search results because these documents where unlikely to be tagged by legitimate users[21].

[21] While we have seen in Chapter 5 that promoter-type spammers in *folksonomies* are a common phenomenon nowadays, they do not seem (yet) to simultaneously impact Web search on a larger scale. A possible explanation is that spammers have not yet focused on concentrated, parallel attacks on both

It should be noted that the quality of personalization depends in our case on the initial quality of search results as returned by a search engine. Since we have used the search engine of Google in our experiments, we can expect that this initial quality is very high, representing the state of the art in terms of Web search. It could thus be argued that users in our study would be satisfied with search results regardless of how Web documents are re-ranked. However, the research studies and our own analyses described in the previous sections have shown that user behavior shows a strong bias towards the positioning of Web documents in the result lists. This means that variations in search rankings will also result in different perceptions and judgements of users with regard to these rankings. Additionally, the difference in users' preferences for personalized and non-personalized search rankings was quite large in our user study. We therefore argue that the personalization approach is indeed able to augment and improve traditional Web search for the benefit of users.

6.6 Discussion

Our experiments described in the previous sections have shown promising results for our proposed personalization approach, and the participants of our user study found that it produced better rankings than the evaluation baseline. Compared to other techniques to augment Web search based on folksonomy data, our approach is not relying on the similarity of collaborative tagging (tags) and search queries (keywords).

We can summarize the further benefits of the personalization approach as follows. Firstly, tagging a Web document will improve future Web searches even if the user is not actively using a search engine. For example, when a user tags a Web document recommended to him via email, it will still affect his user profile. Secondly, the approach can personalize the search results from different search engines. Because the user profile is not managed by a single search engine, he can leverage his tagging activities to personalize multiple (even competing) search engines with the same user data. The approach can even personalize the search results of a Web search engine that natively does not support personalization itself. Thirdly, collaborative tagging and social bookmarking can reach areas of the Web that are inaccessible for search engines (e.g. intranets, access-restricted Web sites or newly created, unlinked Web documents). Hence, it is also possible to collect data from the activities of users tagging these documents. Fourthly, it is comparatively easy to explain users why a Web document has been promoted or demoted during personalization (e.g., "...because you have annotated a lot of Web documents with the tag `foobar`..."), if such information is desirable. Lastly, the computational expense of the client-side personalization process is very low, particularly since we only need to compute $|L|$ profile similarities. As such, the proposed personalization approach can easily be performed on client devices with limited energy or processing power such as mobile phones, and is thus not affected by the drawbacks of similar personalization techniques as described by Jeh and Widom [JW03].

collaborative tagging systems and search engines, i.e. trying to boost the prominence of their content in both domains at the same time.

A commonly used argument against the effectiveness of client-side re-ranking techniques is that they are limited to the re-ranking of the Top $|L|$ search results. In other words, such personalization approaches depend on the quality of the initial list of $|L|$ Web documents returned by a search engine – if, for example, these documents are not sufficiently relevant to the query, the user will not benefit much from further re-ranking of the documents. However, we have noted already that while we have focused our experiments on the Top 10 search results for reasons explained in Section 6.3, our approach is neither conceptually nor technically limited to a specific number or upper limit of search results per query. Recent developments such as the launch of services like *Yahoo! BOSS (Build your Own Search Service)*[22], which gives users free access to the search index of Yahoo!, allows us to retrieve a much larger number[23] of search results per query. Additionally, client-side techniques also help to balance user privacy and search quality [Hav02, XWZC07] because users don't need to identify themselves to search engines to benefit from personalized search.

Although we have focused our discussion on the context of Web search, the personalization approach proposed in this chapter is not limited to this scenario. The personalization algorithm described in Section 6.3.4 accepts as input any list of Web documents, of which a list of search results is but one example. We can thus leverage the approach also in other areas of Web information retrieval. In [Nol09], for example, we have demonstrated how the idea of personalization described in this chapter can be used for creating individual recommendations of Web documents from news feeds.

Possible Improvements

While we have seen that the proposed personalization approach shows good performance for re-ranking Web documents according to their similarity with a user's topics of interests, we have identified some opportunities for improvement.

Firstly, understanding the user's topics of interests may be further improved. It has been found that identifying short-term and long-term trends in a user's interests is beneficial for personalization purposes [SHY04, DSW07]. Hence, deriving such trends, for example from temporal information about users' tagging activities, could thus help to produce an even better ranking of Web documents in search results. Similarly, tags that are related to those in a user's personomy $\mathcal{P}_u$ could be integrated into the personalization approach for improved construction of topical information about users and Web documents [MC07]. Wetzker et al. [WZBA10] propose a user-centric tag model that derives mappings between a user's personal tagging vocabularies $\mathcal{T}_u$ and the global folksonomy $\mathcal{F}$, which helps find similarities between users (and documents) even when their tagging vocabularies do not overlap. Integrating such techniques could thus further improve the representations of users and documents in the topic space.

Secondly, it could be desirable to integrate additional documents into Web search results from a user's personomy or from the folksonomy at large, i.e. documents that

[22] Yahoo! BOSS, http://developer.yahoo.com/search/boss/.
[23] In the case of Yahoo! BOSS, a maximum number of 1,000 search results may be retrieved per query.

are not present in the initial search result list L.[24] In this case, data from folksonomies could be used not only for re-ranking a given list of documents extracted from a search engine's index, but also for injecting documents directly from collaborative tagging systems. For example, prior work has shown that about 40% of search queries are related to re-finding already visited Web documents [TDH07]. Embedding documents relevant to a particular search query from a user's personomy, i.e. documents that he has already read and perceived as useful in the past, might therefore further improve the quality of Web search. Similarly, high quality documents that match a user's query could be retrieved from folksonomies through techniques such as SPEAR (see Chapter 5) that measure the quality and popularity of Web resources within folksonomies (in contrast to an analysis of the Web graph as is done by search engines).

6.7 Summary

In this chapter, we have presented a new approach to personalization of Web search by exploiting folksonomies for deriving topical information about users and Web resources. We have demonstrated how the approach can be implemented in practice at the example of the search engine Google and the collaborative tagging system Delicious. Our experiments have shown that the approach is effective at extracting information from folksonomies in order to tailor search results by re-ranking Web documents according to a user's individual topics of interests. We have also shown that the approach is feasible in practice with regard to the availability of sufficient volumes of folksonomy data about Web documents in the scenario of Web search, and that users have perceived an improvement in the quality of search results compared to the evaluation baseline. Hence, our results support our hypothesis that folksonomies provide sufficiently rich information about users and Web resources to allow for the personalization of Web search.

In the next chapter, we will explore how the concepts of collaborative tagging and folksonomies can be exploited for Web filtering. We will present a case study of a working prototype, *TaggyBear*, and describe and evaluate its system design and anatomy.

[24] The browser add-on that we have developed already highlights existing documents in L that have been tagged by the user submitting the query.

You do ill if you praise, but worse if you censure, what you do not understand.

Leonardo da Vinci (1452–1519)

7 Web Filtering

We have seen in the previous chapters that the convenience and popularity of collaborative tagging and folksonomies have resulted in large volumes of metadata about Web resources, particularly with regard to classification and categorization of these resources. In Chapter 6, we have demonstrated how this information can be leveraged for the personalization of Web search. Web filtering, another domain of Web information retrieval, is concerned with the classification of Web resources as well. While the goal of search personalization is the *facilitation* of access to Web resources, the goal of Web filtering is the *prevention* of access to some of these resources. A typical scenario of Web filtering is the so-called "blocking" of resources that have been assigned to specific content categories such as phishing or pornography by technically preventing that such resources can be retrieved from the Web or displayed on a user's computing device. However, the public reputation of traditional approaches to Web filtering has deteriorated in recent years because they have also been used, for instance, by governments to enforce censorship on their citizens [DPRZ08]. Similarly, existing Web filtering techniques still suffer from problems such as inaccurate classification of resources or lack of acceptance among users, which hinder their effectiveness and success in practice.

Based on the scientific findings described in the previous chapters, we argue that an alternative approach to Web filtering would be to harness the concepts of collaborative tagging and folksonomies in order to create a user-driven filtering application of the Web, i.e. establishing a "democracy" on the Web with regard to content filtering, in the spirit of similar community projects such as Wikipedia[1]. In this chapter, we explore how collaborative tagging and folksonomies can be exploited to implement such an approach in practice, and present a case study of a working prototype called *TaggyBear*. We describe and evaluate its system design and anatomy, and test our hypothesis regarding collaborative tagging and folksonomies:

> **Hypothesis 4 (Web Filtering)**:
> The concepts of folksonomies and collaborative tagging can be exploited for user-driven filtering of the Web, i.e. allowing or blocking access to Web resources based on human input.

[1] Wikipedia, http://www.wikipedia.org/.

7.1 Filtering the Web

The domain of Web filtering is one aspect of the broad area of Web information retrieval. It is concerned with the classification of Web resources for the purpose of preventing the access to such resources that have been assigned to specific content categories. One usage scenario where Web filtering is employed is the protection of users from malicious, illegal or otherwise harmful content. While a comprehensive introduction to Web filtering is beyond the scope of this thesis, we provide a brief summary of its basic components and most important aspects in this section.

Firstly, a *classification scheme* for resources must be defined. Tasks include the decision on the number and type of categories as well as their structure and relations. For example, a filtering application that focuses on protecting users from phishing[2] Web documents [RW08b] could simply use a binary classification of resources into `phishing` and `not phishing`, similar to the scenario of spam filters for emails. Secondly, Web resources need to be analyzed and subsequently classified based on their content. This task, also called the *rating* of the resource, is arguably the most difficult step of Web filtering. The challenges include the correct understanding and interpretation of the content of a Web resource (including any multimedia content such as images or videos, as well as interactive content such as Adobe Flash), and the final decision to which category or categories the resource is assigned to. Thirdly, a filtering *policy* must be defined, which is primarily the selection of a subset of categories for which filtering will be enforced, i.e. the access to any resources assigned to this subset of categories is restricted. In a university network, for example, such a policy could be centrally managed and apply to all users of the university. In a home environment, a user could individually define the policy according to his personal preferences. Lastly, the actual *filtering mechanism* determines how the access restriction to Web resources is technically implemented. For example, the display of a resource on a user's computing device could be obfucsated, or the retrieval of the resource from the Web could be prevented in the first place. It should be noted that these steps may but not need to happen at the same time. For instance, the analysis and subsequent classification of resources may be performed prior to a user retrieving a resource from the Web, i.e. similar to the way search engines crawl and index the Web independently from the actions of users submitting their queries to the search engine [MRS08].

The filtering workflow is straight-forward from a user's perspective: Web filtering applications are commonly implemented as client-side or server-side software, or a combination thereof. Whenever a user attempts to requests a resource from the Web with a Web filter in place, the filter will first determine the rating of the requested resource (e.g. through direct content analysis in real-time, or a lookup of pre-computed ratings against a database). Then, depending on the defined policy, the filter will decide

[2] *Phishing* is the criminally fraudulent process of attempting to steal sensitive information such as password or credit card details from users by masquerading as a trustworthy entity in an electronic communication. For instance, a criminal could send a fake email to a user asking for his bank account credentials, or similarly set up a fake Web site that visually resembles the user's online banking service in order to trick him to enter his username and password.

whether to grant or deny access to the resource, which in the latter case is implemented through the filtering mechanism.

Traditionally, we can distinguish between two general approaches to Web filtering, which can also be combined for increased effect. They mainly differ in how the rating of resources is accomplished. In the following sections, we describe these approaches and characterize their strengths and weaknesses.

7.1.1 Filtering based on Ratings by Humans

The first approach to Web filtering relies on human judgements about Web resources, and as such resembles rating systems such as *MPAA*[3] for movies or *ESRB*[4] for computer software and games. Similar to the general Web taxonomy of the Open Directory Project (see Section 3.1.2), resources are manually rated according to a predefined classification scheme, which can range from simple binary classification to comprehensive taxonomies. These ratings can then be exploited by Web users through compatible software applications in order to grant or deny access to resources on the Web.

Rating by Web Authors

Most of the existing manual rating systems are voluntary and not legally binding. They focus on the authors and publishers of Web resources, who can use these rating systems to manually classify their content with a common description framework and add rating information in the form of special metadata to their Web resources.

Arguably, the most prominent manual rating framework is developed and maintained by the *Internet Content Rating Association (ICRA)*[5]. ICRA was established in 1999 as an independent non-profit organization by a group of international Internet companies and associations such as AOL, British Telecom, Microsoft and Verizon. It has since been supported by the European Commission's Safer Internet Programme[6] and has also participated in several EC-funded projects in the fields of Internet security. Its rating framework has initially been based on the *PICS* standard[7] of the World Wide Web Consortium [JKR+99] but the current framework also supports RDF[8]. The cornerstone of ICRA is the ICRA vocabulary[9] – i.e. its classification scheme – which defines a set of descriptors for classifying and rating online content. The vocabulary covers topics such

[3] Motion Picture Association of America, http://www.mpaa.org/, last retrieved on March 01, 2010.
[4] Entertainment Software Rating Board, http://www.esrb.org/, last retrieved on March 01, 2010.
[5] ICRA, http://www.fosi.org/icra/, last retrieved on March 01, 2010.
[6] Safer Internet Programme of the European Commission, http://ec.europa.eu/saferinternet, last retrieved on March 01, 2010.
[7] Platform for Internet Content Selection, http://www.w3.org/PICS/, last retrieved on March 01, 2010. Recently, PICS has been officially superseded by *POWDER*, the RDF-based Protocol for Web Description Resources, available at http://www.w3.org/2007/powder/. Most interestingly, the POWDER specification also includes the possibility to add "free-text tags" to resources (http://www.w3.org/TR/powder-dr/#tags) – a tribute to the success of folksonomies?
[8] *Resource Description Framework (RDF)* is a standard model for data interchange on the Web. A description is available at http://www.w3.org/RDF/, last retrieved on March 01, 2010.
[9] ICRA vocabulary, http://www.icra.org/vocabulary/, last retrieved on March 01, 2010.

as “nudity”, “violence”, “language”, and “potentially harmful activities”. A selection of ICRA descriptors is given in Table 7.1. Listing 7.1 shows an exemplary ICRA rating which is embedded as HTML metadata into a Web document, with the referenced RDF file presented in Appendix A. A popular client application that supports ICRA ratings is Microsoft’s Internet Explorer: The Web browser ships with a *Content Advisor* feature that can be configured to filter access to Web resources based on ICRA ratings.

Descriptor	Category	Meaning
na 1	Nudity	Exposed breasts
nc 1	Nudity	Visible genitals
nz 1	Nudity	No nudity
sd 1	Sexual Material	Explicit sexual language
se 1	Sexual Material	Erections/explicit sexual acts
vb 1	Violence	Injury to human beings
vi 1	Violence	Torture or killing of animals
lb 1	Language	Profanity or swearing
ca 1	User-generated content	User-generated content such as chat rooms and message boards (moderated)

Table 7.1: **Selected ICRA descriptors for rating Internet content.**

```
<link rel="meta" href="http://www.example.com/labels.rdf" type="application/
    rdf+xml" title="ICRA labels" />

<meta http-equiv="pics-Label" content='(pics-1.1 "http://www.icra.org/pics/
    vocabularyv03/" l gen true for "http://example.com" r (n 3 s 3 v 3 l 3 oa
     2 ob 2 oc 2 od 2 oe 2 of 2 og 2 oh 2 c 3) gen true for "http://www.
    example.com" r (n 3 s 3 v 3 l 3 oa 2 ob 2 oc 2 od 2 oe 2 of 2 og 2 oh 2 c
     3))' />
```

Listing 7.1: **An exemplary ICRA rating.**
In this example, the Web site `http://www.example.com/` is rated as depicting exposed breasts, bare buttocks and visible genitals. Other types of objectionable content may be, but are not known to be, present. This HTML snippet must be included in `<HEAD>` section of each page on the Web site. The referenced RDF file `labels.rdf` is shown in Appendix A.

Such rating systems for Web resources sound promising in theory. Obviously, the availability of manual ratings could make the filtering task per se rather trivial and theoretically more reliable than automated methods by machines. However, the viability and success of these rating systems depend heavily two factors: Firstly, the actual usage and resource coverage of these systems and, secondly, the accuracy and trustworthiness of rating information. In a previous research work, we conducted the – to the best of our knowledge – first empirical study of Internet content rating systems [NM05]. By creating and analyzing a data set of 152,617 Web documents, we showed that the actual usage and coverage of these systems is at best marginal – only 0.6% of the analyzed

Web documents provided *syntactically* correct labels. For estimating the accuracy and trustworthiness of ratings, i.e. their "semantic" correctness, we manually analyzed a random sample of 5,000 Web documents from the original data set and verified their ratings. We identified discrepancies between ratings and actual content in 18.5% of the cases, thus further lowering the usefulness of these ratings in a real-world scenario. Our results show that the performance of a Web filter based on such rating systems would in practice only be slightly better than the two extremes of a) not using a filter at all, or b) unconditionally blocking access to any resource on the Web.

Rating by Third Parties

An alternative to manual rating systems as previously described are services offered by third parties. Internet service providers (ISP), for instance, often provide Web filtering services for their subscribers, and IT security vendors offer commercial whitelists and blacklists of Web resources to their customers. Examples of such third-party services are *Parental Controls* by AOL (US)[10] and *Child Protection Software* by T-Online (Germany)[11].

If public information about the actual implementation of these services is available, the documentation often refers to the use of manual classification approaches or dedicated teams of human experts for classifying and rating Web resources. However, this kind of manual processing will hardly manage to cope with the rapid growth of the Internet, the increasing number of new Web documents and – not to forget – changes to existing ones. Another common drawback in practice is a lack of filter granularity. Instead of rating on a per-document basis, e.g. `http://hostingprovider.com/customer01/bad-page.html`, ratings are often just assigned to second-level domains (here: `http://hostingprovider.com/`). In such cases, a single bad apple ruins the whole basket: Instead of blocking access to a single "bad" Web resource published by one customer, the hosting provider and all its customers may be subject to filtering.

In addition to these practical problems, there are also conceptual problems with manual rating approaches: Firstly, differences in the perception and interpretation of con-

[10] AOL Parental Controls, `http://parentalcontrols.aol.com/`, last retrieved on March 01, 2010.

[11] "T-Home Kinderschutz Software", `http://service.t-online.de/kinderschutz-software/id_12727562/`, last retrieved on March 01, 2010. The current version of the software supports ICRA content ratings. The software documentation explains (freely translated) that all T-Online Web pages have been rated according to ICRA. Anecdotally, a quick test in December 2009 conducted by the author of this thesis revealed that this statement is, unfortunately, false. For example, the celebrity and entertainment section of T-Online at `http://www.vip-spotlight.t-online.de/` showed pictures of a so-called "sexy shooting" of US actress Lindsay Lohan. The ICRA rating – last updated four years ago in December 2005 – of the Web document specified "no nudity, no sexual material, no potentially offensive language". However, the document in fact showed an undressed Lohan lying on a bed in the arms of a male partner in what most people would describe as "Obscured or implied sexual acts" in the ICRA vocabulary. The same Web document also showed a featured video of US actress Pamela Anderson as a so-called "sexy genie" in a corresponding outfit. The video of Anderson also started with an advertisement of Miami Playboy, a new perfume by Playboy company. In short, even T-Online as a commercial provider of child protection services does or did not adhere to its own standards with regard to content rating.

tent may result in different ratings of Web resources between human raters, similar to the scenario of manual subject indexing discussed in Section 2.3.1. As described in studies such as [GSF02] and our findings in Chapter 4, people often disagree on how a Web resource "should" be rated due to a variety of factors such as differences in cultural, familial, educational and religious backgrounds. Therefore, a single rating of a Web resource will most likely not match the opinions of all involved parties, and any "one size fits all" approach will most likely fail in practice. Another issue and often highly debated point is that both rating by Web authors and rating by third parties inherently do not reflect the opinion of end users – the latter are often not involved in the filtering process and thus complain about lack of transparency and express their desire for participation [DPRZ08]. On a political level, rating and filtering "on behalf of the user" by third parties such as Internet service providers, governments or other institutions may also be perceived by end users as censorship and paternalism [OI05, DPRZ08].

7.1.2 Filtering based on Ratings by Machines

The second approach to Web filtering is based on automated analysis and rating by machines. The advantage of automated techniques is that they are better suited to cope with the large scale and rapid growth of the Web. A plethora of algorithms are available for Web filtering tasks: from simple keyword filtering of textual content, i.e. blocking a Web document if it contains certain terms, to content recognition in images [WWG05, KLC06, RJB06, LKCC07, DPN08] and videos [JUB09], and to specialized techniques and sophisticated machine learning algorithms such as Support Vector Machines or neural networks [NNMF06, KFJ06, KJF+06, F̈99, CDI98, WCP07].

However, automated rating and filtering Web content by algorithms faces several challenges: Firstly, it is often difficult to automatically extract information from Web documents because they may contain lots of different content types including images, videos, Java applets or Adobe's Flash and Flex technologies. With the advent of the Social Web, highly interactive Web sites also increasingly employ techniques such as AJAX[12], which further complicate automated analysis due to Web documents changing and updating dynamically according to a user's interactions with the documents[13]. While it is easy for humans to analyze and understand such content, it is a much harder task for algorithms even with modern processing power. For example, image processing algorithms may be able to identify human faces or nudity in images with a certain accuracy, but such techniques are often restricted to very specific problem domains[14] and are rather computationally expensive [FFB96, JR02, RJB06, Sti06]. Secondly, results of machine learning algorithms in particular depend heavily on the quantity and quality of training input, and training input varies with a user's individual preferences and characteristics. An algorithm for binary classification, for instance, will not yield opti-

[12] Asynchronous JavaScript and XML (AJAX).

[13] This development creates challenges also in other domains. For example, search engines face the same problems when trying to crawl and index such "rich" Web documents [WG07].

[14] Anecdotally, there are even specialized "adult content recognition" methods for the detection of nipples in photos [WLWH10].

mal results if it is not trained with a sufficient number of samples of both classes, even though tricks such as PEBL [YHC02, YHC04] may help to a certain extent.

7.2 Folksonomy-driven Web Filtering

We have exploited the characteristics of folksonomies for the purpose of Web search personalization in Chapter 6 in order to *facilitate* the retrieval of resources on the Web. In the following sections, we investigate how collaborative tagging and folksonomies can be leveraged for *preventing* access to Web resources. Based on the recent scientific findings on Web document classification with folksonomies described in research works such as [YLMH09, BS08, ZMF09] and in the previous chapters of this thesis, we argue that an alternative approach to Web filtering would be to harness the concepts of collaborative tagging and folksonomies in order to create a user-driven filtering application of the Web.

Firstly, folksonomies are very popular among Web users and have been shown to cover already a considerable fraction of resources on the Web. For example, 48.1% and 17.8% of Web documents in our experimental data sets CABS120k08 and DMOZ100k06, respectively, were tagged by users, which is a significantly higher percentage than we have observed for existing rating systems such as ICRA (cf. Section 7.1.1). Secondly, users particularly employ tags for classification purposes, i.e. categorization of Web documents, and it has been found that these tag assignments are generally very accurate descriptions of the annotated resources [HKGM08, CSB+07]. Lastly, recent studies [ZMF09] show that classifiers based on folksonomy data can achieve even higher accuracy than content-based analyses. All these findings provide strong support for the applicability of folksonomies to the domain of Web filtering.

Compared to the traditional approaches presented previously, folksonomy-driven Web filtering leverages the most flexible and most powerful content processor available – the human brain – and is able to connect a multitude of participants via a community network. The first aspect helps to properly understand and rate even rich Web content such as images and videos, and overcomes the limitations of automated approaches to analyze such content. The second aspect helps to scale much better with the rapid growth of the Web than existing manual approaches as discussed in Section 7.1.1. Furthermore, folksonomy-driven Web filtering integrates users into the rating and filtering process, thus enabling active participation of users and increasing the transparency of such an approach. It also addresses the conflict of interest and the problem of different perceptions and interpretations when rating Web content: In our approach, the raters and the readers of a Web document are the same persons.

In the following sections, we describe how such an approach of folksonomy-driven Web filtering can be implemented in practice, and present a case study of a working prototype called *TaggyBear*.

7.3 TaggyBear: A Case Study

In this section, we describe the design and architecture of the folksonomy-driven Web filtering service *TaggyBear*, which we have developed as part of the research project with our industrial partner SES ASTRA S.A. The goal of TaggyBear is to provide a user-centric alternative to traditional Web filtering approaches. Its objective is the implementation of a technical platform that enables end users to protect themselves from malicious, objectionable or otherwise harmful content according to their personal preferences, and let them actively participate in the filtering process.

Similar to existing social bookmarking services such as Delicious, TaggyBear allows users to bookmark and tag Web resources. Seen in this way, TaggyBear can be used for normal tasks such as storing a user's personal bookmark collection (i.e. his personomy $\mathcal{P}_u$) and recommending interesting Web resources to friends. However, the same data is also leveraged to create new services which go beyond the standard scheme of collaborative tagging, namely Web filtering. With regard to Web filtering, TaggyBear leverages the "wisdom of the crowd" in the spirit of similar community projects such as Wikipedia or PhishTank[15]. Firstly, TaggyBear allows users to collectively submit ratings of Web resources to the system, which are subsequently analyzed and aggregated for the purpose of Web filtering. Secondly, such rating information can be queried from TaggyBear. A user can specify which content categories – derived from tags – she or he personally deems objectionable or unwanted. Then, for each Web resource requested by the user, client applications such as the TaggyBear browser add-on can query the system for the corresponding rating information, and may grant or block access to the requested resource accordingly. For example, Figure 7.1 shows a screenshot of the TaggyBear browser add-on blocking access to an objectionable Web site because the majority of users have rated it as pornographic.

It is important to note that any filtering of Web resources in this scenario is performed *on the client side*. The technical platform of TaggyBear only provides the necessary rating information on request – it's up to the user and his client application to decide how this information should be used.[16] This approach is very different from current server-side filtering where, for example, access to Web resources by users is blocked by their Internet service providers. Hence, TaggyBear acts as a kind of social "overlay" of the Web, and represents a more democratic approach to Web filtering.

In the following sections, we will focus our description of TaggyBear on its Web filtering component, i.e. we will not discuss other system components such as user management, authentication, browsing its Web site, or system administration.

[15] PhishTank, http://www.phishtank.com/, is a community-based anti-phishing service. Users can submit the URLs of suspected phishing Web sites, and vote on whether a submission is truly a phishing site or not. Last retrieved on March 01, 2010.

[16] For example, a client software for end users should give full control to its users. In a school environment however, the network administrator might want to enforce a global filtering policy, and interface a central Web proxy server with TaggyBear.

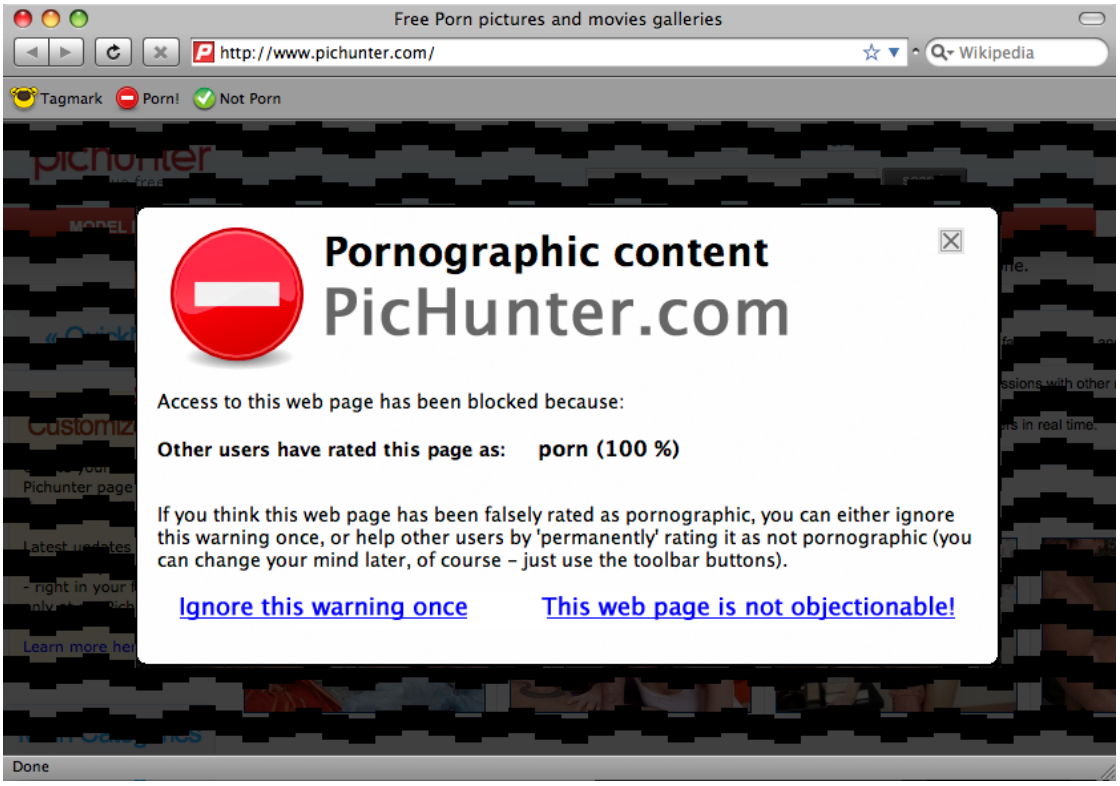


Figure 7.1: **TaggyBear browser add-on.** In this screenshot, the browser add-on has blocked access to the Web site PicHunter.com because a) the majority of users have rated it as pornographic, and b) the user configured the add-on to block access to any pornographic Web site. The browser add-on is described in Section 7.3.4.

7.3.1 Prerequisites and System Requirements

Prerequisites

Due to the power-law usage patterns of folksonomies (see Section 2.4.4), we can make the following general assumption:

> **Assumption:** A user rates only a small subset of all resources within a folksonomy, i.e. $|\mathcal{R}_u| \ll |\mathcal{R}|$.

This assumption has three consequences for the design and anatomy of the TaggyBear system. Firstly, we can expect that most resources for which a user requests rating information from TaggyBear are not in his personomy $\mathcal{P}_u$. This means that most queries for rating information will not return any individual user ratings. Secondly, we can expect that a user u is significantly more likely to query ratings from TaggyBear when browsing the Web than he is to submit such information himself. The system will

therefore have a higher number of read operations than write operations. Thirdly, we can expect that most information about rated Web resources, i.e. $r \in \mathcal{R}$, was (collectively) contributed by users other than the one requesting the information. This means that most rating information returned from the system will be derived from the community's rating activities.

System Requirements

Several requirements must be met in order to implement a folksonomy-driven approach to Web filtering. In this section, we describe the main requirements of TaggyBear with regard to Web filtering.

On the functional side, the basic requirement is that users may query and submit ratings of Web resources to the system. Additionally, the system should be able to distinguish between a user's individual rating and the community rating of a Web resource, so that a user's individual opinion can be properly accounted for even when it diverges from the opinions of other users. Furthermore, we have seen in Section 2.6 that folksonomies are often the target of spammers. A system rating should therefore be integrated to serve as a manual moderator function for hardening the system against abuse and spam, e.g. to protect against rating attacks that would result in unintentionally blocking access to targeted Web sites. However, it should be up to the user to decide which type of rating should take precedence over the other.

On the technical side, a very important requirement is the lookup performance for querying rating information from the system. As we have described in the previous section, most interactions with TaggyBear will be lookup operations for retrieving rating information from the system[17]. Client software such as the TaggyBear browser add-on (see Section 7.3.4) will direct a large and steady number of queries towards the service. Whenever a user visits a new Web document, the add-on will trigger a request for rating information. Particularly, the response times of lookup operations must be very fast so as not to affect the user experience when browsing the Web. The empirical usability guidelines for response times of Web services are described by Nielsen [Nie99], who states that 0.1 seconds (100 ms) is the ideal response time. In this case, the user does not sense any interruption. One second (1,000 ms) is the highest acceptable response time, and response times above one second interrupt the user experience. Hence, we target for a response time of lookups faster than 100 ms.

Furthermore, users expect that any modifications to their personal data, particularly their personomies $\mathcal{P}_u$, take effect immediately. This means that the system should allow for real-time or near real-time access and updates to individual user data, which includes user ratings. Lastly, TaggyBear should be able to analyze and aggregate individual ratings of users into community ratings at a performance that meets real-world demands. In their study of Delicious, which we can use as a reference system, Wetzker et al. [WZB08] estimate about 7.5 million posts (i.e. ratings) submitted to Delicious

[17] This scenario also includes users of TaggyBear that are not registered in the system, i.e. users without user accounts. In this case, only community and system ratings may be retrieved from TaggyBear. See Section 7.3.2 for more information about the different types of ratings in TaggyBear.

in one month. Hence, TaggyBear should be able to process such a volume of data in reasonable time.

We start our description of the TaggyBear system with an outline of its data model, because the data model has a strong influence on the technical implementation of TaggyBear. Then, we continue with the discussion of the actual system components.

7.3.2 Data Model

The basic requirement of TaggyBear is that users may query and submit ratings of Web resources to the system. In this section, we outline the data model and structure of these ratings. Particularly, we describe how users can *submit* ratings of Web resources through folksonomies in the scenario of Web filtering, i.e. how *tagging* can be transformed to *rating* of Web resources (Section 7.3.2). We also describe the different types of rating information that can be *queried* from TaggyBear on request (Section 7.3.2).

From Tagging to Rating

The concept of tagging allows users to conveniently annotate Web resources. Following our argumentation in Section 6.3 about the mapping of tag assignments to a topic space for classification purposes, we can similarly consider a user's tag assignments of a resource as his classification and thus his rating of the resource in the scenario of Web filtering. However, there is a drawback to the standard model of tagging in this context (cf. [RP97]): either tag assignments exist, or they don't. What is missing is a straight-forward way for users to explicitly express non-relation of tags and documents or, generally, to provide negative rating feedback. A community-based Web filtering approach needs such a feature for allowing users to disagree with each other. If, for example, one user tags (and thus rates) a Web document as `pornography`, it should be possible for other users to express their objections to this user's rating, i.e. to voice their opinion "this Web document is *not* pornographic". We have also observed from test users that it is often easier and quicker to use such negations in the context of Web filtering: "I am unsure what this document is about, but I know for sure what it is *not* about". In addition, it allows for better contextualization, and rating systems such as ICRA include veto-type descriptors for this very reason. These descriptors can be used to denote that a Web document shows pictures of a naked woman but, for example, in a medical and thus non-pornographic context. Hence, to truly use collaborative tagging for the scenario of Web filtering and rating Web documents, a voting element must be integrated into the standard model of tagging.

In Section 2.2, we have formally defined the standard model of folksonomies and collaborative tagging. For the purpose of Web filtering, we follow the related ideas of Gruber [Gru07] and Hotho et al. [HJSS06b, HJSS06a], and add a simple but effective extension to this model that integrates a voting feature but is at the same time backwards compatible to the standard model. The extended model of a rating-enabled folksonomy $\mathcal{F}^*$ is shown in Definition 7.3-1. Other definitions such as a user's rating-enabled personomy $\mathcal{P}_u^*$ are adapted accordingly.

Definition 7.3-1 (Rating-enabled Folksonomy). A rating-enabled folksonomy is a quintuple $\mathcal{F}^* := (\mathcal{U}, \mathcal{T}, \mathcal{V}, \mathcal{R}, \mathcal{Y}^*)$, where $\mathcal{U}$, $\mathcal{T}$, $\mathcal{V}$ $\mathcal{R}$ are finite sets whose elements are called *users*, *tags*, *tag votes* and *resources*, respectively. $\mathcal{Y}^*$ is a quaternary relation between these sets, i.e. $\mathcal{Y}^* \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{V} \times \mathcal{R}$, called *ratings*.

As can be seen in the definition, the only addition to the standard model is the vote set $\mathcal{V}$. A tuple (u, t, v, r) represents the rating y that user u rated resource r with tag t and tag vote v. In our implementation, we define the vote set $V := \{0, 1\}$, and use a vote value of 1 to denote a positive relation of tag and resource ("is about") and a value of 0 to denote a negative relation ("is not about"):[18]

$$vote(t, r) := \begin{cases} 1, & \text{if resource } r \text{ is about topic } t \\ 0, & \text{if resource } r \text{ is } \textit{not} \text{ about topic } t \end{cases} \tag{7.1}$$

With regard to the backwards compatibility to the standard model of folksonomies, we consider a default value of 1 ("is about") when the vote information is missing. This enables us, for example, to readily use data from existing collaborative tagging systems (e.g. by importing a user's personomy $\mathcal{P}_u$). With regard to the syntax of tagging, most collaborative tagging systems let users specify tag assignments as space- or comma-delimited lists of words via their user interfaces (see screenshot in Figure 3.1). Integrating tag votes into this popular scheme is very easy: Prefixing a tag with a minus sign "-" is defined as a negative vote (e.g. `-pornography`), whereas in any other case the tag vote is considered to be positive.

In summary, we have enhanced the standard model of folksonomies and collaborative tagging with a voting element, with which we can transform *tagging* to *rating* of Web resources. The enhanced model is backwards compatible, and does not require a change of tagging habits on the users' side. We believe that the latter aspect is very important in practice because ease of use has been found to be a critical success factor of folksonomies (cf. Chapter 2).

Rating Types

Following the system requirements, TaggyBear may return up to three different types of ratings – if available – for a Web resource r:

- the *user rating*
- the *community rating*
- the *system rating*

[18] The video sharing service and collaborative tagging system YouTube, for example, recently moved from 5-stars ratings to simple like/dislike ratings [Raj09] because most of their users focused on assigning either 1 star or 5 stars to a video. This indicates that a binary vote scheme can also be effective for collaborative rating in practice.

These ratings correspond to the opinions of the individual user, the community of users as a whole, and the system operators of TaggyBear, respectively, on how the resource should be rated. Since these judgements about a resource can diverge, all three rating types are returned by TaggyBear, and their order of preference is decided according to the user's preferences specified in client applications (see the browser add-on described in Section 7.3.4). It should be noted that the distinction into these three different rating types is also important for the system design as we will see later.

For convenience, we use the singular "rating" to refer to the set of a user's ratings of the same Web resource, with similar notions for community ratings and system ratings. The terms "rating" and "post" (cf. Section 2.2) are thus equivalent for the context of this chapter.

The *user rating* is the user's individual rating of a resource r, and is thus derived from his personomy $\mathcal{P}_u^*$. When user u queries his personal rating of r from TaggyBear for filtering purposes, the quadruples $(\mathbf{u}, t, v, \mathbf{r})$ are transformed to a list of (t, v) pairs. For the remainder of this chapter, we write such ratings as `tag:vote` pairs, e.g. `research:1` for $(\texttt{research}, 1)$. With regard to the response times of lookup operations, an important characteristic of user ratings is that they must be queried separately for each individual user, i.e. user ratings do not benefit much from optimization strategies such as server-side caching. Hence, they are comparatively costly to retrieve from the system. In Section 7.3.4, we describe how client-side optimizations can mitigate this problem in practice.

The *community rating* is the aggregation of all user ratings of the resource, i.e. the restriction of $\mathcal{F}^*$ to r. When the community rating of a resource is requested from Taggybear, this aggregation of quadruples $(u_i, t, v, \mathbf{r})$ is returned as a list of triples $(t,\ v_{pos}(t,r),\ v_{total}(t,r))$, where $v_{pos}(t,r)$ and $v_{total}(t,r)$ denote the number of positive[19] and total[20] votes, respectively, for tag t. This data can be used for calculations such as the percentage of positive votes per tag, so that client applications can be more flexible with regard to the policy and decision-making process of Web filtering (e.g. "Consider the community vote for a particular tag to be positive when more than 50% of users voted positively, but only if at least 5 users rated the resource."). For the remainder of this chapter, we write such ratings as `tag:pos:total` triples; for example, `science:66:72` means that 66 of 72 users voted positively for the tag `science`. With regard to the response times of lookup operations, community ratings of the same resource do not vary across different users. Hence, they benefit significantly from optimization strategies such as pre-computation and server-side caching.

The *system rating* is a special rating provided by the TaggyBear operators when and where needed, and is returned in the same format as user ratings. It is used to harden the service against spam and abuse, and serves as the manual moderator function of TaggyBear by providing an operator-managed "whitelist" and "blacklist" of Web re-

[19] The number of positive votes for tag t of a resource r is defined as $v_{pos}(t,r) := |\{(u_i, t_j, v, r_k) \in \mathcal{Y}^* | t_j = t \wedge v = 1 \wedge r_k = r\}|$.

[20] The total number votes for tag t of a resource r is defined as $v_{total}(t,r) := |\{(u_i, t_j, v, r_k) \in \mathcal{Y}^* | t_j = t \wedge r_k = r\}|$.

sources similar to services such as Google Safe Browsing[21]. However, a client software may chose not to respect the system rating at all, so the system rating is in fact a "voluntary" moderation feature from a user's perspective and not a hidden type of censorship. With regard to the response times of lookup operations, system ratings of the same resource do not vary across different users. Hence, optimization strategies such as pre-computation and server-side caching can be readily employed.

The three rating types are illustrated in Figure 7.2. To give an impression of how such ratings look in practice, consider the exemplary rating information for the homepage[22] of the Hasso Plattner Institute shown in Table 7.2.

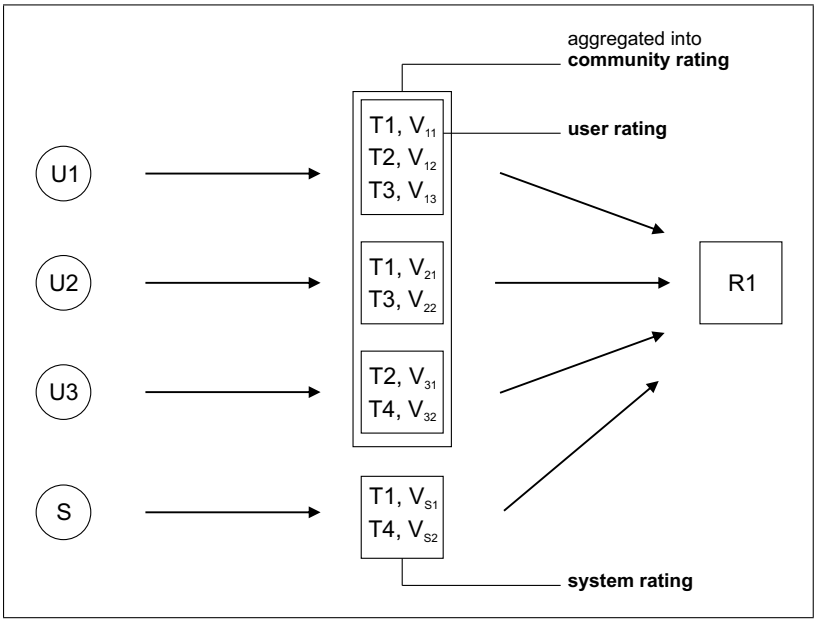


Figure 7.2: **Rating types in TaggyBear.** The data model for ratings is closely related to the model of broad folksonomies shown illustrated in Figure 2.3. In this example, the individual ratings of users *U1*, *U2* and *U3* are aggregated into a single community rating of the resource *R1*. The special system rating is shown at the bottom of the figure.

7.3.3 System Overview

Following the requirements outlined in Section 7.3.1, the TaggyBear system is separated into two complimentary parts. Firstly, there is a "synchronous" part, which is responsible for handling *user ratings*. Secondly, there is an "asynchronous" part, which is responsible for handling *community ratings* and *system ratings*. This separation is partly

[21] Google Safe Browsing, http://code.google.com/apis/safebrowsing/, last retrieved on March 01, 2010.
[22] Hasso Plattner Institute, http://www.hpi.uni-potsdam.de/.

user rating
research:1, science:1, weblab:1, phd:1, hpi:1, porn:0
community rating
hpi:56:56, research:42:42, germany:26:31, ..., porn:0:1
system rating
(none)

Table 7.2: **Exemplary ratings for the homepage of the Hasso Plattner Institute.** Here, 26 out of 31 users voted positively for the tag "germany". A system rating is not available in this example.

due to the different characteristics of the three rating types, namely that community ratings and system ratings can benefit from server-side optimization techniques such as caching, whereas user ratings cannot to the same extent. Additional reasons are given in the explanations described in the following sections.

An overview of the TaggyBear system is shown in Figure 7.3. For the sake of readability, we omitted some technical components such as reverse proxies and caches. The Linux-based server infrastructure consists of six main components as follows, all of which are or are based on free and open source software:

- **HTTP server:** Nginx[23].
- **Application server:** Pylons[24].
- **Data stores:** MySQL Community Server[25], Tokyo Cabinet and Tokyo Tyrant[26].
- **Message buffer:** Pylog (*Py*thon *log*ger), a custom development based on Twisted[27].
- **Distributed computing cluster for data aggregation and analysis:** Hadoop[28].
- **Client application:** Add-on for the Web browser Mozilla Firefox[29].

With regard to the synchronous part, TaggyBear provides users with a real-time interface to their individual ratings. Particularly, there is no delay for adding, updating or deleting user ratings as is the case for community and system ratings. In addition, flexibility with regard to future feature additions to TaggyBear was an important criterion for us. Hence, we chose a relational database management system (RDBMS) as a data store for this type of data, namely the *MySQL Community Server*. It should be

[23] Nginx, http://nginx.net/, last retrieved on March 01, 2010.
[24] Pylons Web framework, http://www.pylonshq.com/, last retrieved on March 01, 2010.
[25] MySQL Community Server, http://www.mysql.com/, last retrieved on March 01, 2010.
[26] Tokyo Cabinet/Tyrant, http://1978th.net/tokyocabinet/, last retrieved on March 01, 2010.
[27] Twisted, http://twistedmatrix.com/, last retrieved on March 01, 2010.
[28] Hadoop, http://hadoop.apache.org/, last retrieved on March 01, 2010.
[29] Mozilla Firefox, http://www.mozilla.org/, last retrieved on March 01, 2010.

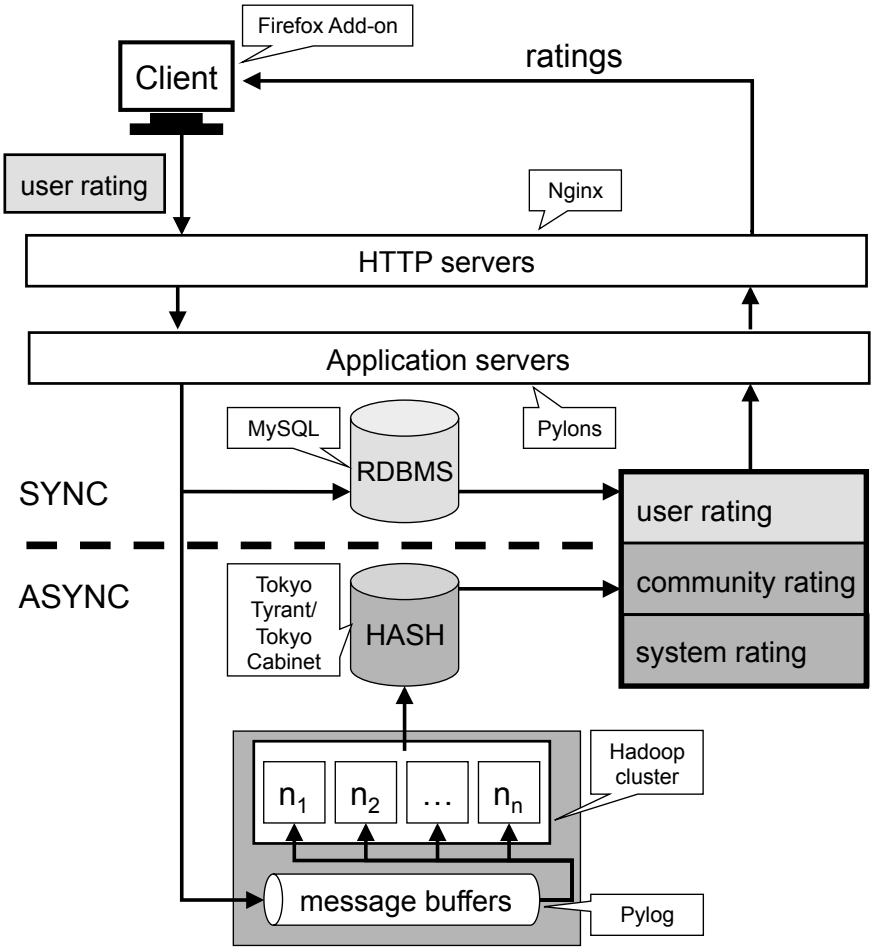


Figure 7.3: **System overview of TaggyBear.** For readability, we have omitted caching components and do not show multiple instances of components such as application servers separately.

noted that the RDBMS is only used for serving a user's personal data but not for data aggregation or the results thereof.

The asynchronous part of TaggyBear is responsible for handling community ratings and system ratings. Next to managing the data store from which users can query these ratings, the asynchronous part performs data analysis and data aggregation in a three-step staging process to populate and update this data store. In the first step, user ratings submitted to TaggyBear are buffered for batch processing. In the second step, the buffered data is analyzed and aggregated into community ratings. In the last step, the data store for community ratings is updated with the aggregated information. System ratings are handled similarly.

More precisely, the idea is to temporarily store any incoming user ratings in a so-

called message buffer[30] (in addition to permanently storing them directly in the RDBMS for the individual user), and periodically process this message buffer through batch jobs in order to aggregate user ratings into community ratings. The advantage is that the RDBMS does not need to be accessed for data analysis and data aggregation, which significantly reduces system load and computational requirements for the RDBMS[31]. The staging setup implies that there is a certain delay before newly submitted information is reflected in the community and system ratings that can be queried from the system. However, opting for such a strategy of *eventual consistency* enables us to optimize the system for scalability of data aggregation.

Community and system ratings are queried from a data store that must be very efficient with regard to lookup operations. Here, the flexibility of query statements of a full-fledged RDBMS with features such as JOIN operations is not needed. We therefore chose a hash table as the data store for community and system ratings, because this key-value data structure allows for constant-time $O(1)$ lookups on average [CLRS01]. We use the hash table implementation of *Tokyo Cabinet* and its companion *Tokyo Tyrant* for this task. Tokyo Cabinet is the successor of the well-known *QDBM* library[32], and provides the actual data store functionality. Tokyo Tyrant adds a network interface to Tokyo Cabinet so that the hash table can be accessed remotely from distributed machines.

For data aggregation and analysis, we selected the *Hadoop* framework, which implements the concepts of Google's patented *MapReduce* framework [DG04] and allows for a distributed, parallel execution of programs on clusters of commodity hardware. It is a popular data processing tool that is used by companies such as Adobe, Amazon, Facebook, Google, and IBM. For example, it is used for powering the Yahoo! search engine [Bal08]. Hadoop also runs every data processing job on its own fault-tolerant distributed file system called *HDFS*, which has been designed and optimized for reading and writing smaller numbers of large files. However, the use case of TaggyBear is the exact opposite, as updates – in the form of newly submitted user ratings – are rather large numbers of very small amounts of data. We therefore developed our own buffering solution called *Pylog* based on the *Twisted* networking framework. Buffered data is then periodically copied to HDFS for further processing.

In the following sections, we describe the TaggyBear system and its components in greater detail.

[30] In the case of TaggyBear, messages are user ratings of Web resources.

[31] For example, one positive effect is that more concurrent requests for personal data can be served by a single RDBMS instance.

[32] Quick Database Manager, http://qdbm.sourceforge.net/, last retrieved on March 01, 2010. QDBM is a library of routines for managing a database. The database is a simple data file containing records, where each record is a key-value pair. Both binary data and character string can be used as a key and a value.

7.3.4 Using TaggyBear

Web Interface – for Humans

Apart from client applications, the main interface and landing point for users is the TaggyBear Web site. It provides users with a number of features such as managing and backing up their personomies, subscribing to news feeds of recently posted resources, or looking up information about Web resources. Figure 7.4 shows a user's personomy as seen on the TaggyBear Web site. An excerpt of the tagging and rating information of a Web resource is illustrated in Figure 7.5.

The Web site and the main application logic of TaggyBear is implemented with the Python Web framework *Pylons*, which thus acts as the application server(s) of the system. Pylons is a very lightweight and very modular framework: It is easy to use third-party libraries for database access, templating, caching, request dispatching et cetera. This is ideal for our intended setup, which involves separated, heterogeneous data stores and caching layers. The HTTP server Nginx is logically placed in front of Pylons for serving static files such as images or CSS files, which reduces server load on the application servers and cuts down response times.

Application Programming Interface (API) – for Machines

TaggyBear allows programmatic access to its data via a Web-based RESTful API [Fie00]. REST, or *Representational State Transfer*, refers to a collection of architectural principles used for transfer of information over the Web, but is now used to describe simple RPC-based protocols using XML over HTTP [Hen06]. Its benefits – which are part of the reasons why a lot of Web applications are using REST – include being lightweight and cacheable (cf. Section 7.3.8), which helps to reduce server load and improves scalability. Additionally, it uses the inherent HTTP security model, which means that system operators can restrict certain HTTP methods (e.g. `DELETE`) to specific URIs through firewall configurations.

Path prefix for REST API URIs: /api/rest/*version*

GET	/ratings/*hash*	get user, community, and system ratings for a resource
POST	/*user*/ratings	add a new user rating
PUT	/*user*/ratings/*hash*	update a user rating
DELETE	/*user*/ratings/*hash*	delete a user rating

Table 7.3: **Excerpt of REST API features of TaggyBear.** Web resources are identified by the MD5 hashes of their URLs. The last three API calls in this listing require user authentication.

Like the Web interface, the TaggyBear API is implemented with Pylons. The API features include adding, modifying, retrieving and deleting ratings from TaggyBear (see Table 7.3). For example, the browser add-on uses the API to submit new user ratings to

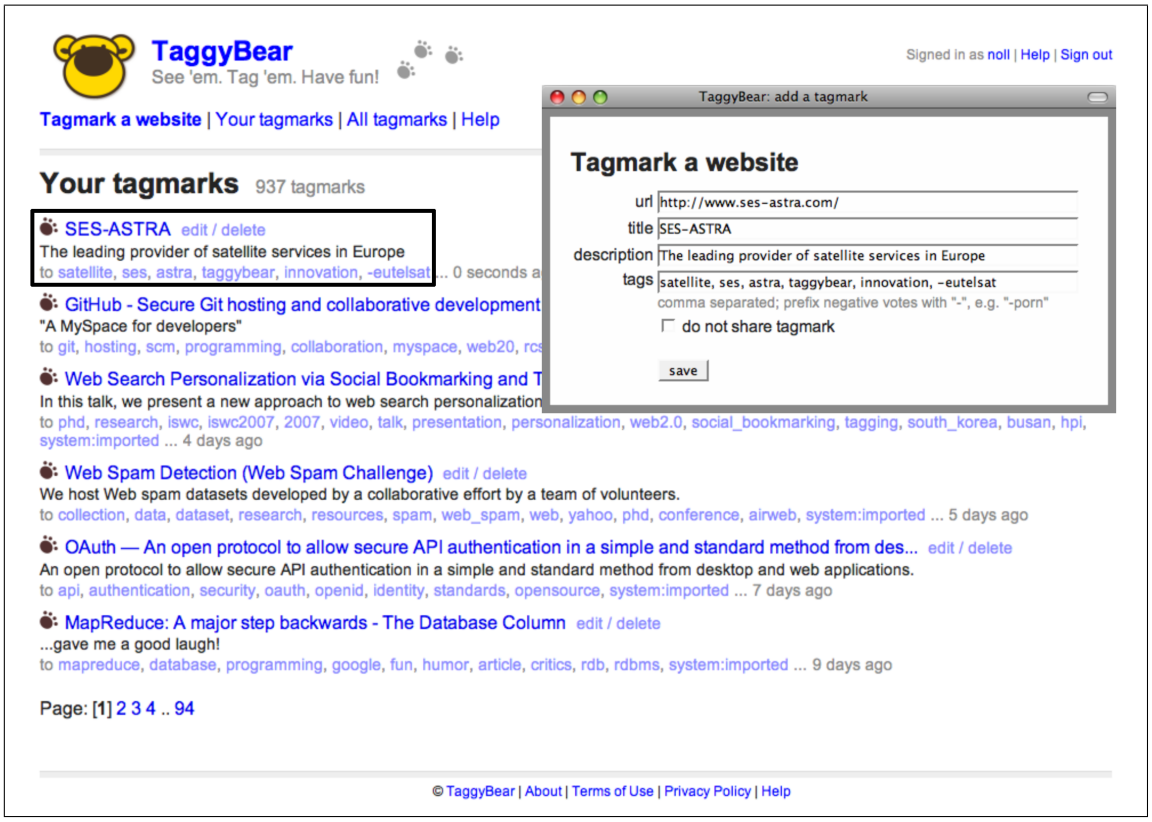


Figure 7.4: **A user's personomy on the TaggyBear Web site.** Posts (ratings) of Web resources are called "tagmarks" in the TaggyBear user interface. Users can also tag/rate Web resources through the user interface as shown in the upper right of the figure. The highlighted post on the left side (black frame) corresponds to the data entered in the dialogue window.

TaggyBear and to lookup rating information about Web resources. Resources are identified by the the MD5 hashes of their URLs. MD5 [Riv92] is a widely used cryptographic hash function with a 128-bit hash value. As an Internet standard, implementations of MD5 are readily available for a plethora of programming languages including C/C++, Java, JavaScript, Perl, PHP, Python and Ruby, which helps developers when creating client applications for TaggyBear.

Client Applications

Since the main purpose of Web filtering is to protect users from unwanted or objectionable content as they browse the Web, we have developed a TaggyBear browser add-on for Mozilla Firefox[33].

[33]Mozilla Firefox Web browser, http://www.mozilla.com/firefox/, last retrieved on March 01, 2010.

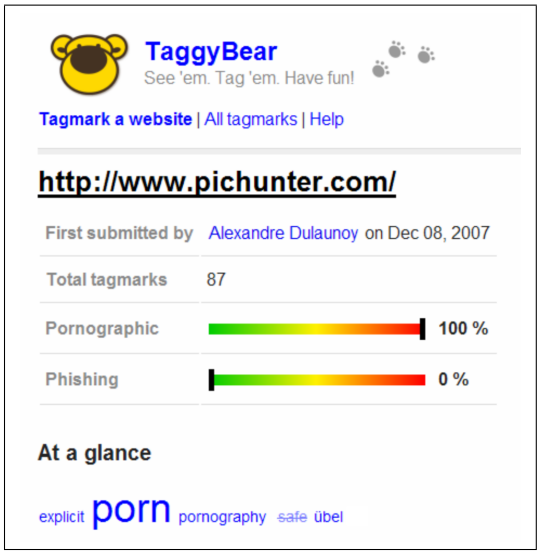


Figure 7.5: **A Web resource's details on the TaggyBear Web site.** In this zoomed screenshot, some information about the Web site PicHunter.com is shown. The user interface highlights the community votes for the two tags `pornography` and `phishing` because these tags have been selected for filtering by the particular user viewing the Web site.

The user can configure the add-on's filtering features according to his personal preferences, particularly by specifying a) which content categories (tags) should be filtered and b) the order of preference for the three rating types. By default, the order of preferences is configured as:

$$\textit{user rating} > \textit{system rating} > \textit{community rating} \tag{7.2}$$

Hence, the end user's own rating is the ultima ratio. The system rating has precedence over the community rating to protect the user against rating attacks on the TaggyBear system. In most cases, however, only community ratings of a resource will be returned by Taggybear, which is due to our assumption described in Section 7.3.1 and because the use of system ratings is restricted to prevent specific cases of system abuse.

Whenever a user visits a new Web document through the Web browser, the add-on queries the TaggyBear API for rating information. When a Web document is blocked, the add-on obfuscates the document's content and presents a warning window to the user, which also informs him why the add-on triggered the protection mechanism as shown in Figure 7.1. The warning window also provides interface elements to temporarily ignore the warning or to permanently overwrite the filtering decision with the user's own, individual rating of the document. The latter effectively allows the user to create his own whitelist and blacklist of Web resources very easily.

The browser add-on also includes interface elements to create new ratings so that

users do not need to visit the TaggyBear Web site for doing so. Furthermore, the add-on provides interface elements to quickly rate and assign Web documents to specific content categories with a single click ("Porn" and "Not Porn" buttons in Figure 7.1).

Another way to use TaggyBear is to interface it with Web proxy servers such as Squid[34] in a centrally managed network. The proxy servers can be configured to query TaggyBear for rating information about any requested Web resources and block access depending on how the community of users have rated these pages.

7.3.5 Data Flows

In this section, we describe the basic data flows for adding user ratings to and retrieving rating information from TaggyBear, respectively. We only describe the data flows for the TaggyBear API because the data flows for the Web interface are similar. In Section 7.3.8, we will discuss how these basic data flows can be further optimized through server-side and client-side techniques.

Submitting user ratings

The data flow for *adding* user ratings, i.e. write operations, is shown in Figure 7.6. When a user submits a new rating through a client application to the API via an HTTP POST request, the request and its payload is first run through several sanity checks and input filtering. If it passes these tests, the rating is stored in the user's personomy $\mathcal{P}_u^*$ in the RDBMS, and then submitted to the message buffer (the RDBMS and the message buffer are described in Section 7.3.6). In this case, the API returns a `200 OK`[35] HTTP status code to the client. If it does not pass the test, the API returns an appropriate error code, for example `403 FORBIDDEN` for user authentication failures.

The previously described steps occur synchronously. The processing of user ratings in the message buffer is performed asynchronously at periodic intervals[36], at which the buffered user ratings are copied to the Hadoop cluster's distributed file system HDFS. As described in Section 7.3.7, Hadoop MapReduce jobs will be run to aggregate the user ratings into community ratings, and the aggregation results will be inserted into the hash table via Tokyo Tyrant.

Retrieving ratings

The data flow for *retrieving* rating information, i.e. read operations (lookup), is shown in Figure 7.7. When a client requests rating information about a Web resource, the query is routed via the API to the hash table (see Section 7.3.6) which will return any available community and system ratings. If the request was sent from an authenticated

[34] Squid is a free and open source caching proxy for the Web supporting HTTP, HTTPS, FTP and other protocols. It is available at http://www.squid-cache.org/, last retrieved on March 01, 2010.

[35] The HTTP status codes are defined in RFC 2616 "Hypertext Transfer Protocol – HTTP/1.1", available at http://www.ietf.org/rfc/rfc2616.txt, last retrieved on March 01, 2010.

[36] In our current implementation, this interval is set to 30 minutes.

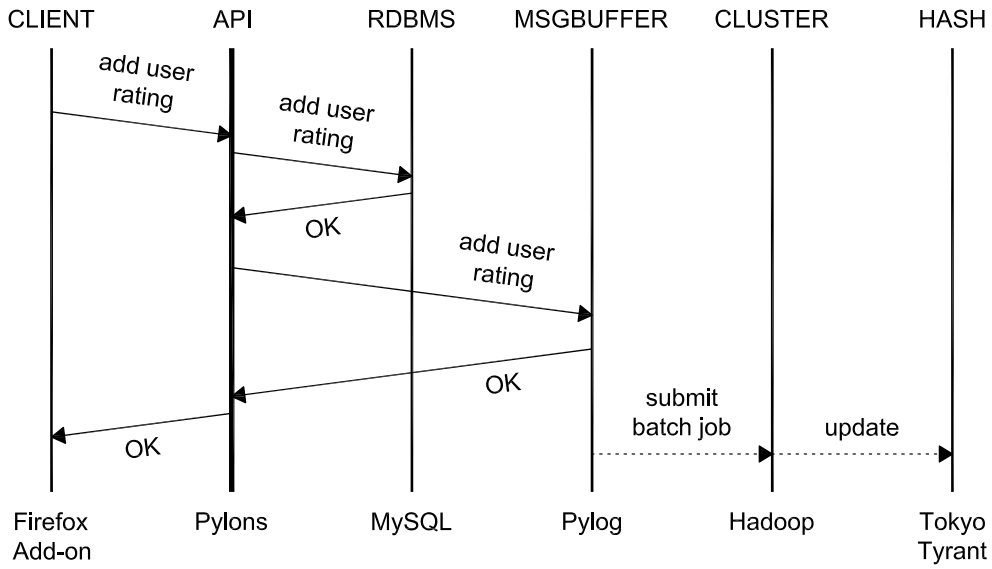


Figure 7.6: **Data flow for submitting user ratings (write operation).** The dashed lines represent asynchronous tasks that are carried out at a later time. For readability, the HTTP server in front of the API has been omitted because it simply routes POST requests directly to the API. The resulting delay is negligible and hardly measurable (<1 ms).

user, the RDBMS is also queried for any available user rating. Finally, the rating results are returned back to the client.

Figure 7.7 shows that unauthenticated requests – e.g. from a Web proxy server configured to interface with TaggyBear, or from people using TaggyBear without registered user accounts – will only result in a query against the hash table for retrieving community ratings and system ratings. The response to such queries can be adequately cached because it is indiscriminately valid for any user. For authenticated users, a request will additionally result in a query against the RDBMS. We describe in Section 7.3.8 how this lookup process can be further optimized.

7.3.6 Data Storage

In this section, we describe the storage components of TaggyBear. The current setup consists of three different data stores:

- RDBMS – MySQL Community Server
- Hash table – Tokyo Cabinet/Tokyo Tyrant
- Message buffer – Pylog

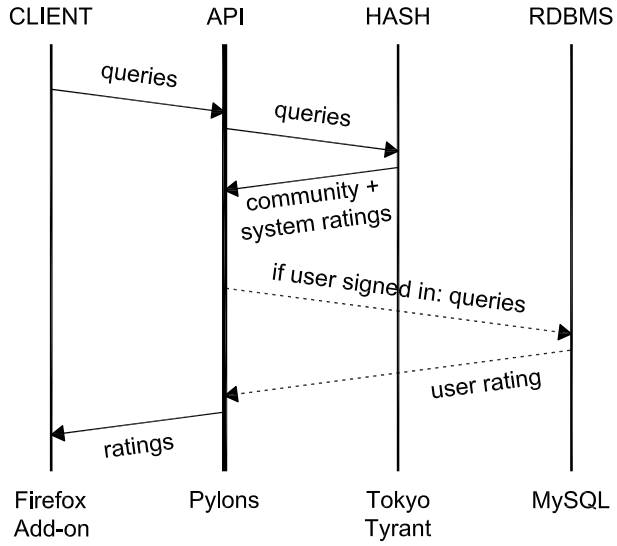


Figure 7.7: **Data flow for retrieving ratings (read operation).** The dashed lines represent tasks that are only carried out if the user is signed into TaggyBear.

As we said previously, the main purpose of the RDBMS is to store individual user data including user ratings, i.e. a user's personomy $\mathcal{P}_u^*$. It maintains a user-resource index, i.e. an index based on (u, r), to quickly retrieve user u's rating of a particular resource r from the data store. On the implementation level, we selected the popular MySQL database as the RDBMS because it is performant, well-tested and has a strong support within the developer community. Additionally, it provides load balancing and high availability features. Since MySQL can be considered as a standard tool for developing Web services, we do not go into details on how to properly scale MySQL databases are omitted. Interested readers are kindly referred to excellent works such as [ZB04, Hen06]. However, we note that an important benefit of our decision to use the RDBMS only for personal user data is that it enables us to horizontally scale MySQL databases by partitioning data by users (so-called *federation* [Hen06]). If, for example the load of a single-server setup reaches a critical level, we can split the load on the RDBMS to multiple server machines by migrating data of usernames starting with A-F to the first server, G-L to the second, and so on.

The hash table is powered by Tokyo Cabinet, and stores community ratings and system ratings. It is a key-value store, where in our case the key is the MD5 hash [Riv92] of the URL of a Web resource, and the value is the tuple of *(community rating, system rating)* of the resource. The hash table provides constant-time $O(1)$ lookup operations on average and $O(log\, n)$ in the worst case. With regard to write performance, the hash table implementation of Tokyo Cabinet can store 1 million records in < 1 seconds [Hir10]. This is an important criterion because we need to import large numbers of community ratings into the hash table after data aggregation. The application Tokyo Tyrant provides a network interface to Tokyo Cabinet, which we use when querying the

hash table for community ratings via the application server Pylons (see Section 7.3.5 on data flows). While there is a certain performance loss when accessing Tokyo Cabinet databases through Tyrant, it is still more than adequate for our needs. Tyrant also supports features for scalability and fault tolerance such as replication, update logs and hot backup.

The message buffer is a special data store as it is only used to buffer incoming user ratings until they can be processed and aggregated by the Hadoop cluster (see Section 7.3.7). Important features for us were, on the one hand, persistent storage of incoming messages to prevent data loss in case of system crashes or power outages, and on the other hand, a lightweight implementation that could easily be understood for customization and optimization purposes. We intended at first to use an off-the-shelf message queue for this task but could not find a software implementation that fully satisfied our needs. At the end, we decided to implement a simple yet efficient message buffer ourselves. *Pylog*, as we call it, is built on top of the *Twisted* networking framework for Python. Its only purpose is to accept "messages", in our case properly encoded user ratings, from the application server Pylons (e.g. submitted via the Web interface or the API) and reliably log them to file as quickly as possible. Unlike a message queue, Pylog does not need to guarantee FIFO[37] behavior. Instead, we add timestamps to messages and subsequently let Hadoop do the sorting of user ratings during the periodical MapReduce jobs. Whenever it is time to start another data aggregation run, we instruct Pylog to rotate its buffer file. The buffer file is then copied to the HDFS file system of the Hadoop cluster for processing. We tested the performance of Pylog on two identical machines with a Xeon E5335 2.0 GHz Quad Core CPU and 4 GB of RAM, running Ubuntu Linux 8.04 Server Edition with the default Linux kernel 2.6.2419-server. The machines were connected by a switched, full duplex Fast Ethernet network (100 MBps). The average throughput was 14,628 ratings per second, more than enough for our scenario. If needed, the message buffer can be horizontally scaled by adding more Pylog instances, and the sum of their buffer files can be jointly copied to the Hadoop cluster for the next MapReduce jobs.

7.3.7 Data Aggregation

In this section, we describe how individual user ratings are aggregated into joint community ratings through distributed computing. Since system ratings are almost identical to user ratings on a technical level, we do not specifically discuss them here.

On the implementation level, we have chosen Hadoop for data aggregation tasks because it allows for linear scaling in terms of data processing, and it is built for running on commodity hardware. If more processing power is needed, it is generally sufficient to just add more machines ("nodes") to the cluster. Computations on Hadoop clusters are performed with so-called MapReduce jobs [DG04, Fou10]: A MapReduce job usually splits the input data into independent chunks which are processed by *Map* tasks on different nodes in the cluster in a completely parallel manner. The Hadoop framework

[37] FIFO is an acronym for First In, First Out.

sorts the outputs of the maps, which are then piped as input to the *Reduce* tasks. Typically both the input and the output of the MapReduce job are stored in the HDFS file system. The Hadoop framework takes care of scheduling tasks, monitoring them and re-executes any failed tasks. For more information on Hadoop, the interested reader is kindly referred to works such as [Whi09, Nol07a].

We implemented the aggregation of user ratings into community ratings through two MapReduce jobs which are chained together, i.e. the output of the first job is the input of the second. The purpose of the first MapReduce job is data consolidation: It analyzes the timestamps of user ratings in the buffer file and merges multiple submissions or modifications of ratings *of the same resource by the same user* into a single "update" user rating. This can easily happen in our setup because using a message buffer can lead to pending updates, i.e. temporal data inconsistency. The purpose of the second MapReduce job is the actual aggregation: It combines the ratings of multiple users per resource into a single "update" community rating. Optionally, we could further analyze the input data during this step and, for example, filter out known spammers or promote the ratings of known expert users. It is thus possible to easily integrate the results of techniques such as SPEAR (see Chapter 5) into TaggyBear.

The total time needed for data aggregation can be approximated by the following formula:

$$t_{total} = t_{fs2hdfs}(I,U) + t_{Hadoop}(I,U) + t_{hdfs2fs}(O,C) + t_{TokyoTyrant}(O,C) \tag{7.3}$$

where $t_{fs2hdfs}(I,N)$ is the time needed to copy a message buffer file of I bytes containing U user ratings from the local file system to HDFS[38]; $t_{Hadoop}(I,U)$ is the time needed to aggregate these user ratings through Hadoop MapReduce jobs; $t_{hdfs2fs}(O,C)$ is the time needed to copy the aggregation output of O bytes containing C community ratings from HDFS to the local file system; and $t_{TokyoTyrant}(O,C)$ is the time needed to insert these community ratings into the hash table.

The times for data import and export, $t_{fs2hdfs}(I,U)$ and $t_{hdfs2fs}(O,C)$ are mainly network-limited, and can be controlled and optimized by proper networking setup. The time for the actual aggregation via the two described Hadoop MapReduce jobs, $t_{Hadoop}(I,U)$, is influenced by a variety of factors such as the number of Hadoop data nodes (which serve HDFS data) and tasktracker nodes (which process data) in the cluster, job parameters such as the number of reduce jobs to be run, and other factors such as the size of intermediate data. Figure 7.8 shows our benchmarking results for a cluster of four machines connected by a switched, full duplex Fast Ethernet network (100 MBps). Each machine was equipped with an Intel Xeon E5335 2.0 GHz Quad Core CPU (four cores), 4 GB of RAM, hardware RAID5 data storage, and was running Ubuntu Linux 8.04 Server Edition with the default Linux kernel 2.6.2419-server. We used Hadoop version 0.18.0 released in August 2008 for the benchmark. The results are averaged over several runs with the slowest and fastest results being removed from the samples.

[38] This includes the time needed for replicating data chunks to multiple cluster nodes as configured by Hadoop's `dfs.replication` parameter. The default replication value is 3.

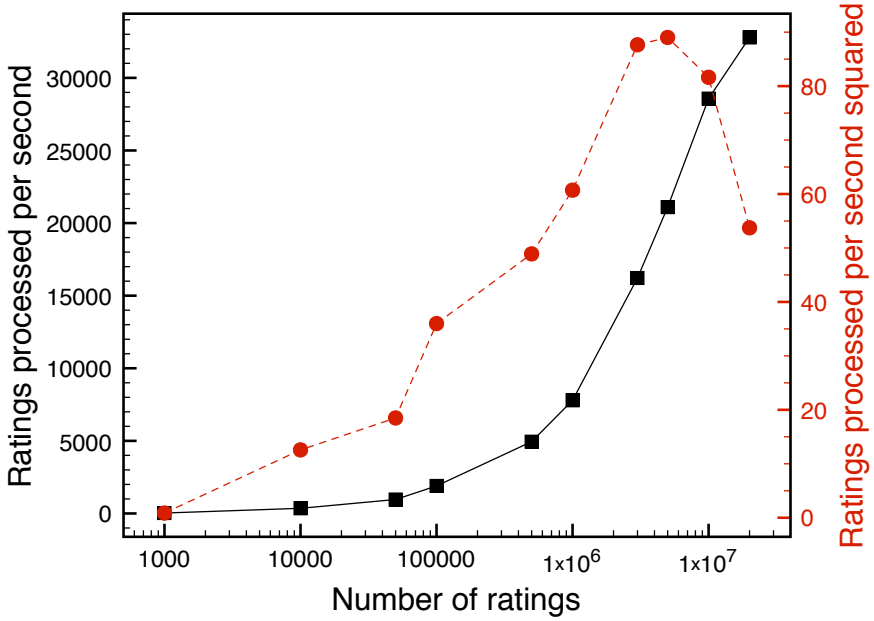


Figure 7.8: **Performance results for the aggregation of user ratings into community ratings through Hadoop MapReduce jobs.** The solid black line and the red dashed line show the ratings processed per second and the rating processed per second squared, respectively. Note the logarithmic scale of the x-axis.

We observed that the number of ratings aggregated per second *increased* with number of input bookmarks. This was mainly due to two reasons. Firstly, there is a rather fixed overhead for starting and running MapReduce jobs, which can be higher than the actual time for processing the input data. Secondly, if the input data is not sufficiently large, less than the total number of cluster nodes may be used for executing the job. In other words, more cluster nodes are only activated by Hadoop when needed, but if that happens they will increase total throughput. Still, the time to process 100,000 ratings was less than one minute. At some point, the "acceleration" that Hadoop receives from increasing amounts of input ratings starts to slow down. In our benchmark, this inflection point was approximately at 5 million ratings. Most likely, this was caused by limitations in network I/O, because we found that the cluster nodes could not receive input data fast enough and ended up idling, waiting for data. We believe that switching from Fast Ethernet to Gigabit Ethernet (1,000 Mbps) would therefore further improve the benchmark times. Nonetheless, the experimental results were encouraging: The cluster aggregated 1 million ratings in about two minutes, and 10 million ratings in less than ten minutes.

Finally, $t_{TokyoTyrant}(O, C)$ is the time needed to insert or update community ratings in the hash table through Tokyo Tyrant. A so-called *PUT* operation is equivalent to adding or overwriting the rating of a resource in the hash table. We measured 8,372 PUTs/s

on average on the same hardware when using Tyrant's Python API.[39] Again, this outcome is more than adequate for our scenario: The estimated number of posts per day on Delicious is, based on the per-month statistics reported by Wetzker et al. [WZB08], about 250,000 posts. This amount of posts can be inserted into the hash table in about 30 seconds. If needed, this performance could be further increased by using Tyrant's C/C++ API.

In a final experiment, we measured the total time t_{total} for processing and aggregating input data sets of varying sizes. The results are shown in Table 7.4. They provide strong support for the practical feasibility of the TaggyBear system.

Number of ratings	Total time
250,000	3.8 mins
1,000,000	6.7 mins
10,000,000	39.1 mins

Table 7.4: **Total data aggregation performance.** To put these numbers into relation, the social bookmarking system Delicious received an estimated number of 250,000 posts (ratings) per day and 7.5 million posts per month, based on the statistics reported by Wetzker et al. [WZB08].

In summary, we have demonstrated in this section that the performance of TaggyBear with regard to data analysis and data aggregation meets the system requirements for real-world scenarios.

7.3.8 Optimization and Data Caching

The performance of TaggyBear, particularly with regard to the data flow of retrieving rating information from the system (see Section 7.3.5), can be further improved through server-side and client-side optimizations.

Preventing unnecessary RDBMS Queries

Querying data from the RDBMS is generally more costly than from the lookup-optimized hash table, and user ratings are therefore less efficient to retrieve than community and system ratings. This means that unnecessary queries for user ratings should be avoided. The basic data flow described in Section 7.3.5 shows that a query for user ratings is sent to TaggyBear whenever an authenticated user visits a new Web document. However, we have explained in Section 7.3.1 that we can generally assume that a user only rates small subset of all resources within a folksonomy, i.e. $|\mathcal{R}_u| \ll |\mathcal{R}|$. We can similarly assume that he has rated only a (very) small subset of all resources on the

[39] Generally, TaggyBear queries the hash table first in order to retrieve the current (i.e. outdated) community rating of a particular resource in order to perform correct updates. In this experiment though, we were more interested in the *write* speed, i.e. PUTs. The reason is that a write operation is significantly more expensive than a read operation in our case.

Web, because folksonomies do not cover the Web fully. The consequence with regard to TaggyBear is that most queries from authenticated[40] users cannot and will not return any user ratings but the RDBMS will be queried regardless. If, for example, a user visits 1,000 Web documents when browsing the Web, and only 10 of these have been rated by the user, 990 hits on the RDBMS will be effectively pointless. A server-side caching strategy can generally not help in this situation because when the user rating of a resource is not available in the cache, it does not necessarily mean that the user has not rated the resource – the rating could simply not be cached yet. A better approach is to keep client applications informed about which resources have already been rated by the user, i.e. the set of resources $\mathcal{R}_u$. This allows client applications to only query for user ratings when there is such information available in the RDBMS. We implemented such a feature for the TaggyBear browser add-on, which locally stores the list of Web resources – more precisely, the MD5 hashes of their URLs – that have been rated by the user. Since a user might use more than one Web browser (e.g. one at home and one at work), the add-on periodically queries TaggyBear for updated information[41], i.e. whether and which new resources have recently been added to a user's personomy $\mathcal{P}_u$ and thus $\mathcal{R}_u$ to ensure consistency between different browsers installations.

Server-side Caching

An important technique that TaggyBear uses to reduce server load and improve scalability is *caching*. A cache temporarily stores data so that future requests for that data can be served faster [Hen06]. The setup of TaggyBear employs server-side and client-side caching at various places, of which the most critical are described in this section.

On the server side, we use memcached[42] to cache community ratings and system ratings. Memcached serves as an in-memory cache of data in the hash table, thus lowering the load on Tokyo Cabinet and Tokyo Tyrant.

Additionally, using memcached has another important benefit. In the TaggyBear system overview shown in Figure 7.3.3, an HTTP server (Nginx) is logically placed in front of the application server (Pylons). On the one hand, the HTTP server Nginx is responsible for serving static content such as images, icons and CSS files because it can perform this task significantly faster and more efficient than the application server Pylons. On the other hand, Nginx can be configured to bypass the application servers for lookup requests of community and system ratings, and reroute these queries directly to the cache. In combination with the client-side optimization regarding local copies of $\mathcal{R}_u$

[40] Non-authenticated users will never hit the RDBMS when querying TaggyBear for rating information. Hence, the described problem does not exist in their case.

[41] The MD5 hash of a URL has an uncompressed size of 32 Bytes. The volume of data that needs to be exchanged during such an update process is therefore quite small even for very active users.

[42] Memcached, http://memcached.org/, last retrieved on March 01, 2010. It is a free and open source distributed memory object caching system, generic in nature, but intended for use in speeding up dynamic Web applications by alleviating database load. On the technical level, memcached is an in-memory key-value store for small chunks of arbitrary data (e.g. strings, objects) from results of database calls, API calls, or page rendering. Notable users of memcached include Wikipedia, Facebook, Google and Yahoo.

described in the previous section, this means that most queries for rating information never even hit the application server(s), which further reduces the system load on these components.

Client-side Caching

For enabling client-side caching, we add ETag headers [FGM+99] to generated Web pages of TaggyBear where appropriate. ETag only helps if the entire page can be cached, and it can prove difficult to set up correctly in a load balancing scenario where a client may request the same content but get a response from different servers on each request [Hen06]. When used correctly though, ETag allows compatible Web browsers – and HTTP clients in general – to perform client-side caching of Web pages, thus further reducing server load.

The TaggyBear browser add-on benefits implicitly from Mozilla Firefox built-in ETag support. Additionally, the add-on also comes with its own TaggyBear-specific caching functionality, particularly for API requests. This is needed mainly for properly handling a user's GUI interactions within Firefox. For example, the add-on must handle tab switches in Firefox for processing and updating the currently displayed Web page in the browser (e.g., showing or hiding the warning window in Figure 7.1). Using a local cache of rating information prevents such user interactions from resulting in unnecessary queries to the TaggyBear system.

7.3.9 System Performance

In addition to the performance measurements described in the previous sections, we conducted further experiments to evaluate the system design and anatomy of TaggyBear. In this section, we report the results of a quantitative analysis of the response time of TaggyBear for lookup operations, i.e. the retrieval of rating information. The response time is the time passed between the submission of a query by a client application and the subsequent reception of the reply.

A variety of parameter combinations could be used for testing the response time. We decided to focus on the two most relevant scenarios in practice: Firstly, we measured the response time for an authenticated user who requests all three rating types at once. This is a costly operation, particularly because the user rating does not benefit much from server-side caching. Hence, we tested TaggyBear for the scenario of retrieving uncached user ratings in combination with cached community and system ratings. Secondly, we measured the response time for a user who requests the community and system rating of a Web resource from cache. This is the most common retrieval scenario, and should thus be the most important evaluation criterion.

We used the Apache Bench benchmarking tool[43], which is part of the popular Apache HTTP Server project[44], for stress-testing TaggyBear. The tool can be configured to gen-

[43] The documentation of Apache Bench is available at http://httpd.apache.org/docs/2.0/programs/ab.html, last retrieved on March 01, 2010.

[44] Apache HTTP Server project, http://httpd.apache.org/, last retrieved on March 01, 2010.

erate various types of data traffic for putting load on Web services. In our experiments, we simulated two variants: The first variant used Apache Bench to simulate one concurrent user sending 1,000 lookup requests to TaggyBear as fast as possible. This test can be considered as a baseline for the system performance. The second variant generated 100 concurrent users who sent a total of 10,000 lookup requests. All tests were run on a local network[45] against a TaggyBear installation consisting of only one instance, respectively, for the HTTP server (Nginx), the application server (Pylons), the RDBMS (MySQL), the hash table (Tokyo Cabinet/Tyrant) and the in-memory cache (memcached). Table 7.5 shows the results of our experiments.

Concurrent Users	Measure	U, not cached C+S, cached	C+S, cached	Target Value
1	μ	**17 ms**	**8 ms**	$\leq$ 100 ms
	σ	5 ms	2 ms	
100	μ	195 ms	**21 ms**	$\leq$ 100 ms
	σ	84 ms	6 ms	

Table 7.5: **Response times for retrieving ratings (read operation).** *U*, *C*, *S* denote user ratings, community ratings and system ratings, respectively. The table shows the mean response times (μ) including standard deviations (σ). Mean values in bold font meet the target value specified by the service requirements in Section 7.3.1.

We observed that TaggyBear responded very fast to lookup requests. It could easily handle the most common retrieval scenario with a mean response time of 21 ms for 100 concurrent users. As expected, queries which also hit the RDBMS took longer to process. While the mean response time was still considerably fast with 195 ms, TaggyBear exceeded the targeted response time of 100 ms when stressed with 100 concurrent users. Still, the result is encouraging considering that the expected percentage of such queries in relation to the number of total queries is rather small, which means that a single RDBMS server instance can still serve quite a large number of registered users[46].

[45] Testing on a local network minimizes the impact of *network latency* on experimental results. Network latency accounts for the time spent over the network to receive the query from the user plus the time spent over the network to send the reply to the user. For example, users accessing TaggyBear via slow network connections would experience slower response times because of the higher latency compared to users on fast network connections.

[46] Generally, three different user types can be distinguished for a system [IBM09]: *Registered users* make up the total population of individuals who are registered at the system. They represent the total user community, and can be active or concurrent at any time. *Active users* are logged on to the system at a given time and can send a processing request at any time. For example, users who are viewing the results returned from a query are active users, although they are not currently stressing the system. *Concurrent users* are not only logged on to the system (active), but are sending a request or waiting for a response. They are the only type of user actually stressing the system at any given time. The *concurrency ratio* is defined as the number of concurrent requests at any moment in time that affect the load on the system. A rule of thumb in estimating the load of a system is that approximately 1% percent of registered users or 10% of active users will equate to the number of concurrent requests the system must manage per second [IBM09].

In summary, we found that TaggyBear could meet the service requirements for the most relevant retrieval scenario. It should also be noted that we tested a single-instance installation of TaggyBear. We have described in the previous sections how the system has been designed to allow for horizontal scaling, which means that more computing resources can conveniently be added (e.g. more RDBMS instances) to cope with increasing service load. Hence, we argue that the results provide strong support for the practical feasibility of the system.

7.4 Discussion

In this chapter, we have proposed an alternative to traditional Web filtering techniques. The implementation of this alternative, the TaggyBear system, is based on the collective rating activities of users in folksonomies, and enables users to filter the Web according to their personal preferences through client-side filtering.

Interestingly, popular Web services such as Delicious have focused on providing users with tools that allow them to *submit* information to these services, but *leveraging* this information for use cases other than the original scenario (e.g. social bookmarking for Delicious) appears to be the exception. While we have put the emphasis of this chapter on how to exploit folksonomies and collaborative tagging for Web filtering, there are additional uses of this kind of user-contributed data. One such approach is to blend our proposed personalization method described in Chapter 6 with TaggyBear for creating a service for safer Web search. Figure 7.9 shows how both techniques can be combined at the example of Google Search. Here, the browser add-on described in Chapter 6 was extended to query TaggyBear for rating information about Web documents in search results. It is thus possible to protect users from objectionable or otherwise unwanted Web resources when performing Web searches as well.

Additionally, we have briefly outlined how TaggyBear can benefit from techniques such as SPEAR (see Chapter 5) to harden the system against junk input and to improve overall quality by properly analyzing and understanding folksonomy data. As such, the data aggregation components described in Section 7.3.7 provide a good starting ground for tackling practical problems of folksonomies such as spam (cf. Section 2.6).

Finally, it should be noted that any folksonomy- or community-driven approach depends on the actual quantity and quality of user-contributed information. We have shown in this chapter how a Web filtering service based on folksonomies and collaborative tagging can be implemented in practice. However, the eventual success (or failure) of such an approach can only be empirically verified with a critical mass of participants who actively use a system such as TaggyBear and who provide feedback on its usefulness and reception. Hence, we are looking forward to deploying the described TaggyBear prototype in a real-world setting and thus opening it to public access.

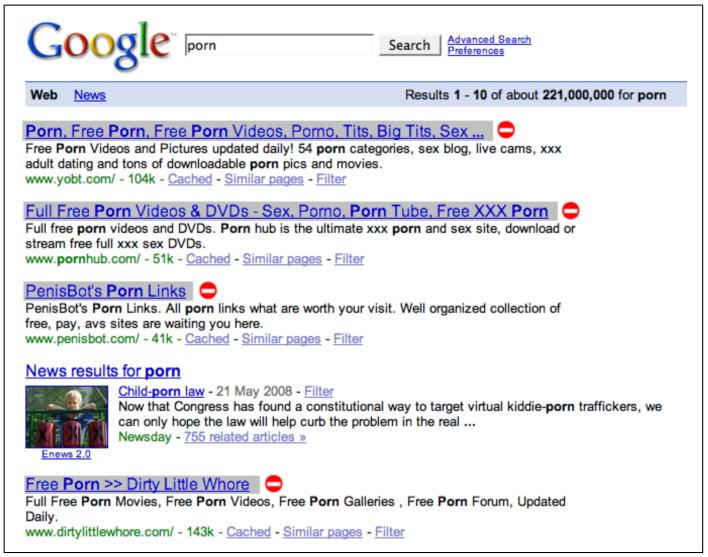


Figure 7.9: **Protecting users from objectionable search results.** In this screenshot, the search results of a query for "porn" are shown. Any links to pornographic Web documents have been flagged as potentially unwanted content according to the user's preferences. Non-pornographic documents, on the other hand, are not affected as can be seen for the news document about "Child-porn law" at position 4.

7.5 Summary

In this chapter, we have proposed a new approach to Web filtering by exploiting the concepts of folksonomies and collaborative tagging. We have presented a case study of a working prototype, *TaggyBear*, to demonstrate how such an approach can be implemented in practice, and have described and evaluated its system design and anatomy. Our experiments and performance analyses have shown that TaggyBear can meet the system requirements for Web filtering. Additionally, the use of such a system is not restricted to the filtering of Web resources as it also provides the normal features of a collaborative tagging system due to the close relation between tagging and rating Web resources. We have also outlined how the system can be combined with our techniques and results from Chapters 5 (Expertise Ranking) and 6 (Web Search Personalization) for increased effect. Hence, our results support our hypothesis that the concepts of folksonomies and collaborative tagging can be exploited for user-driven filtering of the Web.

With this chapter, we have finished the second part of this thesis, where we have focused our studies on leveraging folksonomies for Web information retrieval. In the last part, we will give our conclusions of this thesis based on the analyses and findings described in this and the previous chapters. We will also discuss the implications and significance of our research work, and outline possible future research directions.

Part III

Conclusions and Future Work

Now it is quite clear to me that there are no solid spheres in the heavens, and those that have been devised by the authors to save the appearances, exist only in the imagination.

Tycho Brahe (1546–1601)

8 Conclusions and Key Results

The overall goal of this thesis was to analyze Web users and their contributed data in order to improve the retrieval of information on the Web. The motivation came from the recent evolution of the Web itself, which has been increasingly used for social interactions and user collaboration on both local and global dimensions. This development has been coined the trend of the Social Web, where online services and Web-enabled applications facilitate and stimulate user interactions, and where a surge of user-contributed data such as articles, photos or videos can be observed. In this thesis, we have studied social interactions and user collaboration on the Social Web at the example of folksonomies for the purpose of information retrieval. Folksonomies are suitable subjects for such a study because they represent a very general form of interactions on the Web by involving the acting subjects on the Web (users), the objects containing information on the Web (Web resources), and metadata about these (tags). The main objectives of this thesis were to deepen our understanding of the characteristics, dynamics and hidden semantics of folksonomies, and to explore how this knowledge can be leveraged to enhance and improve techniques in the research area of Web information retrieval. On the one hand, we have analyzed and related folksonomies to other types of Web data and metadata and to Web information retrieval in general. On the other hand, we have demonstrated how folksonomies can be exploited in practice by presenting and evaluating new approaches for three different problem scenarios, namely expertise ranking, personalization of Web search, and Web filtering.

The first part of the thesis started in Chapter 2 with a thorough review of state-of-the-art research on folksonomies and collaborative tagging. While the concept of collaborative tagging is similar to traditional subject indexing, we showed that it offers users much more flexibility and benefits for annotating resources, and that it also provides individual users with incentives to perform these activities collaboratively. We found studies in the literature to span a wide range of topics, including analyses of user motivations and functions of tags, the dynamics and usage patterns of folksonomies, creating recommender systems based on collaborative tagging, and the phenomenon and impact of spam in folksonomies. While there has been a plethora of research on folksonomies in general, there is a lack of in-depth study of folksonomies in the domain of Web information retrieval. To conduct such research was the central goal of this thesis.

After presenting in Chapter 3 the major data sources and main experimental data sets used for the work described in this thesis, we described in Chapter 4 our empirical

study of the characteristics of folksonomies in the context of Web information retrieval. We investigated how much and what kind of folksonomy data is available in practice, and how it compares and relates to other types of data and metadata on the Web such as the content of Web resources, and metadata provided by the authors of these resources. We found that folksonomies provide large volumes of (meta)data about Web resources and that they already cover a considerable fraction of the Web. We also observed that folksonomies provide new data that is not available through content inspection or link analysis of Web resources. Additionally, information in folksonomies was found to be different from other Web metadata types, and seemed to to be particularly suited for classification tasks in Web information retrieval. These results supported our hypothesis (Hypothesis 1) that user-contributed data in folksonomies provides new, complimentary information about Web resources that is not available through traditional types of data and metadata on the Web.

In the second part of the thesis, we turned our attention to leveraging the knowledge and experimental results presented in the first part for enhancing and improving techniques in the domain of Web information retrieval. We investigated the notion of expertise or "trustworthiness" of users in folksonomies in Chapter 5. Identifying experts among users has benefits for a wide range of applications such as recommender systems, resource discovery and social network analysis. We introduced the notion of implicit endorsement between users in folksonomies via an analysis of the temporal dimension of user activity, and conceived two assumptions of experts: Firstly, there exists a relationship of mutual reinforcements between the expertise of users and the quality of Web resources. Secondly, experts are also the discoverers of high quality documents. Based on these ideas, we proposed a graph-based algorithm, *SPEAR*, for ranking users according to their expertise. We evaluated SPEAR with experiments based on large sets of real-world data, and found the algorithm to be effective at identifying expert users and, at the same time, reducing the negative impact of malicious users in a folksonomy. These results supported our hypothesis (Hypothesis 2) that the expertise or trustworthiness of users in a folksonomy can be derived from an analysis of their activity and implicit interactions. We thus showed that an appropriate method such as SPEAR is able to gain a better understanding of the characteristics of users by analyzing their collective behavior in folksonomies.

In Chapter 6, we applied folksonomies to the scenario of Web search and investigated how they can be used for tailoring search results according to the individual interests of users. We showed that both the topics of Web resources and the interests of users can be derived from folksonomies, and how this information can be used to construct profiles of users and resources in a multidimensional topic space. We proposed a new approach to the personalization of Web search personalization, which is based on the re-ranking of search results as returned by a traditional search engine according to the similarity between the profiles of the user and the Web resources. At the example of the search engine Google and the collaborative tagging system Delicious, we demonstrated how this personalization approach can be implemented in practice. We also evaluated the approach with quantitative and qualitative analyses, and found that it is feasible in practice with regard to the availability of sufficient volumes of folksonomy

data about Web resources in the scenario of Web search, and that users have perceived an improvement in the quality of search results compared to the evaluation baseline. These results supported our hypothesis (Hypothesis 3) that folksonomies provide sufficiently rich information about users and Web resources to allow for the personalization of Web search.

Lastly, we explored in Chapter 7 how the concepts of collaborative tagging and folksonomies can be exploited for Web filtering. We described how the concept of tagging can be extended to allow for collaborative rating of resources on the Web, and proposed a folksonomy-driven alternative to traditional Web filtering approaches. We presented a case study of a working prototype, *TaggyBear*, to demonstrate how such the proposed approach can be implemented in practice, and described and evaluated its system design and anatomy. Our experiments and performance analyses showed that the TaggyBear system can meet the requirements for the scenario Web filtering. Additionally, we also discussed how the system can benefit from and be combined with our techniques and results from Chapters 5 (Expertise Ranking) and 6 (Web Search Personalization) for increased effect. These results supported our hypothesis (Hypothesis 4) that the concepts of folksonomies and collaborative tagging can be exploited for user-driven filtering of the Web.

While we have presented these studies separately in this thesis, we like to emphasize that they are closely related to and benefit from each other. Folksonomy-driven Web filtering, for instance, can be used to improve Web search by protecting users from dangerous or unwanted Web resources in search results. Finding experts users, on the other hand, can be helpful to any technique that leverages user-contributed data, of which Web search personalization and Web filtering are but two examples. Similarly, the proposed re-ranking technique for the personalization of Web search can also be applied to other scenarios. For instance, it can be employed to personalize the quality-based ranking of resources within a folksonomy, which is another outcome of our proposed *SPEAR* algorithm.

Our research work described in this thesis presented a thorough investigation of folksonomies and collaborative tagging in the context of information retrieval on the Web. We demonstrated that the characteristics and qualities of users and Web resources can be understood by analyzing their implicit interactions in folksonomies, and how this knowledge can subsequently be exploited to improve information retrieval on the Web. However, the scientific study of these phenomena is certainly not completed with the conclusion of this thesis. We believe that our research work, while having succeeded in answering some important questions, has opened up many possibilities for future research with respect to social interactions and user collaboration on the Web. In the next section, we will outline possible future research directions for folksonomies and the Social Web in general.

We can only see a short distance ahead, but we can see plenty there that needs to be done.

Alan Turing (1912–1954)

9 Future Research Directions

The Web is not purely a product of technologies but also a social phenomenon. Since its inception two decades ago, it has been exhibiting rapid growth and evolution to such an extent that it deserves better understanding beyond its technical aspects. The trend of the Social Web is one of its recent developments, which has been attracting the attention of a wide range of academic disciplines including psychology, economics, social sciences, law, and of course computer science. Folksonomies, which we have studied in this thesis, belong to the prominent and popular features of the Social Web. Like the Web itself, folksonomies have been shown to feature a simple, local action that with increasing scale and usage eventually leads to a huge, complex network structure. They are the result of the social interactions and collective behavior of human users in the virtual world of the Web and the Internet.

We believe that one direction for future research on the Social Web is the analysis of the temporal dimension of user activities and, on a larger scale, the interrelations between the real world and the virtual world of the Web. Since the Social Web is driven by human users, it means that experiences and events in the real world will also influence user behavior on the Web, and vice versa. It has been discovered, for example, that examining the search queries of users on the Web allows for the prediction of flu outbreaks[1]. Web applications such as the micro-blogging service Twitter have been used to spread the word about recent events such as airplane crashes or terrorist attacks faster than traditional news media. Similarly, we have shown in this thesis how deriving implicit interactions from temporal information of user activities can be used to improve the quality and robustness of techniques in Web information retrieval. Pursuing studies such as those described above could therefore help to gain insights into both the real world and the virtual world.

Users of the Social Web are producing huge amounts of data every day, which contain information about nearly every aspect of the users and their lifes. However, this data is often retained within isolated Web applications (e.g. photos on Flickr, videos on YouTube, contacts and social networks on Facebook). The consequence is, for example, that a video published by a user's friend on YouTube is not associated with his contact entry on Facebook. We therefore believe that finding a remedy for these data islands is another direction for future work. Research and development in areas such as the Semantic Web [BLHL01] will help to establish a universal infrastructure that facilitates

[1] Google Flu Trends, http://www.google.org/flutrends/, last retrieved on March 01, 2010.

the portability and interoperability of data across different domains. In a similar context, we argue that it is worth further research to study how user-contributed data on the Social Web can be leveraged outside its original domain. These studies include investigating the correlations between different folksonomies, how collaborative tagging systems influence or are influenced by other applications on the Web (e.g. whether a Web resource becomes popular on Web search first, and only then in folksonomies), or how folksonomies can be translated to ontologies and vice versa. For example, we have shown in this thesis how the data collected by collaborative tagging systems can be exploited to improve Web search. Similarly, our proposed expertise ranking algorithm SPEAR can also be applied to non-folksonomy problems such as citation networks of academic publications in order to measure the expertise of researchers. We therefore believe that such studies would yield benefits in a wide range of areas.

Lastly, the popularity and success of folksonomies in practice have been attributed to their ease of use and the freedom they give to users when annotating resources. The opposite approach are top-down schemes such as ontologies and taxonomies, of which a typical example is the *Dewey Decimal Classification system* [OCL] for cataloging books in libraries. Of course, each approach comes with its own advantages and disadvantages. On a larger scale, an interesting research question is that of the balance between bottom-up, uncontrolled social interactions and those that are top-down and controlled. For instance, studies such as [Gil05] have analyzed and compared the quality of the "bottom-up" Wikipedia and the "top-down" Encyclopedia Britannica. While the results have so far not reached a definite conclusion, an interesting development was that, on the one hand, Wikipedia has recently introduced moderated articles and, on the other hand, Encyclopedia Britannica has been opening itself to end user contributions. Studies that investigate and compare the dynamics and effects of bottom-up and top-down approaches could therefore improve our knowledge on how to properly understand and influence collective user behavior, and how to optimize its use with regard to a specific problem scenario.

Part IV

Appendix

A

Example of an ICRA Content Rating in RDF Format

The following listing shows an exemplary ICRA content rating in RDF format. The corresponding HTML snippet is described in Section 7.1.1.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:label="http://www.w3.org/2004/12/q/contentlabel#"
  xmlns:icra="http://www.icra.org/rdfs/vocabularyv03#">

  <rdf:Description rdf:about="">
    <dc:creator rdf:resource="http://www.icra.org" />
    <dcterms:issued>2009-12-15</dcterms:issued>
    <label:authorityFor>http://www.icra.org/rdfs/vocabularyv03#</
        label:authorityFor>
  </rdf:Description>

  <label:Ruleset>
    <label:hasHostRestrictions>
      <label:Hosts>
        <label:hostRestriction>example.com</label:hostRestriction>
      </label:Hosts>
    </label:hasHostRestrictions>
    <label:hasDefaultLabel rdf:resource="#label_1" />
  </label:Ruleset>

  <label:ContentLabel rdf:ID="label_1">
    <rdfs:comment>Label for all/most of website</rdfs:comment>
    <icra:na>1</icra:na>
    <icra:nb>1</icra:nb>
    <icra:nc>1</icra:nc>
    <icra:sz>0</icra:sz>
    <icra:vz>0</icra:vz>
    <icra:lz>0</icra:lz>
```

```
   <icra:oz>0</icra:oz>
   <icra:cz>0</icra:cz>
   <rdfs:label>Exposed breasts; Bare buttocks; Visible genitals;
      Sexual material may be, but is not known to be, present;
      Violence may be, but is not known to be, present; Potentially
      offensive language may be, but is not known to be, present;
      Potentially harmful activities may be, but are not known to be,
       depicted; User generated content may be, but is not known to
      be, present; </rdfs:label>
  </label:ContentLabel>

</rdf:RDF>
```

Listing A.1: **An exemplary ICRA content rating.**
In this example, the Web site http://www.example.com/ is rated as depicting exposed breasts, bare buttocks and visible genitals. Other types of objectionable content may be, but are not known to be, present.

Bibliography

[ABB+09] Fabian Abel, Matteo Baldoni, Cristina Baroglio, Nicola Henze, Daniel Krause, and Viviana Patti. Context-based ranking in folksonomies. In *HT '09: Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 209–218, New York, NY, USA, 2009. ACM.

[ABC98] David Abrams, Ron Baecker, and Mark Chignell. Information archiving with bookmarks: Personal web space construction and organization. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 41–48, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[AGS07] Ching-man Au Yeung, Nicolas Gibbins, and Nigel Shadbolt. Understanding the semantics of ambiguous tags in folksonomies. In *The International Workshop on Emergent Semantics and Ontology Evolution (ESOE2007) at ISWC/ASWC 2007*, pages 108–121, November 2007.

[AGS08a] Ching-man Au Yeung, Nicholas Gibbins, and Nigel Shadbolt. Web search disambiguation by collaborative tagging. In 2008, editor, *Proceedings of the Workshop on Exploring Semantic Annotations in Information Retrieval (ESAIR) at ECIR'08*, pages 48–61, 2008.

[AGS08b] Ching-man Au Yeung, Nicolas Gibbins, and Nigel Shadbolt. Discovering and modelling multiple interests of users in collaborative tagging systems. In *WI-IAT '08: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 115–118, Washington, DC, USA, 2008. IEEE Computer Society.

[AKCS00] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, and Constantine D. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167, New York, NY, USA, 2000. ACM.

[AKD07] Hend S. Al-Khalifa and Hugh C. Davis. Exploring the value of folksonomies for creating semantic metadata. *International Journal on Semantic Web and Information Systems*, 3(1):13–39, 2007.

[AKTV07] Anupriya Ankolekar, Markus Krötzsch, Thanh Tran, and Denny Vrandecic. The two cultures: Mashing up web 2.0 and the semantic web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 825–834, New York, NY, USA, 2007. ACM.

[AMR08] Arun Kumar Agrahri, Divya Anand Thattandi Manickam, and John Riedl. Can people collaborate to improve the relevance of search results? In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 283–286, New York, NY, USA, 2008. ACM.

[AN07] Morgan Ames and Mor Naaman. Why we tag: Motivations for annotation in mobile and online media. In *CHI '07: Proceedings of the SIGCHI conference on Human Factors in computing systems*, New York, NY, USA, 2007. ACM Press.

[ANG+09] Ching-man Au Yeung, Michael G. Noll, Nicholas Gibbins, Christoph Meinel, and Nigel Shadbolt. On measuring expertise in collaborative tagging systems. In *Proceedings of the WebSci'09: Society On-Line*, March 2009.

[ANG+10] Ching-man Au Yeung, Michael G. Noll, Nicholas Gibbins, Christoph Meinel, and Nigel Shadbolt. Spear: Spamming-resistant expertise analysis and ranking in collaborative tagging systems. *International Journal of Computational Intelligence*, 2010. to appear.

[Arr06] Michael Arrington. Aol: 'this was a screw up'. http://techcrunch.com/2006/08/07/aol-this-was-a-screw-up/, August 2006. Last retrieved on December 01, 2009.

[Arr09] Michael Arrington. Facebook now nearly twice the size of myspace worldwide. http://techcrunch.com/2009/01/22/facebook-now-nearly-twice-the-size-of-myspace-worldwide/, January 2009. Last retrieved on December 01, 2009.

[AS08] Josh Attenberg and Torsten Suel. Cleaning search results using term distance features. In *AIRWeb '08: Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 21–24, New York, NY, USA, 2008. ACM.

[ATD08] Eytan Adar, Jaime Teevan, and Susan T. Dumais. Large scale analysis of web revisitation patterns. In *CHI '08: Proceeding of the 26th annual SIGCHI conference on Human factors in computing systems*, pages 1197–1206, New York, NY, USA, 2008. ACM.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[BAJ00] Albert-László Barabási, Réka Albert, and Hawoong Jeong. Scale-free characteristics of random networks: The topology of the world wide web. *Physica A: Statistical Mechanics and its Applications*, 281(1–4):69–77, 2000.

[Bal08] Eric Baldeschwieler. Yahoo! launches world's largest hadoop production application. http://developer.yahoo.net/blogs/hadoop/2008/02/yahoo-worlds-largest-production-hadoop.html, February 2008. Last retrieved on March 01, 2010.

[Bar05] Albert-Lázló Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[BCSU05] András A. Benczúr, Károly Csalogány, Tamás Sarlós, and Máté Uher. Spamrank: Fully automatic link spam detection work. In *AIRWeb '05: Proceedings of the 1st international workshop on Adversarial information retrieval on the web*, 2005.

[BFNP08] Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. Can all tags be used for search? In *CIKM '08: Proceedings of the 17th ACM conference on Information and Knowledge Management*, pages 193–202, New York, NY, USA, 2008. ACM.

[BH09] Dirk Bollen and Harry Halpin. The role of tag suggestions in folksonomies. In *HT '09: Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 359–360, New York, NY, USA, 2009. ACM.

[BHK+09] Eda Baykan, Monika Rauch Henzinger, Stefan F. Keller, Sebastian De Castelberg, and Markus Kinzler. A comparison of techniques for sampling web pages. In Susanne Albers and Jean-Yves Marion, editors, *STACS*, volume 3 of *LIPIcs*, pages 13–30. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

[BJL+07] Steven M. Beitzel, Eric C. Jensen, David D. Lewis, Abdur Chowdhury, and Ophir Frieder. Automatic classification of web queries using very large unlabeled query logs. *ACM Trans. Inf. Syst.*, 25(2):9, 2007.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[BM02] Krishna Bharat and George A. Mihaila. When experts agree: using non-affiliated experts to rank popular topics. *ACM Trans. Inf. Syst.*, 20(1):47–58, 2002.

[BP98] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[Bro02] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[BRRT05] Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. Link analysis ranking: algorithms, theory, and experiments. *ACM Transactions on Internet Technology*, 5(1):231–297, 2005.

[BS08] Somnath Banerjee and Martin Scholz. Leveraging web 2.0 sources for web content classification. *WI/IAT '08: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 1:300–306, 2008.

[BXW+07] Shenghua Bao, Guirong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 501–510, New York, NY, USA, 2007. ACM Press.

[BYG06] Ziv Bar-Yossef and Maxim Gurevich. Random sampling from a search engine's index. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 367–376, New York, NY, USA, 2006. ACM Press.

[BYT07] Ricardo Baeza-Yates and Alessandro Tiberi. Extracting semantic relations from query logs. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 76–85, New York, NY, USA, 2007. ACM.

[CBHS08] Ciro Cattuto, Dominik Benz, Andreas Hotho, and Gerd Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In *ISWC '08: Proceedings of the 7th International Conference on The Semantic Web*, pages 615–631, Berlin, Heidelberg, 2008. Springer-Verlag.

[CDG+07] Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Murdock, and Fabrizio Silvestri. Know your neighbors: web spam detection using the web topology. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 423–430, New York, NY, USA, 2007. ACM.

[CDI98] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 307–318, New York, NY, USA, 1998. ACM.

[CFL09] Federica Cena, Rosta Farzan, and Pasquale Lops. Web 3.0: Merging semantic web with social web. In *HT '09: Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 385–386, New York, NY, USA, 2009. ACM.

[CFN07] Paul-Alexandru Chirita, Claudiu S. Firan, and Wolfgang Nejdl. Personalized query expansion for the web. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 7–14, New York, NY, USA, 2007. ACM.

[Cha03] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, Amsterdam, 2003.

[Chi06] Michelene T. H. Chi. *Two Approaches to the Study of Experts' Characteristics*, pages 21–30. Volume 1 of Ericsson et al. [ECFH06], 2006.

[Chr06] Stijn Christiaens. Metadata mechanisms: From ontology to folksonomy ... and back. In *On the Move to Meaningful Internet Systems Workshop*, LNCS, pages 199–207, 2006.

[CLP07] Ciro Cattuto, Vittorio Loreto, and Luciano Pietronero. Semiotic dynamics and collaborative tagging. *PNAS: Proceedings of the National Academy of Sciences of the United States of America*, 104(5):1461–1464, January 2007.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2001.

[CLW08] James Caverlee, Ling Liu, and Steve Webb. Socialtrust: tamper-resilient trust establishment in online communities. In *JCDL '08: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 104–114, New York, NY, USA, 2008. ACM.

[CM08] Ed H. Chi and Todd Mytkowicz. Understanding the efficiency of social tagging systems using information theory. In *HT '08: Proceedings of the 19th ACM conference on Hypertext and hypermedia*, pages 81–88, New York, NY, USA, 2008. ACM.

[CNPK05] Paul Alexandru Chirita, Wolfgang Nejdl, Raluca Paiu, and Christian Kohlschütter. Using odp metadata to personalize search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, New York, NY, USA, 2005. ACM.

[CNZ05] Paul-Alexandru Chirita, Wolfgang Nejdl, and Cristian Zamfir. Preventing shilling attacks in online recommender systems. In *WIDM '05: Proceedings of the 7th annual ACM international workshop on Web information and data management*, pages 67–74, New York, NY, USA, 2005. ACM.

[Coo06] Frederick L. Coolidge. *Statistics: A Gentle Introduction*. SAGE Publications, 2nd edition, 2006.

[Cor97] Georgia Tech Research Corporation. Gvu's 7th www user survey. http://www.cc.gatech.edu/gvu/user_surveys/survey-1997-04/, April 1997.

[Cor07] Gordon V. Cormack. Email spam filtering: A systematic review. *Found. Trends Inf. Retr.*, 1(4):335–455, 2007.

[CR87] John M. Carroll and Mary Beth Rosson. *Paradox of the Active User*, chapter 5, pages 80–111. MIT Press, Cambridge, MA, USA, 1987.

[CSB+07] Ciro Cattuto, Christoph Schmitz, Andrea Baldassarri, Vito D. P. Servedio, Vittorio Loreto, Andreas Hotho, Miranda Grahl, and Gerd Stumme. Network properties of folksonomies. *AI Communications Journal, Special*

Issue on Network Analysis in Natural Sciences and Engineering, 20(4):245–262, 2007.

[CSN09] Aaron Clauset, Cosma R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

[Cut09] Matt Cutts. Google does not use the keywords meta tag in web ranking. http://googlewebmastercentral.blogspot.com/2009/09/google-does-not-use-keywords-meta-tag.html, September 2009. Last retrieved on December 01, 2009.

[Cza09] Grzegorz Czajkowski. Large-scale graph computing at google. http://googleresearch.blogspot.com/2009/06/large-scale-graph-computing-at-google.html, June 2009. Last retrieved on December 01, 2009.

[DDL+90] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[DG04] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *OSDI'04: 6th Symposium on Operating System Design and Implementation*, 2004.

[DHS01] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2001.

[DMQU10] Pasquale De Meo, Giovanni Quattrone, and Domenico Ursino. A query expansion and user profile enrichment approach to improve the performance of recommender systems operating on a folksonomy. *User Modeling and User-Adapted Interaction*, 20(1):41–86, 2010.

[Dou02] John R. Douceur. The sybil attack. In *IPTPS '01: Revised Papers from the 1st International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag.

[DPN08] Thomas Deselaers, Lexi Pimenidis, and Hermann Ney. Bag-of-visual-words models for adult image classification and filtering. In *ICPR '08: Proceedings of 19th International Conference on Pattern Recognition*, 2008.

[DPRZ08] Ronald J. Deibert, John G. Palfrey, Rafal Rohozinski, and Jonathan Zittrain. *Access Denied: The Practice and Policy of Global Internet Filtering*. The MIT Press, Cambridge, Massachusetts, USA, 2008.

[DS08] Klaas Dellschaft and Steffen Staab. An epistemic dynamic model for tagging systems. In *HT '08: Proceedings of the 19th ACM conference on Hypertext and Hypermedia*, pages 71–80, New York, NY, USA, 2008. ACM.

[DSW07] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 581–590, New York, NY, USA, 2007. ACM.

[DWV99] Harris Drucker, Donghui Wu, and Vladimir N. Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

[ECFH06] K. Anders Ericsson, Neil Charness, Paul J. Feltovich, and Robert R. Hoffman, editors. *The Cambridge Handbook of Expertise and Expert Performance*, volume 1. Cambridge University Press, USA, 2006.

[EM03] Nadav Eiron and Kevin S. McCurley. Analysis of anchor text for web search. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 459–460, New York, NY, USA, 2003. ACM.

[Fö99] Johannes Fürnkranz. Exploiting structural information for text classification on the www. In *IDA '99: Proceedings of the 3rd International Symposium on Advances in Intelligent Data Analysis*, pages 487–498, London, UK, 1999. Springer-Verlag.

[FD97] Larry Fitzpatrick and Mei Dent. Automatic feedback using past queries: social searching? *Special Issue of the SIGIR Forum*, 31(SI):306–313, 1997.

[FFB96] Margaret Fleck, David Forsyth, and Chris Bregler. Finding naked people. In *European Conference on Computer Vision*, volume 2, pages 592–602, 1996.

[FGM+99] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Rfc 2616: Hypertext transfer protocol – http/1.1. http://www.w3.org/Protocols/rfc2616/rfc2616.html, 1999. Network Working Group, W3. Retrieved on 01 December 2009.

[Fie00] Roy T. Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, CA, USA, 2000.

[FLM+06] Ayman Farahat, Thomas LoFaro, Joel C. Miller, Gregory Rae, and Lesley A. Ward. Authority rankings from hits, pagerank, and salsa: Existence, uniqueness, and effect of initialization. *SIAM J. Sci. Comput.*, 27(4):1181–1201, 2006.

[Fou10] Apache Software Foundation. Hadoop map/reduce tutorial. http://hadoop.apache.org/common/docs/current/mapred_tutorial.html, February 2010.

[FPE06] P. J. Feltovich, M. J. Prietula, and K. Anders Ericsson. *Studies of Expertise from Psychological Perspectives*, pages 41–68. Volume 1 of Ericsson et al. [ECFH06], 2006.

[GGM05] Zoltán Gyöngyi and Hector Garcia-Molina. Web spam taxonomy. In *AIRWeb '05: Proceedings of the 1st international workshop on Adversarial information retrieval on the web*, 2005.

[GGMP04] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with trustrank. In *VLDB '04: Proceedings of the 13th international conference on Very large data bases*, pages 576–587. VLDB Endowment, 2004.

[GH06] Scott Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[Gil05] Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, December 2005.

[Gru95] Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *International Journal on Human-Computer Studies*, 43(5–6):907–928, 1995.

[Gru07] Thomas Gruber. Ontology of folksonomy: A mash-up of apples and oranges. *International Journal on Semantic Web & Information Systems*, 3(2), 2007.

[GS07] Qingqing Gan and Torsten Suel. Improving web spam classifiers using link structure. In *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 17–20, New York, NY, USA, 2007. ACM.

[GSF02] W. I. Grosky, D. V. Sreenath, and F. Fotouhi. Emergent semantics and the multimedia semantic web. *SIGMOD Rec.*, 31(4):54–58, 2002.

[GW99] Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.

[Hav02] Taher H. Haveliwala. Topic-sensitive pagerank. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 517–526, New York, NY, USA, 2002. ACM.

[Hav03] Taher H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Trans. on Knowl. and Data Eng.*, 15(4):784–796, 2003.

[Hen06] Cal Henderson. *Building Scalable Websites*. O'Reilly Media, Sebastopol, CA, USA, 2006.

[HHLS05] Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social bookmarking tools (i): A general review. *D-Lib Magazine*, 11(4), 2005.

[Hic05] Ian Hickson. Google: Web authoring statistics. Technical report, Google, Inc., December 2005. Last retrieved on December 01, 2009.

[Hid02] José María Gómez Hidalgo. Evaluating cost-sensitive unsolicited bulk email categorization. In *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, pages 615–620, New York, NY, USA, 2002. ACM.

[Hir10] Mikio Hirabayashi. Fundamental specifications of tokyo cabinet version 1. http://1978th.net/tokyocabinet/spex-en.html, January 2010. Last retrieved on March 01, 2010.

[HJSS06a] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Bibsonomy: A social bookmark and publication sharing system. In ldo de Moor, Simon Polovina, and Harry Delugach, editors, *Proceedings of the 1st Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, pages 87–102. Aalborg Universitetsforlag, 2006.

[HJSS06b] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Emergent semantics in bibsonomy. In Christian Hochberger and Rüdiger Liskowsky, editors, *Informatik 2006 - Informatik für Menschen. Band 2*, volume P-94 of *Lecture Notes in Informatics*. Gesellschaft für Informatik, 2006. Proceedings of Workshop on Applications of Semantic Technologies, Informatik 2006.

[HJSS06c] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference (ESWC 2006)*, volume 4011 of *LNCS*, pages 411–426. Springer, June 2006.

[HJSS06d] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Trend detection in folksonomies. In *SAMT '06: Proceedings of the 1st International Conference on Semantic and Digital Media Technologies*, volume 4306 of *LNCS*, pages 56–70, Berlin/Heidelberg, December 2006. Springer.

[HKGM07] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11(6):36–45, 2007.

[HKGM08] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search? In *Proceedings of 1st ACM International Conference on Web Search and Data Mining (WSDM'08)*, pages 195–206. ACM, February 2008.

[HKTR04] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[Hoo65] R. S. Hooper. Indexer consistency tests: Origin, measurements, results and utilization. In *IBM Corporation*, 1965.

[HPS08] Xin Hu, Taejoon Park, and Kang G. Shin. Attack-tolerant time-synchronization in wireless sensor networks. In *IEEE INFOCOM: Proceedings of 27th Conference on Computer Communications*, pages 448–456, April 2008.

[HRS07] Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 211–220, New York, NY, USA, 2007. ACM.

[IBM09] IBM. Testing the 'typical bi day': Scalability testing of an ibm cognos 8 bi enterprise deployment. White paper, IBM, 2009.

[JKHS08] Robert Jäschke, Beate Krause, Andreas Hotho, and Gerd Stumme. Logsonomy - a search engine folksonomy. In *ICWSM '08: 2nd International Conference on Weblogs and Social Media*. AAAI Press, 2008.

[JKPT07] Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. I know what you did last summer: Query logs and user privacy. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 909–914, New York, NY, USA, 2007. ACM.

[JKR+99] Varghese Jacob, Ramayya Krishnan, Young U. Ryu, R. Chandrasekaran, and Sungchul Hong. Filtering objectionable internet content. In *ICIS '99: Proceedings of the 20th international conference on Information Systems*, pages 274–278, Atlanta, GA, USA, 1999. Association for Information Systems.

[JP01] Bernard J. Jansen and Udo Pooch. A review of web searching studies and a framework for future research. *Journal of the American Society for Information Science and Technology*, 52(3):235–246, 2001.

[JR02] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[JRMG06] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. Generating query substitutions. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 387–396, New York, NY, USA, 2006. ACM.

[JS06] Ajita John and Doree Seligmann. Collaborative tagging and expertise in the enterprise. In *Workshop on Collaborative Web Tagging at WWW 2006*, 2006.

[JUB09] Christian Jansohn, Adrian Ulges, and Thomas M. Breuel. Detecting pornographic video content by combining image features with motion information. In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, pages 601–604, New York, NY, USA, 2009. ACM.

[JW03] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 271–279, New York, NY, USA, 2003. ACM.

[Kan04] Min-Yen Kan. Web page categorization without the web page. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 262–263, New York, NY, USA, 2004. ACM Press.

[KEG+07] Georgia Koutrika, Frans Adjie Effendi, Zoltán Gyöngyi, Paul Heymann, and Hector Garcia-Molina. Combating spam in tagging systems. In *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 57–64, New York, NY, USA, 2007. ACM.

[KFG+07] Georgia Koutrika, Effendi Frans, Zoltán Gyöngyi, Paul Heymann, and Hector Garcia-Molina. Combating spam in tagging systems: An evaluation. *ACM Transactions on the Web (TWEB)*, 2(4):1–34, 2007.

[KFJ06] Pranam Kolari, Tim Finin, and Anupam Joshi. Svms for the blogosphere: Blog identification and splog detection. In *Proceedings of AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, 2006.

[KHS08] Beate Krause, Andreas Hotho, and Gerd Stumme. A comparison of social bookmarking with traditional search. In *ECIR '08: Proceedings of the 30th European Conference on IR Research*, volume 4956 of *LNCS*, pages 101–113. Springer, 2008.

[Kip08] Margaret E.I. Kipp. Toread and cool: Subjective, affective and associative factors in tagging. In *CAIS '08: Proceedings of the 36th Annual Conference of the Canadian Association for Information Science*, June 2008.

[KJF+06] Pranam Kolari, Akshay Java, Tim Finin, Tim Oates, and Anupam Joshi. Detecting spam blogs: A machine learning approach. In *AAAI '06: Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[KJHS08] Beate Krause, Robert Jäschke, Andreas Hotho, and Gerd Stumme. Logsonomy - social information retrieval with logdata. In *HT '08: Proceedings of the 19th ACM conference on Hypertext and Hypermedia*, pages 157–166, New York, NY, USA, 2008. ACM.

[KK01] José Kahan and Marja-Ritta Koivunen. Annotea: an open rdf infrastructure for shared web annotations. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 623–632, New York, NY, USA, 2001. ACM.

[KLC06] Yung-Ming Kuo, Jiann-Shu Lee, and Pau-Choo Chung. The naked image detection based on automatic white balance method. In *Proceedings of the 2006 International Computer Symposium (ICS) on Image Processing, Computer Graphics and Multimedia Technologies*, pages 990–994, Taipei, Taiwan, 2006.

[Kle98] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA '98: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 668–677, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[Kne06] Torben Knerr. Tagging ontology - towards a common ontology for folksonomies. http://tagont.googlecode.com/files/TagOntPaper.pdf, 2006.

[KRR+00] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tompkins, and Eli Upfal. The web as a graph. In *PODS '00: Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–10, New York, NY, USA, 2000. ACM.

[KS10] Christian Körner and Markus Strohmaier. A call for social tagging datasets. *SIGWEB Newsletter*, Issue Winter(Winter):1–6, 2010.

[KSB+08] Hak Lae Kim, Simon Scerri, John G. Breslin, Stefan Decker, and Hong Gee Kim. The state of the art in tag ontologies: a semantic model for tagging and folksonomies. In *DCMI '08: Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications*, pages 128–137. Dublin Core Metadata Initiative, 2008.

[KSHS08] Beate Krause, Christoph Schmitz, Andreas Hotho, and Gerd Stumme. The anti-social tagger: detecting spam in social bookmarking systems. In *AIRWeb '08: Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 61–68, New York, NY, USA, 2008. ACM.

[KSKP03] Marja-Riitta Koivunen, Ralph Swick, Jose Kahan, and Eric Prud'hommeaux. An annotea bookmark schema. http://www.w3.org/2003/07/Annotea/BookmarkSchema-20030707, 2003. Retrieved on 01 July 2009.

[KT03] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.

[KZ04] Reiner Kraft and Jason Zien. Mining anchor text for query refinement. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 666–674, New York, NY, USA, 2004. ACM.

[Lan98] Frederick W. Lancaster. *Indexing and Abstracting in Theory and Practice*. University of Illinois, Graduate School of Library and Information Science, USA, 2 edition, 1998.

[Lan06] Scott Laningham. Ibm developerworks interviews: Tim berners-lee. http://www.ibm.com/developerworks/podcast/dwi/cm-int082206txt.html, August 2006.

[LBY+07] Rui Li, Shenghua Bao, Yong Yu, Ben Fei, and Zhong Su. Towards effective browsing of large scale social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 943–952, New York, NY, USA, 2007. ACM.

[Lev66] Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, February 1966.

[LGZ08] Xin Li, Lei Guo, and Yihong Eric Zhao. Tag-based social interest discovery. In *WWW '08: Proceedings of the 17th international conference on World Wide Web*, pages 675–684, New York, NY, USA, 2008. ACM.

[LKCC07] Jiann-Shu Lee, Yung-Ming Kuo, Pau-Choo Chung, and E-Liang Chen. Naked image detection based on adaptive and extensible skin color model. *Pattern Recogn.*, 40(8):2261–2270, 2007.

[LLC05] Uichin Lee, Zhenyu Liu, and Junghoo Cho. Automatic identification of user goals in web search. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 391–400, New York, NY, USA, 2005. ACM.

[LSP82] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[LV03] Peter Lyman and Hal R. Varian. How much information? www.sims.berkeley.edu/how-much-info-2003, 2003.

[LVC+99] Wen-Syan Li, Quoc Vu, Edward Chang, Divyakant Agrawal, Kyoji Hirata, Sougata Mukherjea, Yi-Leh Wu, Corey Bufi, Chen-Chuan Kevin Chang, Yoshinori Hara, Reiko Ito, Yutaka Kimura, Kezuyuki Shimazu, and Yukiyoshi Saito. Powerbookmarks: a system for personalizable web information organization, sharing, and management. In *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 565–567, New York, NY, USA, 1999. ACM.

[LW95] Fritz Lehmann and Rudolf Wille. A triadic approach to formal concept analysis. In *ICCS '95: Proceedings of the 3rd International Conference on Conceptual Structures*, pages 32–43, Berlin/Heidelberg, 1995. Springer.

[LYM02] Fang Liu, Clement Yu, and Weiyi Meng. Personalized web search by mapping user queries to categories. In *CIKM '02: Proceedings of the 11th international conference on Information and knowledge management*, pages 558–565, New York, NY, USA, 2002. ACM.

[Mat04] A. Mathes. Folksonomies - cooperative classification and communication through shared metadata. Technical report, University of Illinois Urbana-Champaign, USA, 2004.

[MC07] Elke Michlmayr and Steve Cayzer. Learning user profiles from tagging data and leveraging them for personal(ized) information access. In *Proceedings of the Workshop on Tagging and Metadata for Social Information Organization, 16th Int'l World Wide Web Conference (WWW 2007)*, 2007.

[MC08] Tyler Moore and Richard Clayton. Evaluating the wisdom of crowds in assessing phishing websites. In *FC '08: Proceedings of 12th International Conference on Financial Cryptography and Data Security (Revised Selected Papers)*, volume 5143 of *LNCS*, pages 16–30, Berlin, Heidelberg, 2008. Springer.

[McB94] Oliver A. McBryan. Genvl and wwww: Tools for taming the web. In *Proceedings of the 1st international conference on World Wide Web*, pages 79–90, 1994.

[MCM09a] Benjamin Markines, Ciro Cattuto, and Filippo Menczer. Social spam detection. In *AIRWeb '09: Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, pages 41–48, New York, NY, USA, 2009. ACM.

[MCM+09b] Benjamin Markines, Ciro Cattuto, Filippo Menczer, Dominik Benz, Andreas Hotho, and Gerd Stumme. Evaluating similarity measures for emergent semantics of social tagging. In *WWW '09: Proceedings of the 18th International Conference on World Wide Web*, pages 641–650, New York, NY, USA, 2009. ACM.

[Mer04] Peter Merholz. Metadata for the masses. http://www.adaptivepath.com/publications/essays/archives/000361.php, October 2004. Retrieved on 01 July 2009.

[Mik05] Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *The Semantic Web: Proceedings of 4th International Semantic Web Conference (ISWC)*, LNCS, pages 522–536, Heidelberg, 2005. Springer.

[Mik07] Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):5–15, 2007.

[MM06] George Macgregor and Emma McCulloch. Collaborative tagging as a knowledge organisation and resource discovery tool. *Library Review*, 55(5):291–300, February 2006.

[MNBD06] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM Press.

[MR08] Sergei Maslov and Sidney Redner. Promise and pitfalls of extending google's pagerank algorithm to citation networks. *The Journal of Neuroscience*, 28(44):11103–11105, October 2008.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[NA09] Michael G. Noll and Ching-man Au Yeung. How spear identifies domain experts within delicious. http://blog.delicious.com/blog/2009/08/how-spear-identifies-domain-experts-within-delicious.html, August 2009. Invited Article for Yahoo!, published on the official Delicious Web site; last retrieved on December 01, 2009.

[NAG+09] Michael G. Noll, Ching-man Au Yeung, Nicholas Gibbins, Christoph Meinel, and Nigel Shadbolt. Telling experts from spammers: Expertise ranking in folksonomies. In *SIGIR '09: Proceedings of the 32nd annual international ACM SIGIR conference on Research and Development in Information Retrieval*, July 2009.

[NDH06] Satoshi Niwa, Takuo Doi, and Shinichi Honiden. Web page recommender system based on folksonomy mining for itng '06 submissions. In *ITNG '06: Proceedings of the Third International Conference on Information Technology: New Generations*, pages 388–393, Washington, DC, USA, 2006. IEEE Computer Society.

[New06] M. E. J. Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):323–351, 2006.

[New09] NewScientist.com. Internet growth in 2008. http://www.newscientist.com/gallery/mg20227061900-exploring-the-exploding-internet/8, April 2009. Last retrieved on December 01, 2009.

[Nic97] David M. Nichols. Implicit ratings and filtering. In *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering*, pages 31–36, 1997.

[Nie99] Jakob Nielsen. *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, Thousand Oaks, CA, USA, 1999.

[NM05] Michael G. Noll and Christoph Meinel. Web page classification: An exploratory study of internet content rating systems. In *HACK '05: Proceedings of 1st International HACK Conference*, 2005.

[NM06] Michael G. Noll and Christoph Meinel. Design and anatomy of a social web filtering service. In *CIC '06: Proceedings of the 4th International Conference on Cooperative Internet Computing*, pages 35–44. World Scientific Publishing, 2006.

[NM07a] Michael G. Noll and Christoph Meinel. Authors vs. readers: A comparative study of document metadata and content in the www. In *DocEng '07: Proceedings of the 2007 ACM symposium on Document Engineering*, pages 177–186, New York, NY, USA, 2007. ACM.

[NM07b] Michael G. Noll and Christoph Meinel. Web search personalization via social bookmarking and tagging. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web: Proceedings of 6th International Semantic Web Conference (ISWC) and 2nd Asian Semantic Web Conference (ASWC)*, volume 4825 of *LNCS*, pages 367–380, Heidelberg, November 2007. Springer.

[NM08a] Michael G. Noll and Christoph Meinel. Building a scalable collaborative web filter with free and open source software. In *SITIS '08: IEEE International Conference on Signal Image Technology and Internet-Based Systems*, pages 563–571. IEEE Computer Society Press, 2008.

[NM08b] Michael G. Noll and Christoph Meinel. Exploring social annotations for web document classification. In *SAC '08: Proceedings of the 2008 ACM Symposium on Applied Computing*, pages 2315–2320, New York, NY, USA, 2008. ACM.

[NM08c] Michael G. Noll and Christoph Meinel. The metadata triumvirate: Social annotations, anchor texts and search queries. In *WI/IAT '08: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 640–647, Los Alamitos, CA, USA, 2008. IEEE Computer Society Press.

[NNMF06] Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 83–92, New York, NY, USA, 2006. ACM.

[NO09] Nicolas Neubauer and Klaus Obermayer. Hyperincident connected components of tagging networks. In *HT '09: Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 229–238, New York, NY, USA, 2009. ACM.

[Nol07a] Michael G. Noll. Running hadoop on ubuntu linux (multi-node cluster). http://www.michael-noll.com/wiki/Running_Hadoop_On_Ubuntu_Linux_(Multi-Node_Cluster), August 2007. Last retrieved on December 01, 2009.

[Nol07b] Michael G. Noll. Running hadoop on ubuntu linux (single-node cluster). http://www.michael-noll.com/wiki/Running_Hadoop_On_Ubuntu_Linux_(Single-Node_Cluster), August 2007. Last retrieved on December 01, 2009.

[Nol07c] Michael G. Noll. Writing an hadoop mapreduce program in python. http://www.michael-noll.com/wiki/Writing_An_Hadoop_MapReduce_Program_In_Python, September 2007. Last retrieved on December 01, 2009.

[Nol09] Michael G. Noll. Writing a personal link recommendation engine. *Python Magazine*, 3(2), 2009.

[NS08] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 111–125, Washington, DC, USA, 2008. IEEE Computer Society.

[NZT07] Marc A. Najork, Hugo Zaragoza, and Michael J. Taylor. Hits on the web: how does it compare? In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 471–478, New York, NY, USA, 2007. ACM.

[OCL] OCLC. Dewey decimal classification for use with oclc's online cataloging services. http://www.oclc.org/dewey/. Retrieved on 01 July 2009.

[OI05] OpenNet-Initiative. Internet filtering in china in 2004-2005: A country study. Technical report, OpenNet Initiative, 2005.

[O'R05] Tim O'Reilly. What is web 2.0: Design patterns and business models for the next generation of software. http://oreilly.com/web2/archive/what-is-web-20.html, September 2005.

[OS10] Kieron O'Hara and Nigel Shadbolt. Privacy on the data web. *Commun. ACM*, 53(3):39–41, 2010.

[otDoEECS09] Instructional Support Group of the Dept. of Electrical Engineering & Computer Science. Mapreduce and mpi. http://www-inst.eecs.berkeley.edu/usr/pub/mpi.help, August 2009. Last retrieved on December 01, 2009.

[PCT06] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*, New York, NY, USA, May 2006. ACM.

[Por80] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[PP07] John C. Paolillo and Shashikant Penumarthy. The social structure of tagging internet video on del.icio.us. In *HICSS '07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, pages 85–95, Washington, DC, USA, 2007. IEEE Computer Society.

[PSC+02] James Pitkow, Hinrich Schütze, Todd Cass, Rob Cooley, Don Turnbull, Andy Edmonds, Eytan Adar, and Thomas Breuel. Personalized search. *Communications of the ACM*, 45(9):50–55, 2002.

[QC06] Feng Qiu and Junghoo Cho. Automatic identification of user interest for personalized search. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 727–736, New York, NY, USA, 2006. ACM.

[Rad09] Filippo Radicchi. Human activity in the web. http://arxiv.org/abs/0903.2999, 2009.

[Rai07] Lee Rainie. 28 Technical report, PEW Internet & American Life Project, January 2007.

[Raj09] Shiva Rajaraman. Five stars dominate ratings. http://youtube-global.blogspot.com/2009/09/five-stars-dominate-ratings.html, September 2009. Last retrieved on March 01, 2010.

[RD06] Filip Radlinski and Susan Dumais. Improving personalized web search using result diversification. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 691–692, New York, NY, USA, 2006. ACM.

[Ree01] William J. Reed. The pareto, zipf and other power laws. *Economics Letters*, 74(1):15–19, 2001.

[RGMM07] A. W. Rivadeneira, Daniel M. Gruen, Michael J. Muller, and David R. Millen. Getting our head in the clouds: toward evaluation studies of tagclouds. In Mary Beth Rosson and David J. Gilmore, editors, *Proceedings of CHI*, pages 995–998. ACM, 2007.

[Ric95] John A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, Belmont, Canada, 1995.

[Rit09] Anna Ritchie. Librarything: an interview with tim spalding. *Crossroads*, 15(4):3–6, 2009.

[Riv92] Ron Rivest. The md5 message-digest algorithm. `http://tools.ietf.org/html/rfc1321`, 1992.

[RJB06] Henry A. Rowley, Yushi Jing, and Shumeet Baluja. Large scale image-based adult-content filtering. In *VISAPP '06: Proceedings of 1st International Conference on Computer Vision Theory and Applications*, pages 290–296, Setúbal, Portugal, February 2006.

[Roo10] Facebook Press Room. Facebook statistics. `http://www.facebook.com/press/info.php?statistics`, February 2010. Statistics as retrieved on February 05, 2010.

[RP97] James Rucker and Marcos J. Polanco. Siteseer: personalized navigation for the web. *Commun. ACM*, 40(3):73–76, 1997.

[RW08a] Emilee Rader and Rick Wash. Influences on tag choices in del.icio.us. In *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 239–248, New York, NY, USA, 2008. ACM.

[RW08b] Zulfikar Ramzan and Candid Wüest. Phishing attacks: Analyzing trends in 2006. In *Proceedings of 4th Conference on Email and Anti-Spam (CEAS)*, August 2008.

[Sal91] Timothy Salthouse. *Toward a General Theory of Expertise*, chapter Expertise as the circumvention of human processing limitations. Cambridge University Press, 1991. K. Anders Ericsson (editor).

[SB88] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[Sch06] Patrick Schmitz. Inducing ontology from flickr tags. In *Proceedings of the Workshop on Collaborative Tagging at WWW '06*, Edinburgh, Scotland, May 2006.

[SGMB08] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 259–266, New York, NY, USA, 2008. ACM.

[Sha48] Claude Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27, October 1948.

[SHY04] Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 675–684, New York, NY, USA, 2004. ACM.

[Sim08] Jan Simons. Tag-elese or the language of tags. *Fibreculture Journal*, 12, 2008.

[Sin01] Amit Singhal. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–42, 2001.

[SJWS02] Amanda Spink, Bernard J. Jansen, Dietmar Wolfram, and Tefko Saracevic. From e-sex to e-commerce: Web search changes. *IEEE Computer*, 35(3):107–109, 2002.

[SLR+06] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. tagging, communities, vocabulary, evolution. In *CSCW '06: Proceedings of the 20th anniversary conference on Computer Supported Cooperative Work*, pages 181–190, New York, NY, USA, 2006. ACM.

[Smi04] Gene Smith. Folksonomy: Social classification. `http://atomiq.org/archives/2004/08/folksonomy_social_classification.html`, August 2004. Last retrieved on December 01, 2009.

[Sne06] Chareen Snelson. Sampling the web: The development of a custom search tool for research. *LIBRES: Library and Information Science Research Electronic Journal*, 16(1), 2006.

[SNRI08] Elizeu Santos-Neto, Matei Ripeanu, and Adriana Iamnitchi. Content reuse and interest sharing in tagging communities. In *AAAI Spring Symposium on Social Information Processing*, March 2008.

[Sob02] Markus Sobek. Pr0 - google's pagerank 0 penalty. http://pr.efactory.de/e-pr0.shtml, 2002. Last retrieved on December 01, 2009.

[Sti06] Gary Stix. A farewell to keywords. *Scientific American*, pages 91–93, July 2006.

[Sti09] Wolfgang Stieler. Der fährtenleser im handynetz: Interview mit. *Technology Review (Germany)*, 11, 2009.

[Sur05] James Surowiecki. *The Wisdom of Crowds*. Anchor, 2005.

[SWJS01] Amanda Spink, Dietmar Wolfram, Major B. J. Jansen, and Tefko Saracevic. Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 52(3):226–234, 2001.

[SWUH09] Alan Said, Robert Wetzker, Winfried Umbrath, and Leonhard Hennig. A hybrid plsa approach for warmer cold start in folksonomy recommendation. In *Proceedings of the RecSys'09 Workshop on Recommender Systems & The Social Web*, pages 87–90. CEUR-WS Vol. 532, October 2009.

[SYMY08] Julia Stoyanovich, Sihem Amer Yahia, Cameron Marlow, and Cong Yu. Leveraging tagging to model user interests in del.icio.us. In *AAAI-SIP '08: Proceedings of the AAAI Spring Symposium on Social Information Processing*, 2008.

[SZC05] Xue-Feng Su, Hua-Jun Zeng, and Zheng Chen. Finding group shilling in recommendation system. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 960–961, New York, NY, USA, 2005. ACM.

[TDH05a] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Beyond the commons: Characterizing the value of personalizing search. In *PIA '05: Proceedings of the Workshop on New Technologies for Personalized Information Access*, pages 84–92, 2005.

[TDH05b] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456, New York, NY, USA, 2005. ACM.

[TDH07] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Characterizing the value of personalizing search. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 757–758, New York, NY, USA, 2007. ACM.

[TG06] E. Tonkin and M. Guy. Folksonomies: Tidying up tags? *D-Lib Magazine*, 12(1), 2006.

[TT91] James W Tanaka and Marjorie Taylor. Object categories and expertise: Is the basic level in the eye of the beholder? *Cognitive Psychology*, 23(3):457–482, 1991.

[Ude04] Jon Udell. Collaborative knowledge gardening. http://www.infoworld.com/d/developer-world/collaborative-knowledge-gardening-020, August 2004. Last retrieved on December 01, 2009.

[vABHL03] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Using hard ai problems for security. In *In Proceedings of Eurocrypt*, pages 294–311. Springer-Verlag, 2003.

[vABL04] Luis von Ahn, Manuel Blum, and John Langford. Telling humans and computers apart automatically. *Commun. ACM*, 47(2):56–60, 2004.

[vAD04] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, New York, NY, USA, 2004. ACM.

[Wal05] Thomas Vander Wal. Explaining and showing broad and narrow folksonomies. http://vanderwal.net/random/entrysel.php?blog=1635, February 2005.

[WCP07] Steve Webb, James Caverlee, and Calton Pu. Characterizing web spam using content and http session analysis. In *CEAS '07: Proceedings of the 4th Conference on Email and Anti-Spam*, 2007.

[WD05] Baoning Wu and Brian D. Davison. Identifying link farm spam pages. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 820–829, New York, NY, USA, 2005. ACM.

[WG07] Michael Wyszomierski and Greg Grothaus. A spider's view of web 2.0. http://googlewebmastercentral.blogspot.com/2007/11/spiders-view-of-web-20.html, November 2007. Last retrieved on March 01, 2010.

[Whi09] Tom White. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 2009.

[WLWH10] Yue Wang, Jun Li, Hee Lin Wang, and Zujun Hou. Automatic nipple detection using shape and statistical skin color information. In Susanne Boll, Qi Tian, Lei Zhang, Zili Zhang, and Yi-Ping Phoebe Chen, editors, *MMM*, volume 5916 of *Lecture Notes in Computer Science*, pages 644–649. Springer, 2010.

[WM06] Steve Wedig and Omid Madani. A large-scale analysis of query logs for assessing personalization opportunities. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 742–747, New York, NY, USA, 2006. ACM.

[Wri09] Alex Wright. Exploring a 'deep web' that google can't grasp. *The New York Times*, February 2009. Last retrieved on March 01, 2010.

[WUS09] Robert Wetzker, Winfried Umbrath, and Alan Said. A hybrid approach to item recommendation in folksonomies. In *ESAIR '09: Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 25–29, New York, NY, USA, February 2009. ACM.

[WWG05] Yushi Wang, Weiqiang Wang, and Wen Gao. Research on the discrimination of pornographic and bikini images. In *ISM '05: Proceedings of the 7th IEEE International Symposium on Multimedia*, pages 558–564, Washington, DC, USA, 2005. IEEE Computer Society.

[WZB08] Robert Wetzker, Carsten Zimmermann, and Christian Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *Proceedings of the ECAI 2008 Mining Social Data Workshop*, pages 26–30. IOS Press, 2008.

[WZBA10] Robert Wetzker, Carsten Zimmermann, Christian Bauckhage, and Sahin Albayrak. I tag, you tag: Translating tags for advanced user models. In *WSDM '10: Proceedings of the International Conference on Search and Data Mining*, 2010.

[WZM06] Harris Wu, Mohammad Zubair, and Kurt Maly. Harvesting social knowledge from folksonomies. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 111–114, New York, NY, USA, 2006. ACM.

[WZY06] Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 417–426, New York, NY, USA, 2006. ACM Press.

[XBCY07] Shengliang Xu, Shenghua Bao, Yunbo Cao, and Yong Yu. Using social annotations to improve language model for information retrieval. In *CIKM '07: Proceedings of the 16th ACM conference on Conference on information and knowledge management*, pages 1003–1006, New York, NY, USA, 2007. ACM.

[XBF+08] Shengliang Xu, Shenghua Bao, Ben Fei, Zhong Su, and Yong Yu. Exploring folksonomy for personalized search. In *SIGIR '08: Proceedings of*

the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 155–162, New York, NY, USA, 2008. ACM.

[XFMS06] Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. In *WWW2006: Proceedings of the Collaborative Web Tagging Workshop*, Edinburgh, Scotland, 2006.

[XWZC07] Yabo Xu, Ke Wang, Benyu Zhang, and Zheng Chen. Privacy-enhancing personalized web search. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 591–600, New York, NY, USA, 2007. ACM.

[XXYY08] Dikan Xing, Gui-Rong Xue, Qiang Yang, and Yong Yu. Deep classifier: automatically categorizing search results into large-scale hierarchies. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 139–148, New York, NY, USA, 2008. ACM.

[YHC02] Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. Pebl: Positive example based learning for web page classification using svm. In *KDD '02: Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 239–248, New York, NY, USA, 2002. ACM.

[YHC04] Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. Pebl: Web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 46(1):70–81, 2004.

[YKGF06] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In *SIGCOMM '06: Proceedings of the 2006 ACM conference on Applications, technologies, architectures, and protocols for computer communications*, pages 267–278, New York, NY, USA, 2006. ACM.

[YLMH09] Zhijun Yin, Rui Li, Qiaozhu Mei, and Jiawei Han. Exploring social tagging graph for web object classification. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 957–966, New York, NY, USA, 2009. ACM.

[YSR+06] Stewart Yang, Jianping Song, H. Rajamani, Taewon Cho, Yin Zhang, and R. Mooney. Fast and effective worm fingerprinting via machine learning. In *ICAC '06: Proceedings of the 2006 IEEE International Conference on Autonomic Computing*, pages 311–313, Washington, DC, USA, 2006. IEEE Computer Society.

[ZB04] Jeremy Zawodny and Derek J. Balling. *High Performance MySQL: Optimization, Backups, Replication, Load Balancing & More*. O'Reilly Media, Sebastopol, CA, USA, 2004.

[ZC06] Ziming Zhuang and Silviu Cucerzan. Re-ranking search results using query logs. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 860–861, New York, NY, USA, 2006. ACM.

[ZD07] Aaron Zinman and Judith Donath. Is britney spears spam? In *CEAS '07: Proceedings of the 4th Conference on Email and Anti-Spam*, 2007.

[Zip49] George Kingsley Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley, Cambridge, USA, 1949.

[ZMF09] Arkaitz Zubiaga, Raquel Martínez, and Víctor Fresno. Getting the most out of social annotations for web page classification. In *DocEng '09: Proceedings of the 9th ACM symposium on Document engineering*, pages 74–83, New York, NY, USA, 2009. ACM.

[ZWY06] Lei Zhang, Xian Wuand, and Yong Yu. Emergent semantics from folksonomies: A quantitative study. In S. Spaccapietra et al., editor, *Journal on Data Semantics VI*, volume 4090 of *LNCS*, pages 168–186, Heidelberg, 2006. Springer.