Privacy aware social information retrieval and spam filtering using folksonomies

Dissertation zur Erlangung des
naturwissenschaftlichen Doktorgrades
der Bayerischen Julius–Maximilians–Universität Würzburg

vorgelegt von

Beate Navarro Bullock

aus
Nürnberg

Würzburg, 2015

Eingereicht am: 27.04.2015

bei der Fakultät für Mathematik und Informatik

1. Gutachter: Prof. Dr. Andreas Hotho
2. Gutachter: Prof. Dr. Gerd Stumme

Tag der mündlichen Prüfung: 20.07.2015

Abstract

Social interactions as introduced by Web 2.0 applications during the last decade have changed the way the Internet is used. Today, it is part of our daily lives to maintain contacts through social networks, to comment on the latest developments in microblogging services or to save and share information snippets such as photos or bookmarks online.
Social bookmarking systems are part of this development. Users can share links to interesting web pages by publishing bookmarks and providing descriptive keywords for them. The structure which evolves from the collection of annotated bookmarks is called a folksonomy. The sharing of interesting and relevant posts enables new ways of retrieving information from the Web. Users can search or browse the folksonomy looking at resources related to specific tags or users. Ranking methods known from search engines have been adjusted to facilitate retrieval in social bookmarking systems. Hence, social bookmarking systems have become an alternative or addendum to search engines.
In order to better understand the commonalities and differences of social bookmarking systems and search engines, this thesis compares several aspects of the two systems' structure, usage behaviour and content. This includes the use of tags and query terms, the composition of the document collections and the rankings of bookmarks and search engine URLs. Searchers (recorded via session ids), their search terms and the clicked on URLs can be extracted from a search engine query logfile. They form similar links as can be found in folksonomies where a user annotates a resource with tags. We use this analogy to build a tripartite hypergraph from query logfiles (a logsonomy), and compare structural and semantic properties of log- and folksonomies. Overall, we have found similar behavioural, structural and semantic characteristics in both systems. Driven by this insight, we investigate, if folksonomy data can be of use in web information retrieval in a similar way to query log data: we construct training data from query logs and a folksonomy to build models for a learning-to-rank algorithm. First experiments show a positive correlation of ranking results generated from the ranking models of both systems. The research is based on various data collections from the social bookmarking systems BibSonomy and Delicious, Microsoft's search engine MSN (now Bing) and Google data.
To maintain social bookmarking systems as a good source for information retrieval, providers need to fight spam. This thesis introduces and analyses different features derived from the specific characteristics of social bookmarking systems to be used in spam detection classification algorithms. Best results can be derived from a combination of profile, activity, semantic and location-based features. Based on the experiments, a spam detection framework which identifies and eliminates spam activities for the social bookmarking system BibSonomy has been developed.
The storing and publication of user-related bookmarks and profile information raises questions about user data privacy. What kinds of personal information is collected and how do systems handle user-related items? In order to answer these questions, the thesis looks into the handling of data privacy in the social bookmarking system BibSonomy. Legal guidelines about how to deal with the private data collected and processed in social bookmarking systems are also presented. Experiments will show that the consideration of user data privacy in the process of feature design can be a first step towards strengthening data privacy.

Zusammenfassung

Soziale Interaktion, wie sie im letzten Jahrzehnt durch Web 2.0 Anwendungen eingeführt wurde, änderte die Art und Weise wie wir das Internet nutzen. Heute gehört es zum Alltag, Kontakte in sozialen Netzwerken zu pflegen, die aktuellsten Entwicklungen in Mikroblogging - Anwendungen zu kommentieren, oder interessante Informationen wie Fotos oder Weblinks digital zu speichern und zu teilen.
Soziale Lesezeichensysteme sind ein Teil dieser Entwicklung. Nutzer können Links zu interessanten Webseiten teilen, indem sie diese mit aussagekräftigen Begriffen (Tags) versehen und veröffentlichen. Die Struktur, die aus der Sammlung von annotierten Lesezeichen entsteht, wird Folksonomy genannt. Nutzer können diese durchforsten und nach Links mit bestimmten Tags oder von bestimmten Nutzern suchen. Ranking Methoden, die schon in Suchmaschinen implementiert wurden, wurden angepasst, um die Suche in sozialen Lesezeichensystemen zu erleichtern. So haben sich diese Systeme mittlerweile zu einer ernsthaften Alternative oder Ergänzung zu traditionellen Suchmaschinen entwickelt.
Um Gemeinsamkeiten und Unterschiede in der Struktur, Nutzung und in den Inhalten von sozialen Lesezeichensystemen und Suchmaschinen besser zu verstehen, werden in dieser Arbeit die Verwendung von Tags und Suchbegriffen, die Zusammensetzung der Dokumentensammlungen und der Aufbau der Rankings verglichen und diskutiert. Aus den Suchmaschinennutzern eines Logfiles, ihren Anfragen und den geklickten Rankingergebnissen lässt sich eine ähnlich tripartite Struktur wie die der Folksonomy aufbauen. Die Häufigkeitsverteilungen sowie strukturellen Eigenschaften dieses Graphen werden mit der Struktur einer Folksonomy verglichen. Insgesamt lassen sich ein ähnliches Nutzerverhalten und ähnliche Strukturen aus beiden Ansätzen ableiten. Diese Erkenntnis nutzend werden im letzten Schritt der Untersuchung Trainings- und Testdaten aus Suchmaschinenlogfiles und Folksonomien generiert und ein Rankingalgorithmus trainiert. Erste Analysen ergeben, dass die Rankings generiert aus impliziten Feedback von Suchmaschinen und Folksonomien, positiv korreliert sind. Die Untersuchungen basieren auf verschiedenen Datensammlungen aus den sozialen Lesezeichensystemen BibSonomy und Delicious, und aus Daten der Suchmaschinen MSN (jetzt Bing) und Google.
Damit soziale Lesezeichensysteme als qualitativ hochwertige Informationssysteme erhalten bleiben, müssen Anbieter den in den Systemen anfallenden Spam bekämpfen. In dieser Arbeit werden verschiedene Merkmale vom legitimen und nicht legitimen Nutzern aus den Besonderheiten von Folksonomien abgeleitet und auf ihre Eignung zur Spamentdeckung getestet. Die besten Ergebnisse ergeben eine Kombination aus Profil- Aktivitäts-, semantischen und ortsbezogenen Merkmalen. Basierend auf den Experimenten wird eine Spamentdeckungsanwendung entwickelt mit Hilfe derer Spam in sozialen Lesezeichensystem BibSonomy erkannt und eliminiert wird.
Mit der Speicherung und Veröffentlichung von benutzerbezogenen Daten ergibt sich die Frage, ob die persönlichen Daten eines Nutzers in sozialen Lesezeichensystemen noch genügend geschützt werden. Welche Art der persönlichen Daten werden in diesen Systemen gesammelt und wie gehen existierende Systeme mit diesen Daten um? Um diese Fragen zu beantworten, wird die Anwendung BibSonomy unter technischen und datenschutzrechtlichen Gesichtspunkten analysiert. Es werden Richtlinien erarbeitet, die als Leitfaden für den Umgang mit persönlichen Daten bei der Entwicklung und dem Betrieb von sozialen Lesezeichen dienen sollen. Experimente zur Spamklassifikation zeigen, dass die Berücksichtigung von datenschutzrechtlichen Aspekten bei der Auswahl von Klassi-

fikationsmerkmalen persönliche Daten schützen können, ohne die Performanz des Systems bedeutend zu verringern.

Acknowledgements

This thesis would not have been possible without the help, expertise and patience of many people.
First, I would like to thank my advisor Prof. Dr. Andreas Hotho for his support and encouragement over the years. From the first project in 2007 to the last feedback in 2015 he supported me in the realization of this thesis. During my time as a research assistant, he helped me with teaching, student mentoring and always willingly provided technical know-how. His contributions and encouragement helped turn the past years into an amazing journey and made it possible to finally reach the finish line.
I am also very thankful to Prof. Dr. Gerd Stumme for his continuous support and for always having an open ear during my years in Kassel. In both teams, in Kassel and Würzburg, I found an enthusiastic learning atmosphere. Ideas were welcome and time and resources organized to realise them. I very much enjoyed this creative, optimistic environment.
Thank you to Prof. Dr. Frank Puppe for supporting me by being the chairman of the board of examiners and for integrating me into the Würzburg team.
The ideas, approaches and methods discussed originate from my work in the university teams at Kassel and Würzburg. Thank you to the coauthors and team members, especially to Christoph, Dominik, Elmar, Folke, Robert and Stephan from the Knowledge and Engineering Group in Kassel and Florian, Martin, Peter, Martina and the other team members from the University of Würzburg. Also, many thanks to Sven, Björn, Alexander, Monika and Petra for their support.
Additionally, I would like to thank Hana Lerch and Prof. Dr. Alexander Roßnagel from the legal department at the University of Kassel for their collaboration in the project "Informationelle Selbstbestimmung im Web 2.0". The legal discussion in this thesis is based on their input. Our discussions were stimulating and allowed me to see a new perspective on the handling of personal data.
Finally, I thank my family. My parents, who encouraged and helped me at every stage of my life. My husband Pablo, not only for his patience and humour, but also for his English thesis-guidance and so many Sunday mornings taking our daughters Laura and later Carmen to the playground while I stayed at home continuing to work step-by-step. Just the same, thank you for two curious, active and happy girls, who always make me smile when they are home again.

Contributions

The research presented in this thesis has been conducted together with other collegues of the Knowledge and Data Engineering Team of the University of Kassel, the institute of law of the University of Kassel and the Applied Informatics and Artificial Intelligence Team of the University of Würzburg. The different findings therefore result from cojoint work within the research teams. For each of the three research fields presented in this thesis, the author's contributions and the collaborations with other authors are listed below.

Social Search

Contributions

- The experiments to compare search systems and folksonomies have been conducted by the author. Andreas Hotho and Gerd Stumme supported the work with helpful ideas and the discussion of results. The comparison of rankings was computed by the author using an implementation of the FolkRank algorithm by Robert Jäschke.

- The idea of creating a folksonomy alike structure from logdata stems from Robert Jäschke. The author contributed to this work with the computational transformation of logdata into a logsonomy, and the conductions of experiments considering network properties. Parts of these experiments are based on the implementation and design of experiments conducted by Christoph Schmitz.

- The semantic analysis of a logsonomy was done together with Dominik Benz and Praveen Kumar. The author's input consisted of the compilation of data and the discussion of results.

- The idea of enriching implicit feedback data by using tagging information has been introduced by the author. While the experiments concerning the FolkRank algorithm have been conducted by Robert Jäschke, the other paper's experiments have been computed and analysed by the author. Results were discussed with Robert Jäschke and Andreas Hotho.

Publications

- Beate Krause, Andreas Hotho, Gerd Stumme. (2008). A comparison of social bookmarking and traditional search. ECIR 2008.

- Beate Krause, Robert Jäschke, Andreas Hotho, Gerd Stumme. (2008). Logsonomy - Social Information Retrieval with Logdata. Hypertext 2008.

- Beate Navarro Bullock, Robert Jäschke, Andreas Hotho. Tagging data as implicit feedback for learning-to-rank. Poster at WebScience 2011. Koblenz, Germany

- Benz, Dominik; Krause, Beate; Kumar, G. Praveen; Hotho, Andreas & Stumme, Gerd: Characterizing Semantic Relatedness of Search Query Terms Bled, Slovenia: September (2009).

Spam Detection

Contributions

- The idea of detecting spam arouse from the necessity to fight the spam in the social bookmarking system BibSonomy. The dataset was created by all administrators of the group. Most of the features were implemented by Christoph Schmitz. The additional implementation of features, analysis and comparison of different classifiers were conducted by the author.
- The spam detection framework was implemented by Stephan Stützer in the scope of a master project. It was adjusted to be used with BibSonomy by the author. From beginning 2009 to midth 2011 it was run by the author.
- The idea, implementation and computation of the detection of local patterns in spam data was done by Florian Lemmerich, Martin Atzmüller and Andreas Hotho. The author compiled the spam data and contributed to the interpretation of results.

Publications

- Beate Krause, Christoph Schmitz, Andreas Hotho, Gerd Stumme. (2008). The Anti-Social Tagger - Detecting Spam in Social Bookmarking Systems. AIRWEB 2008.
- Dominik Benz, Andreas Hotho, Robert Jäschke, Beate Krause, Folke Mitzlaff, Christoph Schmitz, Gerd Stumme: The social bookmark and publication management system BibSonomy In: The VLDB Journal, Vol. 19 Berlin / Heidelberg: Springer (2010), S. 849-875.
- Atzmueller, Martin; Lemmerich, Florian; Krause, Beate & Hotho, Andreas: Spammer Discrimination: Discovering Local Patterns for Concept Characterization and Description Bled, Slovenia: September (2009) .

Data Privacy

Contributions

- The analysis of privacy aspects in BibSonomy results from the cooperation with the data privacy group, in particular Hana Lerch and Alexander Roßnagel. The system analysis took place in extensive discussions of the project meetings including Gerd Stumme, Andreas Hotho, Hana Lerch, Alexander Roßnagel and the author.
- The comparison of different features for spam detection and their qualification for data privacy aware data mining was conducted by Hana Lerch and the author. While the author contributed the implemtation and computation of different features, Hana Lerch provided the legal evaluation.

Publications

- Beate Krause, Hana Lerch, Andreas Hotho, Alexander Roßnagel, Gerd Stumme (2010), Datenschutz im Web 2.0 am Beispiel des sozialen Tagging-Systems BibSonomy, Informatik-Spektrum.

- Beate Navarro Bullock, Hana Lerch, Andreas Hotho, Alexander Roßnagel, Gerd Stumme. Privacy-aware spam detection in social bookmarking systems, i-Know 2011, Graz, Austria.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

At the start of the Internet era, people mostly used the Web to search and read static digital content. A paradigm shift occurred with the advent of the *Social Web* where user participation and the creation of social, virtual relationships became an integral part of Internet usage. The term *Web 2.0*, first introduced during a brainstorming session between O'Reilly and MediaLive International in 2004 [O'Reilly, 2005], has become accepted to describe Social Web applications integrating features which center around the participation of users. Such Web 2.0 applications comprise online marketplaces such as eBay [1] or Amazon [2], content sharing sites such as Flickr [3] or Delicious [4], online social networks such as Facebook [5], blogs or the online encyclopedia Wikipedia [6]. Nowadays, social services are part of many user's day-to-day digital experience. Many activities of everyday life have moved partially or fully into the virtual world. Friendships are maintained via social networks, online shopping saves time and energy and news or opinions are communicated via text messaging services.

The growing popularity of Web 2.0 applications has led to a wealth of digital user data which can be integrated and analysed. The exploration of such data helps to gain knowledge about user interests and preferences as well as the connection between them. This information can be harnessed by applications, for example by providing personalized recommendations and personal search algorithms or by tailoring commercial advertisements for each user.

One of the popular types of Web 2.0 applications are social bookmarking systems. In these systems users share online resources in form of bookmarks. To make the bookmarks retrievable for themselves and others they add tags, i. e., descriptive keywords, to their resources. With many different users sharing bookmarks and tagging them, a common information structure is created which can be called a *folksonomy*. The structure emerges over time influenced by the numerous users and the possibility to interact with each other. In general, it reflects a common knowledge, a form of *collective intelligence*, among the

[1] `http://www.ebay.com/`
[2] `http://www.amazon.com/`
[3] `http://www.flickr.com/`
[4] `https://delicious.com/`
[5] `http://www.facebook.com/`
[6] `http://www.wikipedia.org/`

system's users.
The integration of user created content and the collection of user generated data opens a wide field for exploration as many different aspects of handling social data can be considered: From gaining an understanding about the dynamic structures evolving, building user-tailored functionalities based on their personal data to analysing the collected data to solve specific system problems such as spam. This thesis concentrates on these research areas: The first part deals with search in social bookmarking systems compared to existing search engines. How do both systems differ and how can they benefit from each other? The term *Social Search* embraces all research done in this field. The use of social bookmarking systems (and Web 2.0 applications in general) is not always a win-win situation for a system's provider and its users. Two major problems can be identified in this context: The identification of spam users and the protection of user privacy, which are the topics of the second and third thesis' core areas, *spam detection* and the *protection of data privacy* in the Social Web. Each of the just defined research fields will be further described in the following paragraphs.
Social Search. Depending on the amount of users and their interests, social bookmarking systems provide large document collections concerning many different topics. Each bookmark is enhanced with metadata in form of tags. The knowledge collected through sharing and describing bookmarks can be leveraged by information retrieval techniques. People can therefore use social bookmarking systems to find information – in a similar way as they use other existing online information retrieval systems such as search engines.
One major difference between information retrieval in search engines and social bookmarking systems is the way, the document collection is created [Krause et al., 2008a]. Search engine providers automatically crawl the Web. New sources are retrieved by following the hyperlinks of an already processed website. In social bookmarking systems, the document collection is created manually by the system's users who post bookmarks they find interesting.
In search engines, the retrieved documents are indexed using the document's text and other available (meta-)data. Users can search through the document collection by entering search words in a simple user interface and click on the search results shown in a list sorted by relevance. Information retrieval in social bookmarking systems is primarily supported by the system's specific structure. Since each post containing a *bookmark* has been submitted by a *user* and enriched with metadata in form of *tags*, each document in the collection is linked to others by this information. Users can find new sources by browsing through all bookmarks with specific tags or by looking at posts of users they find interesting. The tripartite structure enables serendipitous browsing [Benz et al., 2010a]: Users "stumble" on interesting sources by following the links provided by the system's user interface.
While relevance in search engines is the result of an algorithm taking different aspects such as the page's link structure, content or the site's refresh period into account, the relevance of social bookmarking entries is determined by its users. When they spend time to bookmark and describe a resource they show that the resource is relevant for the selected tags. In recent years, search engines have integrated many social features into their searches. Rankings are influenced by (implicit) user feedback such as a click on a link shown in the result list of a specific search. Further, result pages and user accounts allow users to share their results, store searches or recommend resources. On the other hand, social bookmarking systems have adopted techniques from information retrieval to

enhance their search (one example is the FolkRank algorithm [Hotho et al., 2006a] based on Google's PageRank algorithm [Brin and Page, 1998]).
At the beginning of social bookmarking systems, no one could have predicted where Web 2.0 and social bookmarking systems were headed. Would they establish themselves as an alternative to online search systems or vanish as soon as the hype was over? What made those systems attractive for users? Could search engines generally benefit from social interactions as introduced in Web 2.0 applications? In order to gain a better understanding about differences and similarities of social bookmarking systems and search engines, different aspects such as user behaviour and the system's usage or structure were analysed in several comparative studies (for example [Heymann et al., 2008; Kolay and Dasdan, 2009b; Noll and Meinel, 2008a]).
The first part of this thesis deals with a comparative analysis of search in folksonomies and search in search engines. Similarities and differences between the usage of tags and query terms, the document collection or the structure of the two systems are presented and discussed.

Spam Detection. The spam problem is well known by anyone using search engines or e-mail communication. Malicious users try to manipulate the ranking results for search queries in order to increase the traffic to their websites be it for commercial or political interests, or simply to damage the service provided [Krause et al., 2008c]. For this, spammers use different techniques such as adding specific keywords to their websites or building link farms.
Social bookmarking systems also have become an attractive place for spammers. All they need is an account, then they can freely post entries which bookmark the target spam website. Depending on the social bookmarking system's popularity, one or more posts with the spam bookmark improves the bookmark's ranking result in search engines. Due to the growing percentage of spam entries, the quality of social bookmarking systems decreases. Instead of finding relevant and interesting documents users are exposed to publicity, nonsense information or political and religious ideologies. Legitimate participants tend to lose interest in the system and switch to other tools providing a better service [Navarro Bullock et al., 2011b].
"In order to retain the original benefits of social bookmarking systems, techniques need to be developed which prevent spammers from posting in these systems, or at least from having their malicious posts published" [Navarro Bullock et al., 2011b]. A common method is to complicate the interaction with the system so that automatic spam attacks fail [Krause et al., 2008c]. The use of captchas is a common example for such an approach. In order to verify that inputs are carried out by a human user rather than a computer, users must enter a response that is most likely impossible for a computer to recreate. While this may prevent bots from spamming, the incentive of having a published bookmark seems to be high enough for human spammers to overcome this difficulty. Furthermore, legitimate users might also be put off due to the cumbersome handling of the system. Automatic spam fighting methods on the other hand, do not interfere with users and their activities: So called spam filters identify malicious posts based on their features and previous experiences. Such posts can then be removed silently from the public parts of the social bookmarking system.
Supervised learning methods such as classification algorithms are often used by spam filters. Many of them have been implemented for detecting spam in e-mail communication

(for example [Blanzieri and Bryl, 2008]). Based on a set of features computed for each message, the classification approaches learn how to differentiate between spam and legitimate messages. In order to transfer spam detection approaches to social bookmarking systems, classification methods need to be adjusted and the features to characterize spam and non-spam bookmarks need to be designed carefully. The second part of this thesis concerns the design of appropriate features and the selection of well-performing spam detection algorithms. The spam problem in social bookmarking systems will be analysed in general, experiments concerning the performance of different spam features presented and a framework to actively fight spam in the social bookmarking system BibSonomy introduced.

Protection of Data Privacy. Users participating in online applications with social and interactive features face a dilemma: They benefit from the free service to present themselves and interact with others, on the other hand they disclose sensitive data which can be a risk. As soon as they publish their data on the Web, they loose control about who reads, stores and distributes this information. In addition to the exploration of user generated content, enhanced information technologies and the easy availability of storage make it possible for service providers to trace a user's path when interacting with the system.

On top of that, data from different systems, organisations or websites can be compared and merged. By using intelligent data mining techniques indirect information such as user relations, possible interests and profiles can be exploited and leveraged for purposes other than the users had in mind. In order to ensure a fair and conscientious collection and processing of personal information privacy-enhancing technologies need to be integrated into Web 2.0 systems. Such techniques include the minimization of personal data collected, the anonymisation of user data (for example user log files) or mechanisms to delete or unlink personal information. However, privacy issues can not be solved only through the provision of an appropriate technical infrastructure: Existing legal frameworks need to be kept up to date regarding user needs and privacy requirements [Fischer-Hübner et al., 2011] triggered by the technological changes. Since those changes happen much faster than the establishment of laws and the introduction of privacy enhancing technologies, a more forward-looking perspective on privacy needs to be established among system designers, developers and users.

While data protection has been discussed in context of social networks (for example in Eecke and Truyens [2010]; Krishnamurthy and Wills [2008]), there have been few considerations of data privacy issues in other Web 2.0 applications, especially in social tagging systems. However, such systems heavily rely on collecting and processing data by storing their user's public and private entries and profile information. In order to prevent users from loosing control over their personal information published on the Web, these systems need to come under pressure regarding the kinds of data they collect as well as their handling of possibly personal information.

The third research topic of this thesis looks into the handling of data privacy in social bookmarking systems. Legal guidelines about how to deal with the private data collected and processed in social bookmarking systems are presented. Experiments will show that the consideration of user data privacy in the process of algorithm and feature design can be a first step towards strengthening data privacy.

1.2 Problem Statement

This thesis investigates three different topics in the context of social bookmarking systems: Social search, spam detection and data privacy. For each of these areas the specific research questions are summarized in this Section.

1.2.1 Social Search

The main question asked is:

How does information retrieval in folksonomies compare to that of traditional search engines?

We search for similarities and differences between information retrieval on the Web and in social bookmarking systems. We hereby consider four different aspects of social search:

(1) **Usage behaviour and system content**: The processes of finding information by entering a search term and tagging information by adding keywords to a relevant resource are compared: How do tags differ from query terms? Do users tag and search by using the same vocabulary? Do users click on the same resources in a search result list as they would post in a social bookmarking system?

(2) **Structure**: A user clicking on a search result can be considered as implicit feedback indicating that the link might be relevant for the search request. A tripartite structure linking elements of users, clicked resources and query terms is similiar to the structure consisting of connected elements of tags, resources and users in a folksonomy. Thus, in this part we consider if there is a inherent folksonomy like structure in query log files and how its properties compare to properties found in a folksonomy.

(3) **Semantics**: The emergent semantics of folksonomies have been subject to several analyses as can be found in Cattuto et al. [2008] or Körner et al. [2010]. The question arises however, if similar semantics evolve from query logs and tagging systems.

(4) **Integration**: Principles of traditional search such as the PageRank algorithm have been transferred to the structure of folksonomies (see Hotho et al. [2006a]). This thesis looks at how the information collected in tagging systems can be of use for traditional search.

These questions will be answered by analysing large datasets from the social bookmarking systems BibSonomy and Delicious (described in Section 2.6) and a query log file of the search engine MSN (see Section 6.2).

1.2.2 Spam Detection

The main research question asked is:

How can we best detect and eliminate spam in social bookmarking systems?

The first step will be an analysis of spam in social bookmarking systems. Can different categories of spammers be distinguished? If so, how do they differ from legitimate users?

We will hereby define appropriate characteristics to differentiate between spam and non-spam.
These characteristics are then used as features for different classification algorithms. Using a comprehensive dataset of the social bookmarking system BibSonomy, we will identify the best algorithms considering accuracy and performance.
Further analyses of the spam problem in social bookmarking systems have been conducted at the ECML/PKDD discovery challenge of 2008 [Hotho et al., 2008], where different participants classified spam using a similar dataset to the one used for this study. Finally the main features allowing a successful detection of spam have been tested regarding classification accuracy on the one hand and the capability to preserve data privacy on the other.
The results derived from the different experiments have been implemented into a spam detection framework used to detect spam in the social bookmarking system BibSonomy, thereby demonstrating the applicability in the real world.

1.2.3 Data Privacy

We will consider the following research question:

How can personal information be protected in social bookmarking systems?

At first sight, one might wonder about this question. Compared to online social networks such as Facebook, social bookmarking systems seem to be rather anonymous. People select a user name and place a few links in the system. Is there any personal information required? We will show such traces of personal information by analysing what kind of data is processed in social bookmaring systems and how they are further used for typical bookmarking applications and add ons. Together with legal experts, guidelines have been defined to help system designers, developers and users to preserve data privacy in social bookmarking systems.
Besides general considerations of data privacy in social bookmarking, a specific idea for data protection will be further investigated: privacy-preserving data mining. We will thereby renounce on the development of techniques to hide or disturbe information, but present an approach about how to consider alternatives to personal data in data mining applications. Different privacy levels in which data can be classified are defined. With the help of an example (spam detection in BibSonomy), various spam features using different kinds of system data (inventory, usage and content data) are evaluated by comparing their applicability in terms of performance and level of privacy.

1.3 Thesis Outline

This thesis consists of three parts as pictured in Table 1.1.
After the introduction in Chapter 1, the first part (*Foundations*) summarizes important concepts and related work. Chapter 2 introduces social tagging systems in general, outlines related work for data mining in such systems and describes two examples of social bookmarking systems, BibSonomy and Delicious, as their data is used for several experiments. The following three chapters present the relevant basics for each of the three main research fields:

- Chapter 3 explains important concepts in social search.

Table 1.1: Thesis outline with references of the different research areas to their backgrounds, experimental part and applications (if available).

	Social Search	Spam Detection	Data Privacy
Foundations	Chapter 3	Chapter 4	Chapter 5
Methods	Chapter 6	Chapter 7	Chapter 8
Applications	—	Chapter 9	Chapter 10

- Chapter 4 describes the spam problem in general including a definition of spam, important algorithms for spam classification and evaluation measures. Additionally, the state-of-the-art of spam detection in social media applications is discussed.
- Chapter 5 introduces the necessary terminology to understand the legal background of data privacy.

In the second part (*Methods*) the different experiments carried out are presented:

- Chapter 6 presents our findings regarding the comparative analysis of social bookmarking and search data with respect to user behaviour, structure, semantics and usability for implicit feedback algorithms.
- Chapter 7 identifies important spam features and analyses their performance, describes the spam detection task as well as the results of the ECML/PKDD discovery challenge 2008 and summarizes subgroup discovery for spam data.
- Chapter 8 investigates the combination of performance and data privacy aspects for the selection of features used in spam classification algorithms.

The spam application developed out of the findings in *Methods* as well as the data privacy analysis of the social bookmarking system BibSonomy will be discussed in the third part, *Applications*:

- Chapter 9 provides an overview of the BibSonomy spam detection framework.
- Chapter 10 explores the functions offered by the social bookmarking system BibSonomy with respect to the data used and the legal implications in case of the collection and processing of private data.

Finally, Chapter 11 will summarize the findings of the three research fields and provide an outlook on future research and developments.

Part I

Foundations

Chapter 2

Collaborative Tagging

The experiments presented in this thesis are based on data from social bookmarking systems, one form of collaborative tagging. In order to gain an understanding about these systems, this chapter will introduce social tagging and briefly summarize recent research concerning those systems. Why did they gain so much popularity? What can we learn from analysing the structure and the dynamics of such systems? How can we benefit from the annotated data collected and shared by the system's users?
Section 2.1 starts by introducing indexing systems in general. Different characteristics of social tagging systems (strengths and weaknesses, user motivation and a classification of tags) will then be listed. Section 2.2 presents the formal model of collaborative tagging, called *folksonomy*. Research and findings considering the dynamics of tagging (Section 2.3), its relation to the Semantic Web (Section 2.4) and recommender systems (Section 2.5) will be presented. The chapter ends with a description of two social bookmarking systems: BibSonomy and Delicious (Section 2.6). Both systems provide data used for experiments in this thesis.

2.1 Characterizing Collaborative Tagging Systems

Social tagging systems enable users to store, organize and share resources such as bookmarks, publication metadata or photos over the Web. The resources can be labeled with any keyword (tag) a user can think of [O'Reilly, 2005]. The tags serve as a representation of the document, summarizing its content, describing the relation to the resource's owner or expressing some kind of emotion the resource owner feels. The process itself is coined *tagging*. The tagged recources can be used for "future navigation, filtering or search" [Golder and Huberman, 2005].

2.1.1 Indexing Documents

Adding keywords to digital resources (mostly to documents) existed before the arrival of tagging systems either in form of manual indexing or by automatically extracting keywords from text.
The process of (manual) document indexing is twofold: A *conceptual analysis* of the item helps to understand the meaning and importance of it. This can be different for different people, depending on their background, interests or intentions. *Translation* then helps

to select the appropriate index terms to describe the resource considering the individual resource context [Lancaster, 2003].
In most classical document categorization and indexing schemes resources have to be classified into predefined and firm categories. For example, the Yahoo! [1] directory contains categorized websites. Human editors in the company had to assign the websites to appropriate categories [Angel, 2002]. Another manually created document index is the Open Directory Project (ODP) [2], launched in 1998. It has been created by a community of volunteers from all over the world. As it provides an open content license, it can be used freely [Kapitzke and Bruce, 2006].
Documents can also be indexed based on ontologies. Users can describe a document by using a pre-defined markup which allows the definition of concepts and the linkage between documents by identifying relations and properties [Andrews et al., 2012]. Several vocabularies used to annotate web documents exist. For example, the Common Tag[3] format provides a specification to identify different concepts of a document which are then linked via URIs to databases of structured content.
On the Web, the standard way of organizing documents is by automatic indexing. Manual approaches such as Yahoo!'s directory became more and more difficult to maintain due to the Web's rapid growth and changing nature. Search engines such as Google [4] or Bing [5], automatically index documents. They create an inverted index where documents are ordered according to specific keywords automatically extracted from them. Such key words can then be used in the retrieval process.
With the advent of collaborative tagging systems, manual indexing became popular again. In contrast to pre-defined indexing schemes, tagging allows spontaneous annotation. Keywords can be selected as they come up in one's mind without having to conform to predefined rules or standards. Hence, not only field experts are able to annotate resources, but web participants themselves freely categorize content. The properties and architecture of tagging systems which enable the simple annotation of resources will be discussed in the next section.

2.1.2 System Properties

Tagging systems differ in their design and the functionalities they offer. Marlow et al. [2006] proposed several dimensions which allow the classification of collaborative tagging systems. According to the authors, the applications vary according to the kinds of objects (web bookmarks, photos or videos) they provide storage space for. The source of such objects also differs: In user-centric systems such as BibSonomy or Flickr, users collect material. In other systems, the providers themselves present data which can then be annotated by its users (for example Last.fm, which provides music). Concerning the process of tagging systems support and restrict their users in different ways. Many systems, for example, have implemented recommender systems to suggest tags and help users finding appropriate vocabulary. In some systems users can annotate all resources (Delicious), while users in other systems can explicitly decide if they want other users to be able to

[1] http://dir.yahoo.com/
[2] http://www.dmoz.org/
[3] www.commontag.org
[4] http://www.google.com/
[5] http://www.bing.com/

tag their resources. Also, the aggregation model of tag-resource assignments is different. While different users in BibSonomy are allowed to assign the same tag to a resource, Flickr prohibits the same tag-resource assignments among different users. Finally, many systems provide additional functionalities concerning social connections among users (for example: joining groups) and their resources (for example: organizing photos in an album).
To understand the success of tagging systems, one needs to consider the strengths and weaknesses of social tagging. Several aspects were discussed (see [Golder and Huberman, 2006b; Marlow et al., 2006; Quintarelli, 2005; Wu et al., 2006b] and also the dissertations of Noll [2010]; Yeung [2009]). We will briefly summarize those characteristics in the following.

Strengths

Low cognitive cost and entry barriers While more formal classification systems such as catalogues or ontologies require the consideration of the domain and specific vocabulary or rules, no prior knowledge or training is required when starting social tagging [Wu et al., 2006b].

Serendipity One of the fascinating features of tagging systems is their ability to guide users to unknown, unexpected, but nonetheless useful resources. This ability is triggered by the system's small world property: with only a few clicks one can reach totally different resources, tags and users in the system.

Adaptability In contrast to top-down approaches where a pre-defined classification system is given and experts or at least people knowing the system are required to classify resources according to the classification scheme, social tagging systems allow a bottom-up approach where users can add keywords without having to adhere to a pre-defined vocabulary, authority or fixed classification. The liberty of using arbitrary annotations allows a flexible adaptation to a changing environment where new terms and concepts are introduced [Spiteri, 2007; Wu et al., 2006a]. However, as the majority of users annotates their resources with similar or the same tags, a classification system can still evolve.

Long tail As everyone can participate and no pre-requisites have to be met, "every voice gains its space" [Quintarelli, 2005]. Consequently, the systems do not only contain mainstream contents, but also original and individual items which might turn into popular ideas.

Weaknesses

Ambiguity of tags The missing control of what kind of vocabulary is used in tagging systems entails the typical challenges which a natural language environment provides: ambiguity, polisemy, synonyms, acronyms or basic level variation [Golder and Huberman, 2006a; Spiteri, 2007].

Multiple languages Since people with different cultural backgrounds use tagging systems, multiple terms from different languages with the same meaning can be encountered.

Syntax issues In most of the social tagging systems users can enter different tags by using the space as delimiter. Problems arise when users want to add tags consisting of more than one term. Often, the underscore character or hyphens are used to combine such terms.

No formal structure Tags as they are entered into the system have no relation among each other. One needs to apply further techniques to discover inherent patterns.

2.1.3 Tagging Properties

User Motivation for Tagging

Why do people manually label items? Gupta et al. [2010] distinguished eight motives in their survey of tagging techniques.
Most users annotate resources in order to facilitate their *future retrieval*. By making a resource public, categorizing it and sometimes even adding it to a specific interest community, the resource becomes available for a system's audience (*contribution and sharing*). Often, annotators use popular tags to make people aware of their resources (*attracting attention*) or they use tags to express some part of their identity (*self presentation*). The tag *myown* in the social tagging system BibSonomy, for example, states that the annotating user is an author of the referenced paper. With the help of tagging users can demonstrate their opinion about certain resources (*opinion expression*). Tags such as *funny*, *helpful* or *elicit* are examples of such value judgements. Some tags reflect organisational purposes of the annotator. Often used examples are *toread* or *jobsearch* (*task categorization*). For some users, tagging is a *game or competition*, triggered by some pre-defined rules: Playing the ESP game [von Ahn and Dabbish, 2004] one user need to guess labels another user would also choose to describe images displayed to both users. Other users earn money (*financial benefits*): There are websites such as Squidoo [6] which pay users a small amount of money for annotating resources.

Classification of Tags

In order to get a better understanding of the nature of tags, various authors [Al-Khalifa and Davis, 2007; Bischoff et al., 2009; Golder and Huberman, 2006a; Overell et al., 2009; Sen et al., 2006; Wartena, 2010] identified different types of tags.
Most useful for information retrieval or data mining tasks are *factual tags* indicating facts about the resource such as concepts, time or people. Three kinds of factual tags can be listed:

- Content-based tags describe a resource's actual content such as *ranking-algorithm*, *java*, *subaru* or *parental-guide*.
- Context-based tags can be used to identify an items context in which it was created or can be located. Examples are *San Francisco*, *christmas* or *www-conf*.
- (Objective) attribute tags describe an object's specific characteristics which may not be explicitely mentioned with the object. For example, the blog of the hypertext conference can be tagged with *hypertext-blog*.

[6] `www.squidoo.com`

Attribute tags can also be part of the category of *personal tags*, when they serve to express an opinion or a specific feeling such as *funny* or *interesting*. Personal tasks are often more difficult to use for inferring general, descriptive knowledge. They can be used, however, for specific tasks such as sentiment analysis. Personal tags include:

- Subjective tags state an annotator's opinion.
- Ownership tags express who owns the object, e. g., *mypaper*, *myblog* or *mywork*.
- Organisational tags denote "what to do with the specific resource". ([Navarro Bullock et al., 2011a]) They are often time-sensitive. For example, the *to-do,read,listen* tags loose significance if the task has been carried out.
- Purpose tags describe a certain objective the user has in mind considering the specific resource. Often, this relates to information seeking tasks such as learning about java (*learning_java*) or collecting resources for a chapter of the dissertation (*review_spam*).

Authors (such as [Bischoff et al., 2009; Overell et al., 2009; Wartena, 2010]) intend to automatically identify tag types in order to better explore the semantic structure of a tagging system. Categories can then be used for tag recommendation, categorization, faceted browsing or information retrieval.

Types of Annotators

Another way to look at the annotation process is to describe the nature of taggers. Körner et al. [2010a,b] identified two types of taggers: *categorizers* and *describers*.

- A categorizer annotates a resource using terms of a systematic shared or personal vocabulary, often some kind of taxonomy. Their size of vocabulary is limited and terms are often reused. A categorizer aims at using tags for his or her personal retrieval [Körner et al., 2010a].
- Describers annotate resources having their later retrieval in mind. They consider tags as descriptive labels which characterize the resource and can be searched for. The size of a describer's vocabulary can be large. Often, tags are not reused [Körner et al., 2010a].

In Körner et al. [2010a] it could be shown, that the collaborative verbosity of describers is more useful to extract the semantic structure from tags. Most users show a mixed behaviour. If users own many tags only applied once, they tend to be describers. Additionally, a vocabulary growing quickly hints towards a describer. Categorizers can be identified by their low tag entropy as they apply their tags in a balanced way. Körner et al. [2010a] restrict their findings to moderate verbose taggers excluding spammers, which negatively influence the semantic accuracy.

2.2 Formal Model

The structure of social tagging systems has been termed *folksonomy*. The term is a composition of the two terms: *folk* and *taxonomy*.

Folk refers to people, i. e., the users of the tagging system. A *taxonomy* can be considered as a hierarchical structure used to classify different concepts [Knerr, 2006]. The hierarchy is built from "is-a" relationships, i. e., subsumptions going from general concepts to more specific ones. A concept A is subsumed by a concept B ($A \leq B$) if the set of instances classified under A is intentionally constrained to be equal to a subset of the instances classified under B [Sacco and Tzitzikas, 2009]. Taxonomies are normally designed by an expert of the domain. An example of a taxonomy is the Dewey Decimal Classification system [OCL], introduced by librarians to organize their collections [Breitman and Casanova, 2007]. Web documents have also been categorized with the help of taxonomies (examples here are the Yahoo! Directory [7] and the Open Directory Project (ODP) (see Section 2.1.1). The composition of *folk* and *taxonomy* describes the creation of a lightweight taxonomy which emerges from the fact that many people ("folk") with a similar conceptual and linguistic background as well as common interests annotate and organize resources [Marlow et al., 2006].
Depending on the annotation rules, one can distinguish between narrow and broad folksonomies [Wal, 2005]. In broad folksonomies multiple users add tags to a resource, while in narrow folksonomies resources are normally tagged only by the person who owns the resource.
Formal definitions of a folksonomy have been presented by i.a. Halpin et al. [2007]; Hotho et al. [2006c]; Mika [2005]. They all have in common that they describe the connections between users, tags and resources. We follow the notion of Hotho et al. [2006c], which is depicted in Definition 2.2.1. The definitions further down (Definition 2.2.2 and Definition 2.2.3) are also based on Hotho et al. [2006c].

Definition 2.2.1 *A* folksonomy *is a tuple* $\mathbb{F} := (U, T, R, Y)$ *where*

- U*,* T*, and* R *are finite sets, whose elements are called* users*,* tags *and* resources*, resp., and*
- Y *is a ternary relation between them, i. e.,* $Y \subseteq U \times T \times R$*, whose elements are called tag assignments (TAS for short).*
- $\prec$ *is a user-specific subtag/supertag-relation, i. e.,* $\prec \subseteq U \times T \times T$*, called* is-a relation*.*

Figure 2.1 illustrates Definition 2.2.1. Elements of one of the three sets are connected to elements of the remaining sets through the ternary relation Y. For example, (u_1, t_1, r_1) is a TAS of the depicted folksonomy.
Users of a bookmarking system are normally identified by a unique name they selected when registering. Tags are arbitrary strings. In most of the systems, they are divided by empty spaces. A folksonomy's resource can vary from URLs (for example Delicious, BibSonomy) to photos (Flickr) or videos (YouTube).
The *is-a relation* described in the definition classifies tags in form of super-sub-concept relationships [Hotho et al., 2006c]. Not all tagging systems realise this functionality. In BibSonomy, such relations can be defined by the system's users. Delicious allows the creation of so-called tag *bundles*: users can define a set of tags and assign a group name

[7] `http://dir.yahoo.com/`

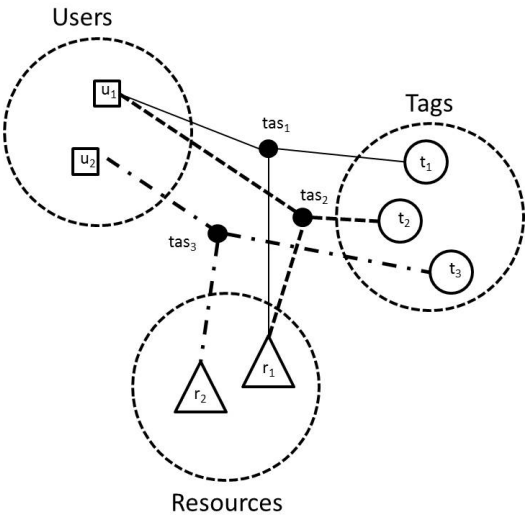


Figure 2.1: Elements of the three finite sets users U, tags T and resources R are connected through the ternary relation Y. For example, user u_1, tag t_1 and resource r_1 is the tag assignment tas_1 represented by a hyperedge.

to them. One can ignore the is-a relation, and simply define a folksonomy as a quadruple $\mathbb{F} := (U, T, R, Y)$.
Definition 2.2.2 describes a folksonomy for one user – a personomy. It basically considers only the tags and resources which the user in question submitted to the system. Figure 2.1 depicts two personomies. Both users have tagged one resource. However, the personomy of user u_1 contains a TAS more since the user assigned two tags to the resource.

Definition 2.2.2 *The* personomy $\mathbb{P}_u$ *of a given user* $u \in U$ *is the restriction of* $\mathbb{F}$ *to* u*, i. e.,* $\mathbb{P}_u := (T_u, R_u, I_u, \prec_u)$ *with* $I_u := \{(t, r) \in T \times R \mid (u, t, r) \in Y\}$*,* $T_u := \pi_1(I_u)$*,* $R_u := \pi_2(I_u)$*, and* $\prec_u := \{(t_1, t_2) \in T \times T \mid (u, t_1, t_2) \in \prec\}$*, where* π_i *denotes the projection on the* i*th dimension.*

An important concept in the world of folksonomies is a *post*, which is presented in Definition 2.2.3. A post basically represents the set of tag assignments of one user for one resource. Figure 2.1 depicts two posts. The post of user u_1 is composed of two tag assignments, while the post of user u_1 contains one tag assignment.

Definition 2.2.3 *The set* P *of all* posts *is defined as* $P := \{(u, S, r) \mid u \in U, r \in R, S = \mathrm{tags}(u, r), S \neq \emptyset\}$ *where, for all* $u \in U$ *and* $r \in R$*,* $\mathrm{tags}(u, r) := \{t \in T \mid (u, t, r) \in Y\}$ *denotes all tags the user* u *assigned to the resource* r*.*

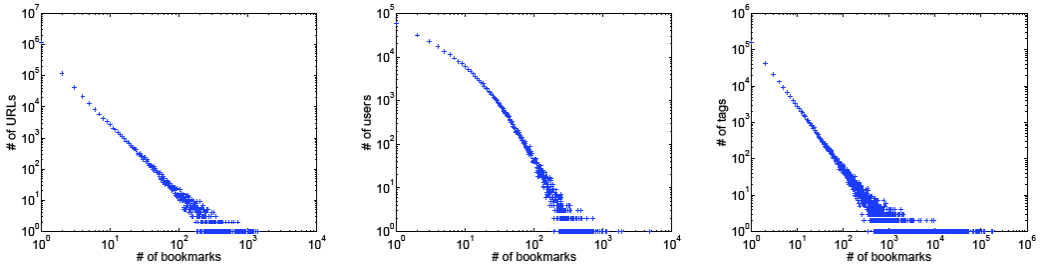
Though we focus on a folksonomy with users, tags and resources as elements, it is possible to enhance the structure to include more dimensions [Abel, 2011]. The authors of Wu et al. [2006b], for instance, consider a fourth set of elements: timestamps which are assigned to tag-resource pairs in order to consider temporal aspects in their analysis.

2.3 Tagging Dynamics

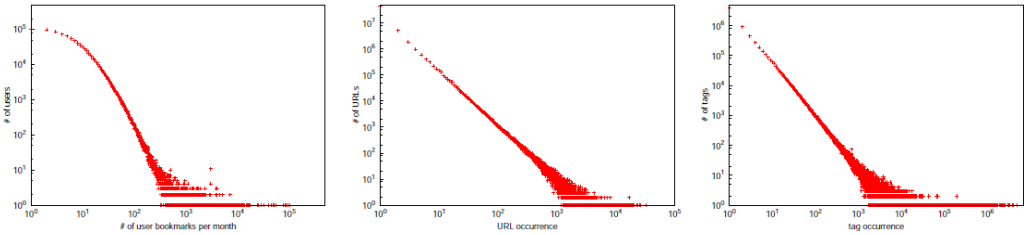
From a network analysis perspective, "a folksonomy can be considered as a tripartite undirected hypergraph $G = (V, E)$" connecting users, tags and resources [Jäschke, 2011]. $V = U \dot{\cup} T \dot{\cup} R$ is the set of nodes, and $E = \{\{u, t, r\} \mid (u, t, r) \in Y\}$ is the set of hyperedges. Those hypergraphs show interesting properties which help to understand a folksonomy's structure.
In order to gain a better understanding of the basic properties of complex networks, especially scale-free and small-world networks, this section will present the main concepts and characteristics.
Power Law Distributions and Scale-free Networks. A *power law* describes the phenomena where highly connected nodes in a network are rare while less connected nodes are common [Adamic, 2002]. It indicates that the probability $P(k)$ that a vertex in the network is connected to k other vertices decays according to $P(k) \sim k^{-\gamma}$ [Barabasi and Albert, 1999] where $\gamma > 0$ is a constant and $\sim$ means asymptotically proportional to as $k \to \infty$. The distribution of a power-law is highly skewed having a *long tail*, which means that the probability of selecting a node which has less connections than the average is high. According to Willinge et al. [2009] "most nodes have small degree, a few nodes have very high degree, with the result that the average node degree is essentially non-informative". Plotted on a log-log scale power-law distributions will appear as a straight line with the gradient $-\gamma$.
Scale-free networks refer to networks which have a power-law degree distribution. No matter how many nodes the network consists of, the characteristic constant (γ) does not change, which makes such networks "scale-free" [Newman, 2005].
Various network structures induced from the structure of folksonomies exhibit power-law distributions which can be produced by scale-free networks. Especially the distribution of tag usage and tag co-occurrence have been carefully examined in respect to power laws. Based on such findings, different statements about the user behaviour and the overall network dynamics can be made.

- Quintarelli [2005] mentions the power law distribution of tag usage in broad folksonomies. He states that the "power law reveals that many people agree on using a few popular tags but also that smaller groups often prefer less known terms to describe their items of interest."
- Halpin et al. [2006] show, that for a small dataset of 100 URLs which were tagged at least 1000 times and their 25 most popular tags, the tag usage frequency follows a power law. The authors conclude that "the distribution of tag frequencies stabilizes [into power laws]", which indicates that users tend to agree about which tags optimally describe a particular resource.
- Cattuto et al. [2006] examine tag co-occurrence using some preselected tags. They found a power law decay of the frequency-rank curves for high ranks while the curve for lower-ranked parts was flatter.
- Wetzker et al. [2008] and Li et al. [2008] analyse the distributions of tags per post, users per post and bookmarks per post in the social bookmarking system Delicious. Their results are similar to our results in Section 6.4.2 where we compare folksonomies to the tripartite structure of clickdata. While the occurrence of tags and URLs

follow a power law distribution, the distribution of users is less straight (see Figure 2.2). Still, one can observe that many users have only few posts, while a few users hold many posts.



(a) Frequency distributions of Li et al. [2008]. The figures show the frequency distributions of URLs, users and tags.



(a) user bookmarks per month (b) bookmarks per URL (c) tag occurrence

(b) Frequency distributions of Wetzker et al. [2008] in a slightly different order. The figures show the frequency distributions of users, URLs and tags.

Figure 2.2: Frequency distributions of Li et al. [2008] and Wetzker et al. [2008]. Please note that, while the order of the figures and the scale of the datasets used for the experiments is slightly different, the distributions themselves are similar.

Small world networks. Small world networks are types of graphs where most nodes can be reached by a small number of steps. Such networks have a relatively shortest path between any two nodes in the network and exhibit significant higher clustering coefficients than random networks where nodes are connected randomly [Watts and Strogatz, 1998].
The shortest path length measures the average distance between any two nodes in the graph. The clustering coefficient quantifies the extent to which a node's immediate neighborhood is completely connected. The local clustering coefficient of an undirected graph is given by

$$C_i = \frac{2E_i}{k_i(k_i - 1)} \tag{2.1}$$

where E_i is the number of edges connecting direct neighbours of node i, k_i is the degree of node i and $k_i(k_i - 1)$ is the number of possible edges among the immediate neighbours of node i.
Cattuto et al. [2007] analyse the network properties of folksonomies. They adjust various measures (i. e., the average shortest path length and the clustering coefficient defined above) to the tripartite structure of folksonomies and demonstrate small world characteristics in this type of graph. In Section 6.4.2 we will compare the properties of networks

inferred from folksonomies and data from log files based on the measures defined in [Cattuto et al., 2007].

Preferential attachment. What are the underlying models explaining the stabilization of tag distributions (into a power law)? Different assumptions are currently being discussed. The major one is the phenomenon of *preferential attachment*, where users imitate each other either directly by looking at each other's resources or indirectly by accepting tags suggested by tag recommenders. In this "rich get richer" approach [Barabasi and Bonabeau, 2003], a newly added node preferentially connects to a node which already has high network degree. In respect to tagging systems, a tag that has already been used to describe a resources is more likely to be added again [Halpin et al., 2007].

Various tag generation models have been introduced since to describe the tagging process[for a survey see Gupta et al., 2010]. For example, users do not only imitate each other. They seem to have a similar understanding of a resource due to sharing a similar background. Based on this thesis Dellschaft and Staab [2008] present a generative tagging model integrating the two perspectives: a new tag is assigned to a resource either by imitating a previous tag assignment or by selecting a tag from the user's active vocabulary. When choosing tags users can also be influenced by external factors. Lipczak and Milios [2010], for example, analysed the influence of a resource's title on tagging behaviour and found that tags which appear in a resource's title are more often used than other tags with the same meaning.

Research about network structure and tagging dynamics has helped to understand the creation of folksonomies. It can be used to improve mining algorithms such as tag recommender systems (see Section 2.5.2) and to develop methods to make the implicit knowlegde and structure of a folksonomy explicit.

2.4 Collaborative Tagging and the Semantic Web

Collaborative tagging systems are observed with interest by the Semantic Web community. Its members hope that the uncontrolled vocabulary of a folksonomy can be used to complement and enhance the creation of formal knowledge representations for the Semantic Web (for example [Mika, 2005]).

While tagging systems based on folksonomies can be seen as a collection of terms with an inherent structure, the Semantic Web builds on a formal representation of knowledge by means of structured taxonomies.

The creation of such taxonomies can be tedious and awkward: Its definition and use requires domain experts, the possibility of categorizing the domain, stable and restricted entities and a clear coordination [Shirky, 2005]. In contrast, the process of collaborative tagging creates an emergent, unstatic and non-hierarchical classification system consisting of "idiosyncratically personal categories and those that are widely agreed upon" [Golder and Huberman, 2005]. The question of interest for the Semantic Web community is whether or not the collective tagging effort resulting in a folksonomy can be used as a bottom-up approach to generate more formal knowledge representations.

The difficulty of inferring semantics from the emerging vocabulary of a tagging system is the linguistic variety and freedom in the tagging process (already mentioned in the description of system weaknesses in Section 2.1.2). Common problems are the resolution of polysemy (one word having many meanings) and synonymy (multiple words have the

same meaning). A variety of works analyse the vocabulary of social tagging systems and deal with linguistic challenges (for example Golder and Huberman [2005]; Halpin et al. [2007]; Marlow et al. [2006]).
The identification of related tags can be considered as a first step towards the creation of more formal knowledge from folksonomies. The results can be used for synonymy detection, ontology learning and also related fields dependent on the underlying semantics of folksonomies, such as tag recommendation or query expansion. In Cattuto et al. [2008] the authors examine several distance measures based on co-occurrence, cosine similarity and the FolkRank algorithm to identify similar tags in a folksonomy. By comparing their results to the semantic distance of word pairs in WordNet [Fellbaum, 1998a], they can better characterize the different methods. For example, tag co-occurrence similarities tend to yield synonyms, while related tags computed from the FolkRank relatedness include more general tags. A further examination of tag relatedness measurement is given in Markines et al. [2009b]. The authors evaluate several similarity measures from information-theory, statistical, and practical fields using different aggregation methods to deal with the tripartite data. Similar to Cattuto et al. [2008], the evaluation is based on a comparison of word distances of the tags in WordNet and the Open Directory Project.
Further approaches concentrate on the creation of a hierarchical structure inferred from the tags of a folksonomy. Schmitz [2006] uses a machine learning approach to identify is-a relationships between tags in Flickr. In Schmitz et al. [2006] the authors create relations between tags by applying association rule mining. Their associations allow the probabilistic determination of further tags if a certain tag has been added to a resource. Zhou et al. [2008b] apply a divisive hierarchical clustering algorithm to construct tag hierarchies in the tagging systems Delicious and Flickr. Sacco and Bothorel [2010] use a graph clustering technique to represent the hierarchical structure of tags and transform this structure into a semantic format. The authors of Strohmaier et al. [2012] evaluate three folksonomy induction algorithms creating a hierarchical structure from tagging data. They can show that induction algorithms outperform traditional hierarchical clustering methods.
External resources such as other social media sites or existing ontologies can also help to infer semantics from tagging systems. For example, the authors of [Damme et al., 2007] propose the integration of multiple online lexical resources such as WordNet or Google, different methods of co-occurrence analysis and ontology mapping approaches. Specia and Motta [2007] use existing ontologies available on the Web to enhance the semantics of tags. Corcho et al. [2012] use DBpedia to define the meaning of a tag. External knowledge can also be introduced by users in order to structure the tags. For example, Garcia-Castro et al. [2009] enable users to assign tags to tags. A complete cycle from (automatically) extracting semantic relations between tags over users adding and refining relations to enhancing the folksonomy with semantic assertions is presented in Limpens et al. [2010]. Monachesi and Markus [2010] enrich existing ontologies with data extracted and processed from social media sites.

2.5 Mining Tagging Data

The following sections will briefly review the state-of-the-art in mining folksonomy data. What kind of patterns can be discovered for such data and how can it be used for improving a system's service? We consider three different mining applications: ranking, recommen-

dation and community detection. A forth could be part of this section – spam detection. However, as this field is one of the major research parts of this thesis, it will be introduced in detail in Chapter 4.

2.5.1 Ranking

In order to find relevant information from the tremendous amount of pages on the Web, search applications need to find related pages and bring order to the search results. The heart of such applications are ranking algorithms which can cope with the amount of data on the Internet, its different formats and varying quality. Traditional IR-techniques such as the vector space model [Manning et al., 2008] cannot handle those challenges. Due to their reliance on the occurrence of terms in the documents, they tend to "fall for" spam pages stuffed with keywords.
Two algorithms (and many variations developed afterwards) proposed in the 1990s deal with the challenge of Web page ranking by considering the hyperlink structure of the Web: PageRank [Brin and Page, 1998] and the Hyperlink-Induced Topic Search (HITS) algorithm [Kleinberg, 1999a]. Both algorithms model the Web as a network where nodes correspond to web pages and a directed edge between two nodes exists if one page has a hyperlink to the second one. Such direct links express importance. The more a page is linked by others, the more important it is.
The search algorithms developed for retrieving information from a folksonomy are based on PageRank and HITS. In this thesis, we use the one prominent ranking algorithm for folksonomies: the FolkRank algorithm (see Chapter 6). As it is based on the PageRank algorithm, this section will provide a more detailed view of both methods. To get an overall picture of ranking algorithms in folksonomies, further approaches are introduced briefly thereafter.

PageRank

The PageRank algorithm [Brin and Page, 1998; Page et al., 1999] is the foundation of the popular search engine Google [8]. The algorithm models the behaviour of a random Web surfer who randomly follows a link without showing any preferences for specific pages. Consequently, all links on a page have equal probability of being followed. Periodically, the random surfer does not follow the offered links but jumps to a randomly selected page. The random surfer model can be expressed by means of a directed graph $G = (V, E)$. V corresponds to the set of nodes representing the web pages and E represents the set of ordered pairs (i, j), called edges, corresponding to the links between the web pages. $(i, j) \in E$ if the node i links to node j. One can further define $out - degree(i)$ as the number of edges outgoing from i. One can construct a row stochastic adjacency matrix A (also called *link matrix*) by setting a_{ij} as follows:

$$a_{ij} = \begin{cases} \frac{1}{\text{out}-\text{degree}(i)} & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

The rank of a node in the graph can be calculated by means of the weight spreading computation

$$\mathbf{w}_{t+1} \leftarrow dA^{\mathrm{T}}\mathbf{w}_t + (1-d)\mathbf{p} \ , \tag{2.3}$$

[8] http://www.google.de

where $\mathbf{w}$ is a weight vector with one entry for each node in V and $\mathbf{p}$ is the random surfer vector which follows a uniform distribution. $d \in [0, 1]$ is determining the strength of the influence of $\mathbf{p}$. Page et al. [1999] suggest to set d to 0.85. By normalizing the vector $\mathbf{p}$, one enforces the equality $||\mathbf{w}||_1 = ||\mathbf{p}||_1$.

The FolkRank Algorithm

Several adaptations of the PageRank algorithm to the folksonomy structure exist. The one used in this thesis is the FolkRank algorithm as presented in [Benz et al., 2010a; Hotho et al., 2006a].
The algorithm operates on an undirected, tripartite graph $G_{\mathbb{F}} = (V, E)$, where $V = U \dot{\cup} T \dot{\cup} R$ and the set of edges E results from splitting tag assignments into three undirected edges each, i. e., $E = \{\{u, t\}, \{t, r\}, \{u, r\} \mid (u, t, r) \in Y\}$.
The PageRank formula as introduced in Section 2.5.1 can then be iteratively applied to the folksonomy graph.

$$\mathbf{w_{t+1}} = dA^T\mathbf{w_t} + (1 - d)\mathbf{p},$$

where $\mathbf{p}$ is the random surfer vector (which we select as preference vector) and $d \in [0, 1]$ is a constant which controls the influence of the random surfer. A is the row-stochastic version of the adjacency matrix of $G_{\mathbb{F}}$.
One can specify preference weights which are set in the preference vector $\mathbf{p}$ in order to compute a ranking of tags, resources and/or users tailored to the preferred item. In the case of web search, the tags representing search terms receive a higher weight compared to the remaining items (i. e., remaining tags, all users and all resources) whose weight scores follow an equal distribution. Overall, the equation $||\mathbf{w}||_1 = ||\mathbf{p}||_1$ needs to hold. The algorithm is outlined in Algorithm 2.1.

Algorithm 2.1: FolkRank

Input: Undirected, tripartite graph G_F, a randomly chosen baseline vector $\mathbf{w_0}$ and a randomly chosen FolkRank vector $\mathbf{w_1}$.

1: Set preference vector $\mathbf{p}$.
2: Compute baseline vector $\mathbf{w_0}$ with $\mathbf{p} = \mathbf{1}$ and $\mathbf{1} = [1, \ldots, 1]^T$.
3: Compute topic specific vector $\mathbf{w_1}$ with specific preference vector $\mathbf{p}$.
4: $\mathbf{w} := \mathbf{w_1} - \mathbf{w_0}$ is the final weight vector.

Output: FolkRank vector $\mathbf{w}$.

As can be seen in Algorithm 2.1, the computation consists of two runs: First, a baseline with a uniform preference vector needs to be computed. The result of this iteration is the fixed point $\mathbf{w_0}$. Second, the fixed point $\mathbf{w_1}$ is computed by setting a preference vector. The final weight vector for a specific search term is then $\mathbf{w} := \mathbf{w_1} - \mathbf{w_0}$. The subtraction of the baseline reinforces the items which are close to the preferred items, while it degrades items which are popular in general.

Social PageRank

The Social PageRank algorithm was introduced in Bao et al. [2007]. Both the FolkRank and Social PageRank algorithm are based on spreading weights along the link structure

of the folksonomy graph. The difference concerns the path a random surfer can follow. While FolkRank allows all sorts of paths through the tripartite network, SocialPageRank restricts possible paths to resource-user-tag-resource-tag-user combinations [Abel et al., 2008]. We use the notation of Abel et al. [2008] to present the algorithm.

Algorithm 2.2: Social PageRank

input : Association matrices A_{TR}, A_{RU}, A_{UT} , and a randomly chosen SocialPageRank vector $\mathbf{w_{r_0}}$

until $\mathbf{w_{r_0}}$ converges do: $\mathbf{w_{u_i}} = A_{RU}^T * \mathbf{w_{r_i}}$

$\mathbf{w_{t_i}} = A_{UT}^T * \mathbf{w_{u_i}}$

$\mathbf{w'_{r_i}} = A_{TR}^T * \mathbf{w_{t_i}}$

$\mathbf{w'_{t_i}} = A_{TR} * \mathbf{w'_{t_i}}$

$\mathbf{w'_{u_i}} = A_{UT} * \mathbf{w'_{u_i}}$

$\mathbf{w_{r_{i+1}}} = A_{RU} * \mathbf{w'_{r_i}}$

output : SocialPageRank vector $\mathbf{w_r}$.

The same authors also proposed the SocialSimRank algorithm [Bao et al., 2007], which is based on SimRank. This algorithm is used to calculate similarity between items based on the resources they were assigned to.

Adjusted versions of the HITS algorithm to a folksonomy

The HITS algorithm [Kleinberg, 1999a] has also been adjusted to rank items in a folksonomy. Two versions exist which have been called by Abel et al. [2008] *Naive* HITS and *Social* HITS.

The challenge of using HITS to rank resources in a folksonomy is the transformation of the undirected tripartite graph to a directed graph. The two algorithms above are based on the transformation proposed by Wu et al. [2006b]: A tag assignment $(u, t, r) \in Y$ is split into two edges $u \rightarrow t$ and $t \rightarrow r$. The resulting structure is a directed graph where hubs are users and authorities are resources (as resources have no outgoing links their hub weights become 0). While the *naive* HITS implementation uses this structure, *social* HITS extends the graph by allowing for authority users and hub resources. Given a tag assignment $(u, t, r) \in Y$ they derive two directed edges from user actions: $u \rightarrow t$ and $u \rightarrow r$. Additionally, they create an edge $u_h \rightarrow u_a$, whenever an arbitrary user u_h annotated a resource after it had already been tagged by user u_a.

2.5.2 Recommender Systems

Recommender systems are concerned with the identification of items which match the interests of a specific user. To find those items, a variety of information sources related to both the user and the content items are considered, for example history of purchases, ratings, clickdata from logfiles or demographic information [Jäschke et al., 2009]. Typical domains for recommender systems are online shopping systems (recommendations of certain products such as books at Amazon.com [Linden et al., 2003]), social networks (proposition of people one might know [Chen et al., 2009b]) or multimedia pages (music or movie recommendations).

One of the most prominent recommender algorithms is collaborative filtering. The approach creates user profiles based on user preferences, behaviour or a user's demographic situation. In order to recommend appropriate items, user profiles are compared. Items of user profiles most similar to the one for whom a recommendation shall be made are then selected.
A second prominent approach is content based filtering, which processes the information about an item in order to find other similar ones. Content based filtering methods generate an item-item matrix showing the similarity between pairs of items. Again, the most similar ones can be used for recommendation. The two approaches (collaborative filtering and content based filtering) can be combined in so-called hybrid approaches. The algorithms used for calculating the similarities vary from simple statistic calculations over graph based computations to data mining procedures (see [Adomavicius and Tuzhilin, 2005] for an overview).
The three dimensions of tagging systems allow for multi-mode recommendations, e. g., finding resources, tags, or users [Marinho et al., 2011]. Similar to other recommendation settings, recommender methods in tagging systems need to deal with common problems such as the cold start problem where it is difficult to propose items to new users, the sparsity of the data or the ability of real-time recommendations. For each of the three dimensions we will briefly summarize relevant research.

Tag Recommendations

In order to support users when assigning tags to a resource, many tagging systems recommend tags based on recent user and resource information. The input of tag recommender algorithms are pairs of user and resources. The output is a set of tags, T_r which are the top n tags resulting from computing a weight $w(u, r, t)$ for each tag and a specific user-document pair [Gemmell et al., 2009].
In 2008 and 2009 two tag recommendation challenges were conducted at the ECML/PKDD conference [Eisterlehner et al., 2009; Hotho et al., 2008]. In 2009, a tag recommender framework was developed for BibSonomy. Besides a flexible integration of different recommendation strategies, the framework tracks all stages of the recommendation process and offers evaluation capabilities [Jäschke et al., 2009].
The best tag recommendation approach based on the test results of the challenge Jäschke et al. [2007, 2008] is the FolkRank algorithm (see Section 2.5.1) adjusted to the tag recommendation scenario. It outperforms several baselines such as most-popular models and collaborative filtering algorithms. Its weakness is a slow prediction runtime which makes realtime computations for tag recommendations difficult.
Many more efforts in respect to improving tag recommendation systems have been made since then. The authors of Zhang et al. [2011] give an overview of recent works in the field of network-based methods, tensor-based methods and topic-based methods. Furthermore, in subsequent publications, similar or better results than produced by the FolkRank algorithm could be presented. For example, Cai et al. [2011] use low-order tensor decompositions to achieve slightly better experimental results. Another method is to better reflect human behaviour, for example by introducing a time-dependent forgetting process as humans are influenced more by recent activities (i. e., used tags) than by activities carried out a while ago [Kowald et al., 2015].
Finally, besides the testing platform of [Jäschke et al., 2009] several frameworks have been

introduced [Domínguez García et al., 2012; Kowald et al., 2014] in order to facilitate the implementation and evaluation of recommender algorithms.

Resource Recommendations

Resource recommendation in social tagging systems can be carried out by compiling a resource list either based on profile information of a specific user, a query (tags) or both. The task is similar to ranking resources in a folksonomy (see Section 2.5.1). The different approaches proposed in this field are difficult to compare as different datasets, evaluation methods and measures have been used.
For instance, Niwa et al. [2006] present a resource recommendation system based on user interests modeled with tags. Each user is assigned to a tag cluster. Resource propositions are generated by comparing the similarity of the resource's tag to the user's tag cluster. Gemmell et al. [2008] propose a hierarchical agglomerative clustering technique to compute resource recommendations triggered to a specific user. The recommendations are used to present a user a personalized list of resources after having clicked on an arbitrary tag. Stoyanovich et al. [2008] explore various methods to produce "hotlists" – resource lists customized for users. As information source they use tags as well as explicitly stated or derived social ties. Wetzker et al. [2009] use a probabilistic latent semantic analysis approach whereby a topic model is built from resource-tag and resource-user co-occurrences. Using the two distributions, the authors benefit from a more collaborative filtering based approach (user-resource distribution) as well as a content filtering based approach (tag-resource distribution). In Cantador et al. [2010], content-based recommendation algorithms based on the Vector Space and Okapi BM25 ([Manning et al., 2008]) retrieval models are built from user and item profiles of tags. Guan et al. [2010] develop a graph-based algorithm. They create a semantic space of users, resources and tags where related objects appear close to each other. The documents closest to the user and not tagged by him or her are then recommended. The authors of Doerfel et al. [2012] propose scientific resources given a user profile in BibSonomy. They extend the FolkRank algorithm (see previous paragraph) by including a fourth dimension – user group information – and by manipulating the preference vector so that higher preference is given to similar users, recently posted resources or popular resources. A combination of different recommender algorithms is proposed by Gemmell et al. [2012]. They later extend their algorithm by partitioning users into different groups and computing different weights for each of these groups [Gemmell et al., 2014].

User Recommendations

User recommendations can be obtained either by considering the social relationships between users (i. e., if user A connects to user B and C, user B and C might be interested in each other as well) or by considering their shared resources or tags.
Symeonidis [2009] perform latent semantic analysis and dimensionality reduction using a 3-order tensor which models the three dimensions user, tags and resources. They evaluate their algorithm on part of the BibSonomy dataset by comparing similarities of the documents of users found by their method and by a simple baseline finding similar users based on shared tags. Zhou et al. [2010] present a user recommendation framework where user interests are modeled based on tag co-occurrence information. The profiles are rep-

resented by discrete topic distributions. By means of the Kullback-Leibler divergence similarities between user topic distributions can be measured. The framework's evaluation is conducted on a Delicious dataset. Manca et al. [2015] recommend users based on a combination of similarity scores of the Pearson correlation coefficient between tag vectors and the percentage of resources shared.
The social bookmarking system BibSonomy recommends users based on the Folk-Rank algorithm (see Section 2.5.1). If the recommended users are of interest, the user can be added to a special list: the follower list. The associated followers page then shows all recent posts of the follower list's users [9].
The task of user recommendation can also be seen as a form of community detection: similar users are grouped together. A review of recent community detection approaches in folksonomies can be found below in Section 2.5.3.

2.5.3 Community Detection

Detecting clusters or communities in large real-world graphs such as large social or information networks is a problem of considerable interest. Most approaches to detect communities either consider the tripartite hypergraph or projections of it. For instance, Nair and Dua [2012] build bipartite tag-resource graphs for a specific user, join them and transform them into uniform tag graphs. Liu and Murata [2011]; Neubauer and Obermayer [2011] present community detection algorithms suitable for the tripartite hypergraph of a folksonomy.
Most users of tagging systems can not be fully assigned to one specific user group but have multiple topical interests with different communities. Several studies deal with such overlapping community memberships, i. e., nodes can be assigned to more than one cluster. For instance, Schifanella et al. [2010] group users according to their tagging behaviour. Wang et al. [2010] built a bipartite graph of users and tags and cluster the edges of this graph. Overlapping communities are then built by grouping all user nodes who are part of edges belonging to one cluster. Chakraborty et al. [2012] apply link clustering algorithms to the tripartite structure of a folksonomy graph by building a weighted line graph whereby the hyperedges of the folksonomy graph are nodes, and nodes are connected if two hyperedges have at least one common node in the folksonomy graph. A challenge of community detection algorithms is their evaluation since the size of the datasets makes it difficult for humans to create a 'ground truth' where each user is assigned a membership to one or more interest groups. One method is to use synthetic data [for example Chakraborty et al., 2012]. Another method to overcome this difficulty is to use the existing links between users of a social network. The idea is to compute clusters built from the implicit social connections (for example the shared tags and resources) and then compare the extracted communities to explicit social links between users of the network [for example Chakraborty et al., 2012; Ghosh et al., 2011b; Wang et al., 2010]. Mitzlaff et al. [2011] propose a set of "evidence-networks" reflecting typical user interactions in a tagging system. These networks can be used as an approximation of (explicit) user groups. In [Mitzlaff et al., 2014] the authors present experiments showing that users in such evidence-networks tend to be semantically related. Neubauer and Obermayer [2011] present an interactive tool visualizing the clustering tree. Besides the possibility of navigation the tool can be used to compare results of

[9] `http://www.bibsonomy.org/help_en/Followers`

different clustering algorithms.

2.6 Example Systems

Prominent examples of collaborative tagging systems are Delicious [10] for bookmarks, Connotea [11] and CiteULike [12] for publication metadata, BibSonomy [13] for bookmarks and publication metadata, Flickr [14] for photos or YouTube [15] for videos. The process of tagging has been included into many websites. Examples are Technorati [16] (weblog posts) or Twitter [17] (micromessaging posts). As the datasets used in this thesis have been generated from the two systems Delicious and BibSonomy, we will concentrate on a detailed description of these.

BibSonomy

BibSonomy was introduced in 2006 [Hotho et al., 2006b]. The social bookmarking system is hosted by the Knowledge Engineering Group at the University of Kassel and the Data Mining and Information Retrieval Group at the University of Würzburg. The target user group are university users including students, teachers and scientists. As their work requires both the collection of relevant web links and the collection of relevant publications, BibSonomy combines the management of both types of resources. Hence, users can either post web links or publication references.
As of April 2011 the system has about 6700 active users which share about 380.000 bookmarks and 580.000 publication metadata entries. Additionally, the system contains about one million publications and 20.000 homepages of research workshops or persons, which have been automatically copied from the computer science library DBLP [18].
Further system features were developed to support researchers in their daily work, e. g., finding relevant information, storing and structuring information, managing references and creating publication lists be it for a diploma thesis, a research paper or the website of the research group. BibSonomy also promotes social interactions between users by offering friend connections and the possibility to follow the posts of other users. A more complete description of the system's features can be found in Benz et al. [2009a]; Hotho et al. [2006b].
In Benz et al. [2010a], BibSonomy was used as a research platform for the research group of the Knowledge and Data Engineering team of Kassel. The team conducted experiments concerning different aspects of data mining and analysis including network properties, semantic characteristics, recommender systems, search and spam detection.
Finally, BibSonomy offers system snapshots to other researchers in order to support investigations about tagging data. In two challenges (ECML/PKDD discovery challenge

[10] http://del.icio.us
[11] http://www.connotea.org/ - as of March 2013, the service stopped.
[12] http://www.citeulike.org
[13] http://www.bibsonomy.org
[14] http://www.flickr.com
[15] http://www.youtube.com
[16] http://technorati.com/
[17] https://twitter.com/
[18] www.dblp.uni-trier.de

2008 and 2009) BibSonomy data was used (see Section 7.4 for a further description). Several papers were published using those datasets, among them Papadopoulos et al. [2010] exploring the semantics of tagging systems, Papadopoulos et al. [2011] analysing communities, Belém et al. [2014]; Djuana et al. [2014]; Jin et al. [2010]; Peng et al. [2010]; Rendle and Schmidt-Thieme [2010]; Yin et al. [2011] building recommender services, Ignatov et al. [2014]; Markines et al. [2009a]; Neubauer and Obermayer [2009]; Neubauer et al. [2009]; Yazdani et al. [2012a] creating features and algorithms for spam detection. Several of the mentioned papers have been discussed in the context of tagging system research in this chapter or will be discussed in the following chapters (especially research about spam detection in Chapter 4).

Delicious

One of the first social bookmarking systems to become popular was *Delicious*. The system, founded by Joshua Schachter, went online in September 2003 [19]. It arouse from a system called Memepool in which Schachter simply collected interesting bookmarks. Over time, users sent him more and more interesting links so that he wrote an application (Muxway) which allowed him to organise his links with short labels - tags. He then realised that not only him, but other internet users might be interested in organizing and sharing their internet links with the help of tags - and rewrote Muxway so that it became the website Delicious. Soon, Delicious became very popular. From December 2005 it was operated by Yahoo! Inc.. As the system did not provide financial benefits for the company, it was sold to the company AVOS which was startet by the founders of YouTube, Chad Hurley and Steve Chen [20] in 2011. Since then, several aspects of the system have been redesigned in order to introduce more social features into the system [21]. In May 2014 they sold the system to Science Inc., a Californian technology investment and advisery firm [22].

[19] http://www.technologyreview.com/tr35/profile.aspx?trid=432

[20] http://techcrunch.com/2011/04/27/yahoo-sells-delicious-to-youtube-founders/

[21] http://mashable.com/2012/10/04/delicious-redesign/

[22] http://mashable.com/2014/05/08/delicious-acquired-science-inc/

Chapter 3

Social Information Retrieval

In recent online discussions and literature, the term *social information retrieval* was used to describe approaches integrating user behaviour and user interactions into the search process. In this chapter, we want to analyze this trend. We will first discuss what *social information retrieval* stands for (Section 3.1). In the following, we will depict specific topics of social information retrieval which are relevant as background information for the experiments concerning the comparison of social bookmarking and online search systems presented in Chapter 6. This includes a review of clickdata, its usage in learning-to-rank scenarios and folksonomy like structures built from clickdata (Section 3.2) as well as an analysis about how tagging data can leverage information retrieval tasks (Section 3.3).

3.1 Characterizing Social Information Retrieval

Traditional information retrieval methods have focused on document-query matching approaches or — with the advent of hypertext — on link analysis. Up to this point, search has been viewed as a single user action: people submit keywords in a search engine and get more or less relevant results.
During recent years, interests broadened towards exploiting these user (inter)-actions. Researches and developers became aware of the fact that users actively participate in the search process, for example, by entering specific search queries which other people also entered or by clicking on specific search results. Furthermore, they form communities by sharing interests, interacting with each other and influencing one another.
This social perspective of search has become an emerging research focus in the last years. A commonly agreed upon definition does not exist, but different aspects are considered when referring to social search. A very general definition, including the most important aspects is presented by Evans and Chi [2010]:

> *Social search* is an umbrella term used to describe search acts that make use of social interactions with others. These interactions may be explicit or implicit, co-located or remote, synchronous or asynchronous.

Croft et al. [2010] mention similar characteristics: Social search involves “communities of users actively participating in the search process. It goes beyond classical search tasks”

by allowing "users to interact with the system" and to collaborate "with other users either implicitly or explicitly".
The application areas concerned with social information retrieval vary among different authors. The most prominent ones are collaborative searching, collaborative filtering, tagging, social network search, community search and question answering systems. In this thesis we focus on *collaborative search*. Morris and Horvitz [2007] classified collaborative search systems into different categories depending on their focus.

Sensemaking systems focus on processing and organizing information in order to make it understandable. Supportive tasks can be commenting on functions or the combination of different text pieces. For example, Google Notebook [1], an online service, allows the creation of documents ("notebooks") in which different information sources (text, links or images) can be brought together and enhanced with user-written text.

Multi User Web Browsing allows several users to process online information by providing a collaborative interface. Groups can view other member's navigation paths, set pointers to an important web page or comment on jointly-viewed web pages. For instance, the system SearchTogether [2] allows users to see the query terms other users in their group submitted or view a summary of all pages that have been rated or commented on by other group members.

Multi-User Search aims at enabling social activities in the search process itself. Such activities can vary from providing chatting possibilities during search to sharing of retrieved websites within a group or commenting on other people's searches. The instant messaging client Windows Live Messenger (now discontinued [Windows, 2013]) allowed to search during a chat. Chat participants could enter a query into a box, click on search and the search results would have been shown to all chat members.

Social bookmarking systems as discussed in the previous chapter enable different users to store bookmarks and share them with other users. Collaboration takes place by creating a common index and tagging information with descriptive tags which can be used for search.

Passive collaboration systems incorporate implicit information, inferred from user's interaction with the search engine. The most prominent examples of this group are search engines using query logs and clickthrough data to improve search.

In this thesis, we concentrate on analyses and methods for the last two categories: Social bookmarking and passive, collaboration systems. We are interested in the similarities and differences of both approaches and on how to combine the user knowledge of both systems (see Chapter 6).

[1] After July 2012, Google Notebook has been integrated into Google Docs at `https://drive.google.com/ob?usp=web_ww_intro`

[2] `http://research.microsoft.com/en-us/um/redmond/projects/searchtogether/tutorial.html`

3.2 Exploiting Clickdata

This section will review the exploitation of feedback data from passive collaboration systems, in our case from query log files of search engines. After introducing query logs, search engine evaluation with query logs will be discussed. This comprehends an analysis of the quality of clickdata, the generation of training data for learning-to-rank algorithms and Ranking SVM, a learning-to-rank algorithm trained with feedback generated from clickfiles. This background is needed in Section 6.5, where we compare the feedback of query logs to the feedback of tagging systems. Finally, the information of clickdata will be used in another way: With the users, their queries and clicked resources extracted from logfiles, logsonomies can be build. Their structure should be similar to the one of folksonomies. A comparison of folk- and logsonomies is presented in Section 6.4.

3.2.1 Query Log Basics

Search engine query logs record the interaction of users with web search engines. These interactions consist of communication exchanges which occur between searchers and search engines. Major search engines store millions of such exchanges per day. Different kinds of search data can be recorded, such as the "client computer's Internet Protocol (IP) address, user query, search engine access time and referrer site, among other fields" [Jansen, 2006]. Typical transaction log formats are access logs, referrer logs or extended logs.

Characterizing queries

Similar to the classification of tags presented in 2.1.3, queries have been characterized. A query refers to a string list of several terms submitted to a search engine [Jansen et al., 2000]. Searchers start with an *initial query*. Depending on their search success they may submit *modified queries*. Broder [2002] introduced three different types of queries:

Navigational queries: a user wants to reach a particular website

Transactional queries: a user wants to perform some web-mediated activity

Informational queries: a user needs to acquire some information

Broder [2002] manually classified queries of a log file of the web search engine AltaVista into his three defined categories. He found that most of them were informational. Broder's taxonomy is used or built upon by many following query log studies. For instance, Jansen et al. [2008] identify characteristics of each query category and propose an automatic classification method to sort different queries into the three categories.

3.2.2 Search Engine Evaluation with Query Logs

A practical usage of query logs is the evaluation and tuning of search algorithms. In order to decide whether an algorithm performs well, one needs a founded base ("ground truth") - mostly generated from explicit user feedback such as ratings. Many search engines compute their rankings with the help of automatic machine learning techniques, which are based on a ranking model generated from training- and test data. If clicks can be interpreted as preferences, such data can be automatically derived from query logs instead of having to create the dataset manually.

This section discusses the generation of training data from clickdata. First, the traditional approach (creating a gold standard) is presented. Section 3.2.2 reviews several studies dealing with quality issues of clickdata. Finally, different strategies to infer preferences from clickdata introduced by Joachims [2002] are assessed.

Traditional evaluation: The gold standard

A common approach to tune a search engine is to create a "gold standard" against which different ranking mechanisms can be evaluated. Such datasets are normally constructed by human annotators who rate a set of web pages retrieved for a specific query according to the perceived relevance.
The first test collection available was the *Cranfield* collection. It consists of 1398 abstracts of aerodynamics journal articles, 225 one- or two sentence topics and exhaustive relevance judgments of all (query, document) pairs [Manning et al., 2008]. As the collection is rather small, it is not used for evaluation purposes anymore. However, the systemantic way of constructing a test collection and conducting comparable, reproducible evaluations has pioneered the evaluation of ranking systems. The approach is now referred to as the *Cranfield paradigm*.
The *Text REtrieval Conference (TREC)* started in 1992 with the goal of creating a framework for evaluating retrieval tasks using large test collections. New collections and tasks are published annually. Anyone can participate by solving the tasks related to the different test collections. Results are then presented and discussed in the scope of an annual workshop. Over the years, different tasks were added including filtering, question answering, web, Chinese or spoken document retrieval. In order to respond to the changing data requirements, the "classic" collection of about 5 GB text of newswires or patents was extended by the TREC web collections.
Further test collections include the *NIST test document collections* [3], the *NII Test Collections for IR Systems (NTCIR)* [4], concentrating on east asian languages and cross-language retrieval, or the *Cross Language Evaluation Forum (CLEF)* [5] which focuses on European languages and cross-language retrieval.
The creation of a gold standard has several pitfalls.

- Different studies [Bailey et al., 2008; Voorhees, 1998] demonstrated, that human annotators do not always agree with each other.
- Furthermore, the evaluation of web search algorithms require high amounts of testing data. On top of that the creation of labels for such data is expensive and laboursome.
- Often, only few (or just one) judges are asked to label the data. The data may therefore be error prone and not reflect the preference perception of the majority of web search engine users.
- The creation of a document corpus is a static process. Changes in the document collection or in the focus of popular searches can not be considered [Dupret et al., 2007].

[3] `http://www.nist.gov/tac/data/data_desc.html`
[4] `http://research.nii.ac.jp/ntcir/index-en.html`
[5] `http://clef.isti.cnr.it/`

Alternatives to a gold standard evaluation have been considered and are presented in the following section.

Quality of clickdata

During the last few years, a new form of search engine evaluation has gained the researcher's and practionioner's attention: the automatic generation of labels inferred from the search engine's logfiles. The approach avoids some of the difficulties related to the manual creation of preferences: By automatically collecting user click data no human labels need to be generated. Furthermore, the continuous storage of user data allows keeping pace with changes in a collection.
Several studies about the practicability of using clickdata for evaluation purposes exist. They either compare feedback from clickdata to explicit relevance judgments [Fox et al., 2005; Joachims et al., 2007] or they look at differences between search engine rankings based on clicks or human labels [Kamps et al., 2009; Macdonald and Ounis, 2009; Xu et al., 2010].
The authors of Fox et al. [2005] compare explicit and implicit feedback by developing models based on implicit feedback indicators to predict explicit ratings of user interest in web searches. They developed a browser which enabled the collection of various implicit relevance indicators such as time and scrolling activities, which page was clicked or if a page was added to a user's favourites or printed. The browser also allowed storing explicit judgements of a web page's relevance and a user's degree of satisfaction with the entire search session. Based on the implicit indicators tracked for different users, Bayesian models and decision trees were built to predict the user's explicit ratings. It could be shown that a combination of clickthrough, time spent on the search result and how a user exited a result or ended a search session performed best.
The authors of Joachims et al. [2007] conducted an eyetracking user study to evaluate the expressiveness of clickdata. The authors asked 34 participants to answer 10 questions by using the search engine Google. The questions required either navigational or informational searches. With the help of an eyetracker, the movement of the user‘s eyes could be compared to their clicks. Users also had to give explicit relevance judgements. The key finding of Joachims et al. [2007] was that clicks can not be seen as an unbiased assessment of the absolute importance of a web page. As users normally read search results from the top to the bottom of a result list, a click can be rather considered as an indication of relative importance: the clicked web page is considered more important than the un-clicked results which appeared before it in the ranking [Silvestri, 2010].
Kamps et al. [2009] analysed the difference between clicks from query logs and explicit human judgements and compared rankings computed from the two approaches. They mapped queries from a MSN log file and a proxy log file to relevance judgments of an IR test collection of Wikipedia articles. One major difference they found was the number of relevant documents per topic (query): While topics derived from the query logs mostly have one to 13 relevant documents, topics from the manually labeled ad-hoc dataset contain, on average, 69 relevant documents. They also analysed rankings based on human and implicit judgements and found large differences between the two approaches.
Macdonald and Ounis [2009] propose different sampling strategies to create training data from clickthrough files. The training data can then be used to learn the parameters of a ranking function, in their case the parameters of a field-based weighting model. The

learned model is compared to a baseline system which is trained using a mixed set of TREC Web task queries. It can be demonstrated that the model inferred from click training data usually performed as good as the model derived from human assessed training data, sometimes even significantly better.
The authors of Xu et al. [2010] test the quality of clickdata in a slightly different setting. They propose using clicks in order to automatically detect errors in training data for learning-to-rank algorithms and improve their quality. They develop two discriminative models to predict labels. If a predicted label differs from a label assigned by humans, the human label is considered as an error. Experiments demonstrate that correcting erroneous labels with the help of click data helps to improve the performance of ranking algorithms. Geng et al. [2012] deal with the handling of noise produced by training examples generated from clickdata. They propose a graphical model which differentiates between true labels and observed labels. Their experiments show good results.

Automatic generation of training data

Training or evaluation data for ranking algorithms consists in several queries and for each query a list of documents labeled with a specific rating. One of the most popular works of how to generate such labels from clickdata is the one of Joachims [2002]. The author claims that clickdata can be interpreted as a form of relative feedback, whereby a clicked document indicates that it is more important than previous non-clicked documents. He based his proposition on a user study (see 3.2.2) which showed that search engine users tend to view search results from the top to the bottom of a result list. Based on the concept of relative importance, different strategies about how to infer labels from a result list of search results and corresponding clicks were defined. We will use the example of Silvestri [2010] to explain the different strategies. Let q be a query returning result pages p_1 to p_7. Suppose a user clicks on pages p_1, p_2, p_4, and p_7. This can be denoted as:

$$p_1^*; p_2^*; p_3; p_4^*; p_5; p_6; p_7^*$$

$rel(\cdot)$ is the function measuring the relevance of a page: $rel(p_i) > rel(p_j)$ means p_i is more relevant than p_j in the click-set C. Joachims et al. [2005] defined the following strategies for extracting feedback from such clicks. Some of them are adjusted to our experiments where we compare the performance of learning-to-rank algorithms from feedback generated from clickdata to the feedback generated from tagging data (see Section 6.5.1). The notation and definitions are taken from Joachims et al. [2005].

Strategy 1 (Click > Skip Above) *For a ranking $(p_1; p_2; \ldots)$ and a set C containing the ranks of the clicked-on links, extract a preference example $rel(p_i) > rel(p_j)$ for all pairs $1 < j < i$ with $i \in C$ and $j \notin C$.*

The strategy indicates that when users click on a document but ignore previous documents, those previous documents are not considered as relevant. The preference examples we can infer from strategy 1 are the following: $rel(p_4) > rel(p_3), rel(p_7) > rel(p_5), rel(p_7) > rel(p_3)$, and $rel(p_7) > rel(p_6)$.

Strategy 2 (Last Click > Skip Above) *For a ranking $(p_1; p_2; \ldots)$ and a set C containing the ranks of the clicked-on links, let $i \in C$ be the rank of the link that was clicked temporally last. Extract a preference example $rel(p_i) > rel(p_j)$ for all pairs $1 < j < i$ with $j \notin C$.*

This strategy assumes that a searcher who sequentially clicks on several result pages has neither been satisfied with the previous pages he or she clicked on nor with the previous document's abstract in the result list. The last clicked result therefore seems to be the most relevant one for the specific query. From the running example, using Strategy 2, the features $rel(p_7) > rel(p_6), rel(p_7) > rel(p_5)$, and $rel(p_7) > rel(p_3)$ are extracted.

Strategy 3 (Click > Earlier Click) *For a ranking* $(p_1; p_2; \ldots)$ *and a set* C *containing the ranks of the clicked-on links, let* $t(i), i \in C$ *be the time when the link was clicked. We extract a preference* $rel(p_i) > rel(p_j)$ *for all pairs* j *and* i *with* $t(i) > t(j)$.

Let assume that the pages of our example were clicked in this order: p_4, p_1, p_2, p_7. According to Strategy 3 the following preference relations can be extracted from this order: $rel(p1) > rel(p4), rel(p2) > rel(p4), rel(p2) > rel(p1), rel(p7) > rel(p4), rel(p7) > rel(p1)$, and $rel(p7) > rel(p2)$. Strategy 3 resembles the previous one in that later clicks seem to be more relevant results than earlier ones. On top, the current strategy does not consider the list's position, but it considers the temporal dimension: clicked documents later in time are more relevant than earlier clicks.

Strategy 4 (Last Click > Skip Previous) *For a ranking* $(p_1; p_2; \ldots)$ *and a set* C *containing the ranks of the clicked-on links, extract a preference example* $rel(p_i) > rel(p_{i-1})$ *for all* $i > 2$ *with* $i \in C$ *and* $(i-1) \notin C$.

The strategy assumes that abstracts which appear immediately above the clicked document have most probably also been evaluated (and not considered to be of relevance). Considering the running example, preference pairs $rel(p_4) > rel(p_3)$, and $rel(p_7) > rel(p_6)$ are extracted.

Strategy 5 (Click > No-Click Next) *For a ranking* $(p_1; p_2; \ldots)$ *and a set* C *containing the ranks of the clicked-on links, extract a preference example* $rel(p_i) > rel(p_{i+1})$ *for all* $i \in C$ *and* $(i+1) \notin C$.

Considering Strategy 5 for the last example, one can extract the preference pairs $rel(p_2) > rel(p_3)$, and $rel(p_4) > rel(p_5)$.

In contrast to Joachims [2002] who considered the extraction of user data from single user clickdata, Dou et al. [2008] propose the aggregation of user clicks and an extraction of pairwise preferences from such aggregated data. They verify their approach by analyzing the correlation between pairwise preferences extracted from clickthrough data and those extracted from data labeled by human judges. Their analysis shows that, generally, human judgements and click frequencies are only weakly correlated. They correlate stronger when unclicked documents are included in the creation of preference pairs. Further, they can show that the correlation becomes better when a) the click difference between query-document pairs is considered and b) only queries with a small entropy (i. e., navigational queries) are analysed.

Another work exploring the aggregation of user clicks is the one of Agichtein et al. [2006]. Their results confirm previous research showing that users are influenced in their click behaviour by the underlying ranking algorithm. Users tend to click more on the top search results even if those are not relevant. To correct this bias, they compute a prior background distribution which is the expected clickthrough for a result at a given position and subtract this from the observed clickthrough frequency at a given position. Based on the

"cleaned" distributions, they define different heuristic strategies similar to the ones presented in Joachims et al. [2007]. Additionally, the authors propose to learn a ranking function to derive relevance preferences from features based on implicit feedback information. In order to build such a model however, they need a labeled training set. Their evaluation using one indicates that the automatic extraction of relevance preferences performs significantly better than the heuristic strategies designed by humans.

3.2.3 Learning-to-Rank

Most machine learning techniques require a large training and test corpus. Because of this, it is difficult to apply such algorithms to ranking problems. As was shown in Section 3.2.2, only a few official test corpora exist, and even for search engines it is labourous and costly to generate new corpora. The discovery of clickdata as input for the generation of training and test corpora finally made the usage of machine learning techniques in ranking problems more feasable. Applied to a ranking setting, these methods are summarized under the term *learning-to-rank*. As has been noted in blogs and literature, for example Macdonald et al. [2013]; Sullivan [2005], not only the research community has embraced learning-to-rank methods, but also search engines are interested in this methods for delivering search results. Several datasets have been published from commercial search engines to support the development of ranking algorithms in this field. The LETOR (LEarning TO Rank) datasets have been released by Microsoft [6]. Yahoo released a dataset in the scope of their Learning to Rank challenge [7].
The basic concept of learning-to-rank methods resembles the one of other machine learning areas such as classification or clustering. In a first step, features need to be identified which characterize the importance of a document with respect to a specific query. In a second step, a training set corpus representing a subset of web documents needs to be generated. Relevance can either be decided manually or by using clickdata. Finally, the generated data is used to train a machine learning algorithm.

Definition

According to Liu [2011], learning-to-rank algorithms have two major properties: they are *feature based* and they are *discriminative*. A *feature* or *feature vector* is an n-dimensional vector $x = \Phi(d, q)$ where Φ represents a method to extract numeric characteristics from a document d associated with query q. Such features can be statistical information (query term frequencies in the document or the length of the document's title), HITs [Kleinberg, 1999b], PageRank [Page et al., 1999] computations or computations from probabilistic retrieval models. A good overview of possible features is given in the description of the LeTOR dataset, which is a popular dataset for evaluating learning-to-rank retrieval models [Liu et al., 2007].
Discriminative learning is characterized by the following four key components [Liu, 2011]:

Input space contains the objects under investigation; in learning-to-rank settings, these are mostly feature vectors.

[6] http://research.microsoft.com/en-us/um/beijing/projects/letor/
[7] http://learningtorankchallenge.yahoo.com/

Output space contains the learning target with respect to the input objects.

Hypothesis space defines the class of functions mapping the input space to the output space.

Loss function measures to what degree the prediction of the hypothesis is in accordance with the ground truth label.

Methods

One can divide existing methods for learning-to-rank into three categories [Liu, 2009]: the pointwise approach, the pairwise approach and the listwise approach. Each category differs in the way it defines the input, output and hypothesis space as well as the loss function.

Pointwise approach

The *pointwise approach* considers single documents (i. e., their feature vectors) as inputs. The hypothesis space includes so-called scoring functions producing a specific relevance score or label for each document (the output space). The different documents can then be sorted into a ranked list according to their scores.
Scoring functions are selected from three different machine learning areas:
regression-based algorithms ([Chu and Ghahramani, 2005], [Cossock and Zhang, 2006]), classification-based algorithms ([Li et al., 2007], [Nallapati, 2004]) and ordinal regression-based algorithms ([Gey, 1994]). The loss function considers each document and compares the ground truth label to the computed score/label.

Pairwise approach

The *pairwise approach* considers the feature vectors of document pairs as input for a specific query. The hypothesis space contains functions taking those input pairs and computing the relative order between them (output space). The loss function measures the gap between the computed relative order of two documents and their ground truth order.
Pairwise ranking has been introduced by Freund et al. [2003]; Herbrich et al. [1999]. It was further examined by Joachims [2002], who transformed the problem of classification support vector machines into ranking support vector machines (see the next section for more details). RankNet, which was Microsoft's Live search engine is based on a neural network approach [Burges et al., 2005].

Listwise approach

Finally, *listwise approaches* consider a set of document feature vectors for a specific query q. The output is an ordering (permutation) of those input documents. Typically, a scoring function f computes a score for each of the documents which allows ranking them in a descending order. By considering ranked lists as inputs and outputs, loss functions of the listwise approach can be based on information retrieval evaluation measures [Li, 2011a], as they take the ranked list and a ground truth list into consideration.
A detailed overview of algorithms using the listwise approach is presented in [Li, 2011a].

In this work we use a pairwise approach – the Ranking SVM [Joachims, 2002] – as the learning model (see Section 6.5.3) for our comparisons of tagging and clickdata. Ranking SVM allows a direct comparison of preferences derived from clickthrough data and preferences derived from tagging feedback. Furthermore, it has been widely used in previous work.

Ranking SVM

Support Vector Machines (SVM) can be applied for tasks such as classification, regression or ranking. The general idea is to transform input feature vectors into a vector space of higher dimension. Based on given training data, the algorithm constructs a hyperplane in the higher dimensional vector space which separates positive and negative examples. By means of a loss function, the hyperplane is optimized. SVMs for classification purposes are described briefly in Paragraph 4.2.2 in the context of spam detection.
Ranking SVMs as introduced by Herbrich et al. [1999]; Joachims [2002] allow documents to be ranked by classifying the order of pairs of documents [Li, 2011b].
Let $R*$ be a preference ranking of a set of documents with two document vectors $d_i, d_j \in R*$. Let f be a linear learning function and $\succ$ a relation indicating that d_i is in favour over d_j. Then, we can define

$$d_i \succ d_j \Rightarrow f(d_i) > f(d_j) \tag{3.1}$$

Let $\Phi(q, d)$ be a function which maps documents onto features to characterize the association of document d to query q. The function f is associated with the weight vector $\mathbf{w}$ as follows: $f(d) = \mathbf{w} \cdot \Phi(q, d)$ with

$$f(d_i) > f(d_j) \Leftrightarrow \mathbf{w} \cdot \Phi(q, d_i) > \mathbf{w} \cdot \Phi(q, d_j) \tag{3.2}$$

In order to allow some of the preference constraints to be violated, one can introduce a non-negative slack variable ξ_{ij}. The vector $\mathbf{w}$ can then be computed by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},\xi_{ij}} \quad & g(\mathbf{w}, \xi) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i,j}\xi_{ij} \\ \text{subject to} \quad & \forall(q, d_i, d_j) \in R* : \mathbf{w} \cdot \Phi(q, d_i) \geq \mathbf{w} \cdot \Phi(q, d_j) + 1 - \xi_{ij} \\ & \forall(i, j) : \xi_{ij} \geq 0 \end{aligned} \tag{3.3}$$

where C is a parameter that balances the size of the margin against the training error [Joachims, 2002].
If we set all ξ_{ij} to 0 (which assumes that the data is linearly separable), we can order the data points onto the weight vector $\mathbf{w}$. The ranking ("support") vectors are the vectors nearest to each other on the hyperplane. In the example of Figure 3.1 for weight vector $\mathbf{w}_2$, the closest points would be d_1 and d_4, with a distance between them of δ_2. In order to generalize $\mathbf{w}$, one needs to maximize the distance between the closest points, which can be computed as $\frac{\mathbf{w}(\Phi(q,d_i)-\Phi(q,d_j))}{||\mathbf{w}||}$.

3.2.4 Clickdata as a tripartite Network: Logsonomies

As logdata contain queries, clicks and session IDs, the classical dimensions of a folksonomy can be reflected: Queries or query words represent tags, session IDs correspond to

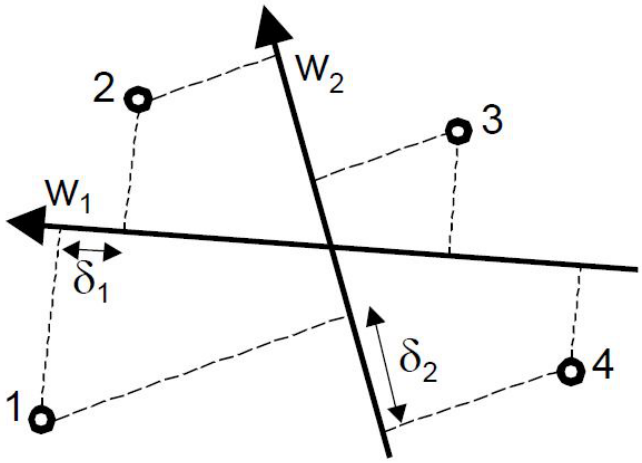


Figure 3.1: Example of two weight vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ ranking four points. Source: Joachims [2002]

users, and the URLs clicked by users can be considered as the resources that they tagged with the query words. Search engine users can then browse this data along the well known folksonomy dimensions of tags, users, and resources. This structure can be referred to as a "logsonomy". This section will define and characterize logsonomies based on the definitions and descriptions given in [Krause et al., 2008b]. A comparison of structural and semantic properties of logsonomies is presented in Chapter 6.4.

Logsonomy Construction

Let us consider the query log of a search engine. To map it to the three dimensions of a folksonomy, we set

- U to be the set of *users* of the search engine. Depending on how users in logs are tracked, a user is represented either by an anonymized user ID, or by a session ID.
- T to be the set of *queries* the users gave to the search engine (where one query either results in one tag, or will be split at whitespaces into several tags).
- R to be the set of *URLs* which have been clicked on by the search engine users.

In a logsonomy, we assume an association between t, u and r when a user u clicked on a resource r of a result set after having submitted a query t (eventually with other terms). The resulting relation $Y \subseteq U \times T \times R$ corresponds to the tag assignments in a folksonomy. One can call the resulting structure a *logsonomy*, since it resembles the formal model of a folksonomy as described above. Additionally, the process of creating a logsonomy shows similarities to the one of creating a folksonomy. The user describes an information need in terms of a query. He or she then restricts the result set of the search engine by clicking on those URLs whose snippets indicate that the website has some relation to the query. These querying and clicking combinations result in the logsonomy.

Major differences to a folksonomy

Logsonomies differ from folksonomies in some important points, which may affect the resulting structure of the graph. These points need to be taken into consideration when we discuss the structural and semantic properties of logsonomies in Section 6.4.

- Users tend to click on the top results of a list. In query log analysis, these clicks are usually discounted. To construct a logsonomy, this bias may be integrated by introducing weights for the hyperedges.
- While tagging a specific resource can be seen as an indicator for relevance, users may click on a resource to check if the result is important and then decide that it is not important. However, in our case, the act of clicking already indicates an association between query and resource.
- Users might click on a link of a query result list because they find the resource interesting even though it does not match the query.
- A user may click on a resource several times in response to the same query when repeating a search several times. This information is lost when constructing the logsonomy as TAS are not weighted.
- In logsonomies, a tag is created with a search click. Composed queries are thus another intentional creative process to describe the underlying resources.
- Queries are processed by search engines, leaving open to which extent the terms influence the search results.
- When a resource never comes up in a search, it cannot be tagged.
- Session IDs (in the MSN case) do not reflect the various interests of a typical user. They are probably more coherent as they contain the information needs of a restricted period of time.

3.3 Exploiting Tagging Data

The question about how tagging data can leverage information retrieval tasks has been studied by several authors. In this section we will summarize state-of-the-art studies considering bookmarking (Section 3.3.1) and tagging information (Section 3.3.2).

3.3.1 Exploiting Bookmarks

First experiments investigating bookmarks for information retrieval were conducted by Heymann et al. [2008]. The authors created a dataset of the social bookmarking system Delicious to run different analyses considering the system's tags and bookmarks. The authors found that the set of social bookmarks contains URLs which are often updated and also appear to be prominent in the result lists of search engines. A weak point is the fact that URLs produced by social bookmarking systems are unlikely to be numerous enough to impact the crawl ordering of a major search engine.

Kolay and Dasdan [2009b] analyse the suitability of bookmarks for web search. They used Delicious as well as randomly selected URLs to feed a crawler. The authors found that the average external outdegree of Delicious URLs close to the seed was more than three times larger that that for the neighbors of random URL seeds. Based on this finding, they conclude that Delicious URLs are a good source for discovering new content. Furthermore, the clickability rate of Delicious URLs is higher compared to a random selection of examples meaning that users tend to click on search results which also have been tagged in Delicious. This finding could be used for influencing the rank score of a page.
Morrison [2008] performed a user study to compare rankings from social bookmarking sites against rankings of search engines and subject directories. Participants had to rate results from both systems after having submitted a query. The authors found that search results of both systems are overlapping. Furthermore, hits appearing in both search lists have a higher probability of being relevant than those returned by only one of the two systems.

3.3.2 Exploiting Tags

There are several studies examining tags as a source of metadata to describe web resources. Most of them compare tagging data to web search queries, anchor text or the content of web pages. Such investigations improve our knowledge about whether social annotations in form of tags can be of help for improving search results and – the other way around – whether query log data may improve the recommendation of tags.
Noll and Meinel [2008b] explore the characteristics of tags added by "readers of web documents" by comparing them to the "hyperlink anchor text provided by authors of web documents and search queries of users trying to find web documents". They group their analysis according to five different aspects: length, novelty, diversity, similarity and classification. Adding the dimension of relevance, we use this scheme to categorize the different study results.

- *Length* The average length of all three metadata types lies between 2 and 3 terms. Noll and Meinel [2008b] guess that users seem to select "only 2 or 3 terms per action even across different problem domains (social bookmarking, hyperlink creation, searching the Web)".

- *Relevance* In general, tags are considered as relevant for capturing the intent and content of web documents [Heymann et al., 2008; Li et al., 2008]. Li et al. [2008] compare user-generated tags with web content. For instance, they compute the most important keywords of a web page using *tf-idf*-based weights and show that most of them have been used as a tag of the specific website. The authors conclude that "in general, user-generated tags are consistent with the web content they are attached to, while more concise and closer to the understanding and judgments of human users about the content. Thus, patterns of frequent co-occurrences of user tags can be used to characterize and capture topics of user interests".

- *Novelty* Comparing the overlap of the different kinds of metadata to the content of documents it can be shown that many of the tags used for annotating URLs can also be found in the document [Heymann et al., 2008; Noll and Meinel, 2008b] or in other metadata fields [Jeong, 2009]. According to Heymann et al. [2008], one in six

tags also appear in the title and one in two in the page's content. Additionally, tags are often mentioned in the URL's domain (for example "java" for "java.sun.com."). Noll and Meinel [2008b], calculating the overlap of social bookmarks, anchor texts or search queries state that "the majority of available metadata [...] add only a small amount of new information to web documents." Jeong [2009], after examining tags and other metadata fields of the video sharing platform YouTube points out that the overlap of tags and terms in other metadata fields such as title or description is very high.

- *Diversity* By measuring the entropy of tags and comparing it to the entropy of anchor terms and search queries per document it can be shown that tags are less diverse than search queries, but more diverse than anchor tags [Noll and Meinel, 2008b]. This can be explained by the fact that searchers formulate their information need before looking at documents while tags are created knowing the document's content. Nevertheless, tag noise exists and needs to be handled – for example by restricting the number of tags per document [Cattuto et al., 2008].

- *Similarity* Comparing tags, search terms and anchor tags to each other and the categories of the Open Directory Project (ODP) (mentioned in Section 2.1.1) using the cosine similarity, the results reveal that tags are more similar to the classification system than to search queries or anchor tags [Noll and Meinel, 2008b].

- *Distributions* A detailed comparison of query term and tag distributions is presented in Carman et al. [2009]. Similar to the findings of Noll and Meinel [2008b], comparing the plain vocabulary overlap, search terms seem to be more similar to page content than to tags. When considering frequency distributions, queries and tags resemble each other more than they resemble content terms. The authors suggest using tags for smoothing document content models and show in their first results that tagging data may be useful.

Besides the comparison of tags to the content of web documents and queries, the suitability of tags as a knowledge base for information retrieval tasks such as web search result disambiguation, classification, ranking or query expansion was analysed. A preliminary exploration of the suitability of tags for web search result disambiguation was conducted by Au Yeung et al. [2008]. Using four search terms as examples, they show how to identify different meanings of the terms using tags of a folksonomy and propose an algorithm to match tags to page content in order to find out the specific meaning of the page.
In Noll and Meinel [2008a], the authors matched tags added by users to a specific website to the categorization scheme created by the editors of the Open Directory Project. The higher the hierarchy level of a specific category, the more matches could be found with the tags. This was especially true for web sites with high popularity. The authors conclude that tags are better suited for broad classification purposes, i. e., the classification of the entry pages of websites. A more narrow categorization, considering pages at a deeper level, would probably be better handled by content analysis.
The authors of Zubiaga et al. [2009] did a similar study comparing not only tags, but different social annotations and their suitability for web page classification. Among the different annotations were tags, notes, highlights of content on a page, reviews and ratings. Their results show that classifying web sites with tags and comments performs better than

content-based classification. The combination of content-based approaches with social annotations yielded the best results, however.

Ranking functions can be enhanced by social information either by re-ranking the documents of a result list or by personalizing a result list. Li et al. [2012] use tagging information to re-rank documents. They assume that documents with high similarity score between document terms and tags should retrieve a similar retrieval score. After a preliminary ranking, they compute similarities between documents in the ranking list using matrix factorization methods and utilize the similarity degree to re-rank documents. The authors of Lee et al. [2012] propose the construction of a social inverted index taking not only the document and its terms but also the user tagging the document and its tags into account. Bouadjenek et al. [2013] propose a linear weighting function which integrates a vector representing the social representation (i. e., tags) of a document into the Vector Space Model. Additionally, they take care for a user's personal interests by computing the similarity between a user profile and the social document representation.

Finally, several authors explore the use of social annotations for query expansion [Biancalana et al., 2013; Guo et al., 2012; Lin et al., 2011; Zhou et al., 2012]. They enhance existing expansion techniques with tagging information. For example, the authors of Guo et al. [2012] extend the co-occurrence matrix to measure how often tags and query terms appear together. Overall, the different studies show that tags serve as a knowledge base for information retrieval tasks.

Chapter 4

Spam Detection

With the growing popularity of social tagging systems, not only honest users started to organize their bookmarks, but malicious ones began to misuse collaborative tagging for their own benefits. In the social bookmarking system BibSonomy, for example, 90% the bookmarks are spam-related. Wetzker et al. [2008] could also show that most of the highly active users in Delicious are spammers. While spam in Web 2.0 applications such as social bookmarking or social networks has been a research field only in recent years, spam detection techniques in applications such as e-mail or the Web have been discussed and refined for several years. Happily, many of the techniques developed can be adjusted and transferred to spam detection in social tagging systems (see Chapter 7 and Chapter 9 for the application of spam detection algorithms in this thesis).
This chapter briefly reviews the field of spam detection. It starts with characterizing the task of spam detection in general, including a definition of spam and the presentation of existing spam detection methods. Finally, the peculiarities of spam in the context of the Social Web will be discussed and social spam fighting approaches presented.

4.1 Definition

In today's digital world, it might seem unnecessary to characterize spam. Everybody has dealt with it – be it via e-mail, web pages or SMS. In the scope of the definition of a spam filter for the BibSonomy project we found, however, that not every annotator felt the same about what can be regarded as spam. Different cultures, educational backgrounds and attitudes lead to a different perception about the border cases of spam bookmarks. This is why we will briefly define spam in the first part of this chapter.
Jezek and Hynek [2007] described spam in the context of digital information dissemination, a form of *electronic publishing* which refers to the electronic distribution of e-books, websites, blogs or e-mail:

> If published and distributed properly, it "contributes to exchanging information on the Web, but used in a malicious way, it serves for broadcasting (mis)information to the general public." [Jezek and Hynek, 2007]

Commonly known under the term of spam, this form of unsolicited messaging absorbs much time of system developers, administrators and users.

Normally, it is not difficult to distinguish spam messages from legitimate ones. However, there are cases, where different classifiers (be it human or machines) would disagree. In order to establish a common understanding, Cormack [2008] identified four major characteristics. The authors had e-mail applications in mind. Nevertheless, they serve as a good characterization for all kinds of information dissemination including spam in social media applications.

- *Unwanted*: The majority of a system's users are not interested in the content presented by the spammer.
- *Indiscriminate*: The content is not aimed at reaching a specific target group (for example scientific users), but could be posted in any kind of application.
- *Disingenuous*: The postings in order to not be detected by spam filters need to be presented as legitimate and attractive as possible.
- *Payload bearing*: The payload refers to the message carried by the spam mail which relates to an eventual benefit for the spammer. Obvious messages may be product names, political slogans or adresses. Indirect messages can be a strange name or reference, which tempts the recipients to search in the internet and be forwarded to the spammer's website [Cormack, 2008].

4.2 General Spam Detection Approaches

Spam detection approaches can be split into two major groups: *heuristic techniques* and *machine learning techniques*. The first group includes techniques which rely on human input — be it by the direct classification of users, traffic analysis, the design of positive or negative indicators or the creation of rules for spam detection. The second group considers mostly supervised or unsupervised algorithms. In the supervised setting, different kinds of models are learned from a given set of training instances which consist of labeled examples. In the unsupervised setting, algorithms such as clustering approaches find a solution without depending on labeled instances. The next paragraphs will shortly introduce different techniques from the heuristic and machine learning fields which have been employed in the experiments discussed in Chapter 7 and in Chapter 9. Further surveys and reviews on spam detection can be found in Blanzieri and Bryl [2008]; Guzella and Caminhas [2009].

4.2.1 Heuristic Approaches

Black and whitelists

One of the earliest and simplest method to identify spam is the creation of black- and whitelists which – depending on the application – explicitly state good or bad e-mail addresses, users or IPs. The lists are either updated manually or fed by algorithms identifying black- or whitelist candidates.

Collaborative spam filtering

Collaborative spam filtering leverages feedback of all users working with the system [Gray and Haahr, 2004]. Many Web 2.0 applications ask their users to report it to the system provider when they find malicious content. This is often realised with the help of a button users click when they want to classify something as spam. System providers analyse the incoming user requests and decide how to proceed with the rated items (see for example [Han et al., 2006]).
Though collaborative spam filtering is an interesting opportunity to easily collect user feedback about possible spammers, it is difficult to rely on such filtering techniques. Often, users have a different perception of what is or is not spam. Furthermore, spammers might misuse the feedback mechanisms to declare legitimate entries as spam. Service providers need to consider these disadvantages when implementing such a filter.

Rule-based filtering

Rule-based filtering techniques are content-based methods which include the specification of rules. Such rules often refer to scanning a list of words (for example typical spam terms or IP addresses) or checking against regular expressions (for example filtering e-mail addresses from universities). For a new item multiple rules are checked to see whether they apply or not.
For instance, the SpamAssassin [1] application, an open source solution released under the Apache License 2.0, uses rules (among other techniques) for e-mail spam classification. Each rule has a positive (spam) or negative (non-spam) score assigned. For each message, the scores of each rule are summed-up. Based on a threshold value, it can be decided if the message should be classified as spam.
In general, the definition of rules is time consuming. Rules need to be defined carefully in order to achieve high accuracy. Also, as spammers adjust their methods to avoid being filtered, rules need to be updated regularly.

4.2.2 Machine Learning Approaches

In the area of spam filtering, most machine learning approaches consist of classification methods. Such techniques use labeled training examples to infer a model which can then be used to assign labels to unknown training instances.
More formally, the task of classification can be defined as follows: Given a labeled training set of n examples $\{\mathbf{x}_i, y_i\}$ $(i = 1, \ldots, n)$, where $\mathbf{x}_i \in \mathbb{R}^m$ with $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ denoting the feature vector with m features, n the size of the training data set, and $y_i \in \{-1, +1\}$, one seeks to find a function f (the "model" or "classifier") that minimizes the expected loss of the mapping $\mathbf{x}_i \rightarrow \{-1, +1\}$.

Feature Engineering

The generation of classifiers is based on training data which consists of labeled examples. The latter are normally represented by means of feature vectors. The generation of such features is a key factor for a spam filter's success. Depending on the data, some features

[1] `http://spamassassin.apache.org/`

are easy to extract, while others need to be carefully preprocessed (for example text or images). In the domains most similar to social spam applications (e-mail detection and web spam detection), these features are mostly generated from the message's or website's content, e. g., from text. Documents are represented as a *bag of words*, where each feature corresponds to a single term found in the document. Normally, terms are preprocessed by removing case information, punctuation, stop words and very infrequent words. Sometimes several morphological forms of a word are mapped to the same root term by applying a stemming algorithm which eliminates suffixes from words.
A text document d is then represented by a set of terms $T = \{t_1, t_2, \ldots, t_m\}$. The values of the features of $\mathbf{x}_i$ are given as a function which weighs the occurrence of t_i in d. Functions can either consider binary weights (a word is present or not) or they account for the term's occurrence in the text and / or in the entire corpus.
One of the most popular weighting measures is *tf-idf* [Salton and Buckley, 1988], which denotes a composition of raw frequency (*tf*) and inverse document frequency (*idf*). *tf* counts the number of words in the document with the intuition that the importance of each word is proportional to its occurrence in the document. *idf* considers the occurrence of a word in the entire document corpus. Words which rarely occur in the corpus but frequently appear in the specific document are more valuable for text classification than words occurring frequently in many documents. Thus, the importance of each word is inversely proportional to its occurrence in a document d. *tf-idf* can then be expressed as:

$$\text{tf-idf} = tf(d, w) \times \log(N/df(w)) \tag{4.1}$$

Summaries of further weighting functions considering term scores can be found in Blanzieri and Bryl [2008]; Guzella and Caminhas [2009].
A general problem of representing documents by their terms and training a classifier based on such documents is the difficulty to identify spammers using new terms which have not been part of the training corpus. The filter needs to be re-trained so that unknown words, misspellings or word variations are acknowledged by the classification model. One way to avoid this conflict is to integrate more features not directly based on text. Selecting such features ranges from taking into account the ip address of the message transmitter, using other black-or white-mail lists or Google's PageRank scores for websites to considering temporal aspects and clickdata. The features we computed for detecting spam in social bookmarking systems are described in Chapter 7.3.1.
Considering the wealth of features – especially in case of text features – it can be useful to preselect certain features before training the classifier. Some classifiers, however, such as support vector machines (see Section 4.2.2), have the ability to handle high dimensional input spaces as their classification decision does not depend on the number of features but on the margin which separates the data [Joachims, 1998a].

Algorithms

The selection of appropriate classification algorithms depends on the requirements the domain of spam filtering imposes. Such requirements include good classification performance, fast prediction of new entries, fast adaptations to the changes of spammers and robustness to high dimensionality, as feature vectors tend to have many dimensions, especially when using text features. In the context of online updateable algorithms, Sculley [2008] also mentions the requirement of scalable updates, where the costs of updating a

model should not depend on the amount of training data. The following paragraphs briefly present the most prominent classifiers used in spam detection tasks, which we also used for our spam experiments in Section 7.3 and 8.3.1.

Decision Tree Learning

Decision tree learning allows the classification of new examples by traversing a decision tree [Mitchell, 1997]:

- The *nodes* in the tree represent a test for a certain instance attribute.
- The classification process starts with the *root* node.
- Depending on the test outcome, a *branch* is selected to move down the tree to the next node.
- The resulting *leaf* node reflects the classification decision.

A prominent algorithm to construct decision trees is the ID3 algorithm and its variants [Quinlan, 1986]. The family of ID3 algorithms infers a decision tree by building them from the root downward, greedily selecting the next attribute for each new decision branch added to the tree. The decision about which attributes are best can be obtained by using different quality measures such as the information gain (see [Quinlan, 1986] for a definition).
One issue in decision tree learning is the problem of overfitting the data i. e., a model is learned which perfectly classifies the training examples but increases the test data error. Several strategies exist in order to prevent overfitting. For example, one can stop the tree construction before it reaches the point where it fits all the examples of the training data (pre-pruning). Post-pruning methods, in contrast, reduce the tree after it has been entirely built. Several improvements in respect to noise handling, missing features, better splitting criterias or better computing efficiency have been introduced to the ID3 algorithm since then. In 1993, Quinlan released the C4.5 algorithm, an extension of the ID3 algorithm, which also addresses these issues [Quinlan, 1993].

Naive Bayes

Naive Bayes classifiers were first proposed for e-mail spam filtering by Sahami et al. [1998]. Since then, they have been widely used. Their attractiveness stems from their efficiency (training time is linear to the number of training examples and storage time is linear to the number of features), simplicity and comparable performance to other classification algorithms.
The Bayes classifier calculates the probability that a given feature representation $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ belongs to a class y_k by applying the Bayes' theorem which states that the *posterior probability* $P(y_k \mid \mathbf{x})$ of a target value can be calculated from the *priori probability* $P(\mathbf{x})$ of a random feature vector represented by $\mathbf{x}$ together with $P(\mathbf{x} \mid y_k)$ and $P(y_k)$, i. e., the probabilities that a feature vector $\mathbf{x}$ is classified as y_k is represented by:

$$P(y_k \mid \mathbf{x}) = \frac{P(y_k) \times P(\mathbf{x} \mid y_k)}{P(\mathbf{x})} \tag{4.2}$$

Naive Bayes classifiers rely on the assumption that the components x_j, $j = 1, 2, \ldots, m$ are conditionally independent and can be written as:

$$P(\mathbf{x} \mid y_k) = \prod_{i=1}^{m} P(x_j \mid y_k) \tag{4.3}$$

Therefore, 4.2 can be transformed into:

$$P(y_k \mid \mathbf{x}) = \frac{P(y_k) \times \prod_{j=1}^{m} P(x_j \mid y_k)}{P(\mathbf{x})} \tag{4.4}$$

Bayes classifiers then assign the most probable target value given the feature values for a new instance.

$$y_{NB} = \underset{y_k \in Y}{\operatorname{argmax}} P(y_k \mid \mathbf{x}) \tag{4.5}$$

$P(x_j \mid y_k)$ can be computed by counting the frequency of various data combinations within the training examples [Mitchell, 1997].
One can distinguish two event models which include the naive Bayes assumption as described above [McCallum and Nigam, 1998]. In the *multi-variate Bernoulli* event model, a document is represented by a vector of binary attributes indicating that words occur or do not occur in the document. Probabilities for a document are computed by multiplying the probability of all word occurrences, including the probabilities of not-occurring terms. In the *multinomial* event model, a document is represented by the set of word occurrences from the document. Just like the multi-variate Bernoulli, the order of words gets lost. Nevertheless, the information about how many times a term occurred in the document is retained. Probabilities are then calculated by multiplying the probability of words which occur in the document. Different comparisons show that the multinomial model performs better in the case of large vocabulary sizes [McCallum and Nigam, 1998].

Logistic Regression

Logistic regression algorithms belong to the discriminative supervised machine learning methods.
The basic goal is to find the weights for the linear model which can then be used to classify a new example either as a positive or a negative instance. In contrast to other discriminative methods, the logistic regression classifier computes the *probability* of a specific feature vector $\mathbf{x}$ being part of the positive class $y_k = 1$. This is done by applying the logistic function which maps an input value in the range of ∞ to $-\infty$ to the output $[0, 1]$. The logistic function is defined as

$$\sigma(t) = \frac{1}{1 + e^{-(t)}} \tag{4.6}$$

Applying a linear function of $\mathbf{x}$ one can express t as

$$t = \mathbf{w^T x} + b. \tag{4.7}$$

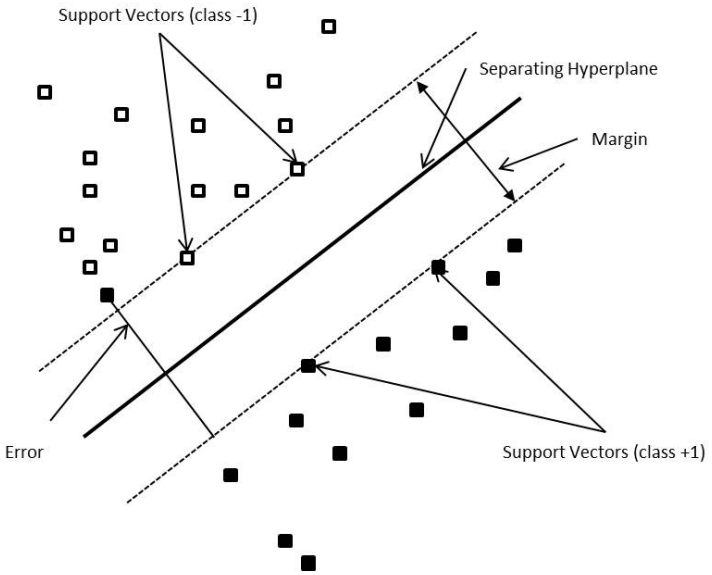


Figure 4.1: Hyperplane which separates positive and negative examples in a multidimensional space

The probability, that an input vector $\mathbf{x}$ belongs to class $y_k = 1$ is then

$$f(\mathbf{x}) = P(y_k = 1 \mid \mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{w^T x} + b)}} \tag{4.8}$$

The positive label can be predicted, if the probability exceeds a certain threshold τ, i. e., $f(\mathbf{x}) > \tau$. More information, especially how weights are computed in the software toolkit Weka (the tool used for the experiments in Section 7.3 and 8.3.1) can be found in le Cessie and van Houwelingen [1992].

Support Vector Machines

Support Vector Machines (SVMs) [Cortes and Vapnik, 1995] received great attention in the last years as they out-performed other learning algorithms with good generalization, a global solution, the number of training parameters and a solid theoretical background [Amayri and Bouguila, 2010]. An introduction to SVMs for ranking has been presented in Section 3.2.3.
A linear classifier in the form $f(\mathbf{x}) = \mathbf{w^T x} + b$ is employed to assign a class to a new example. The weight vector $\mathbf{w}$ is derived by finding a hyperplane which separates positive and negative examples with the maximum possible margin. This turns out to be a quadratic programming problem of the form

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & g(\mathbf{w},\xi) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & \forall\{(\mathbf{x_i}, y_i)\} \quad y_i(\mathbf{w^T x_i} + b) \geq 1 - \xi_i \\ & \forall i \quad \xi_i \geq 0 \end{aligned} \tag{4.9}$$

where n is the number of training examples and ξ_i is called a slack variable. Such variables allow for handling data which are not linearly separable by allowing a certain amount of

error. This error increases the further the misclassified point is from the margin's boundary. C controls the trade-off between minimizing the errors made on the training data and minimizing the term $\frac{1}{2}||\mathbf{w}||^2$, which corresponds to maximizing the margin's size. Figure 4.1 shows an example of a hyperplane separating positive and negative examples.
In Joachims [1998b], SVMs have been shown to work well to classify text examples. Comparing them to other classifiers solving the task of spam detection they demonstrated high accuracy rates [Drucker et al., 1999]. Results can even be improved using string kernels and different distance-based kernels than the classical ones [Amayri and Bouguila, 2010]. Another line of research around SVMs is their adjustment to the online setting of spam filtering, where classifiers need to be updated continuously [Sculley and Wachman, 2007].

4.2.3 Evaluation Measures for Spam Filtering

Several evaluation measures have been used to assess spam filtering applications. Sculley [2008] describes the requirements for such a measure as follows:

> Clearly, we would like to maximize the number of True Positives (TPs), which are actual spam messages correctly predicted to be spam, and the number of True Negatives (TNs), which are actual ham messages correctly predicted as such. Furthermore, we would like to minimize the number of False Positives (FPs), which are good ham messages wrongly predicted to be spam, and to minimize the number of False Negatives (FNs), which are spam messages predicted to be ham.

Precision and recall are two simple and well-known measures for evaluating classification algorithms.
Precision denotes the fraction of positive examples correctly classified among all examples, which have been classified as positive:

$$Precision = \frac{TP}{TP + FP} \tag{4.10}$$

Recall is the fraction of positive examples correctly classified among all possible positive examples, i. e., true positives and false negatives:

$$Recall = \frac{TP}{TP + FN} \tag{4.11}$$

The F_1-measure combines precision and recall using the harmonic mean, which better reflects the understanding of "average" in respect to ratios than the arithmetic mean.

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4.12}$$

By introducing the parameter β, the balance between precision and recall can be controlled. With $\beta = 1$ the F-measure becomes the harmonic mean, if $\beta > 1$, it puts more weight on recall, if $\beta < 1$ it is more precision-oriented (F_0 = Precision).

$$F_\beta = (1 + \beta^2)\frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{4.13}$$

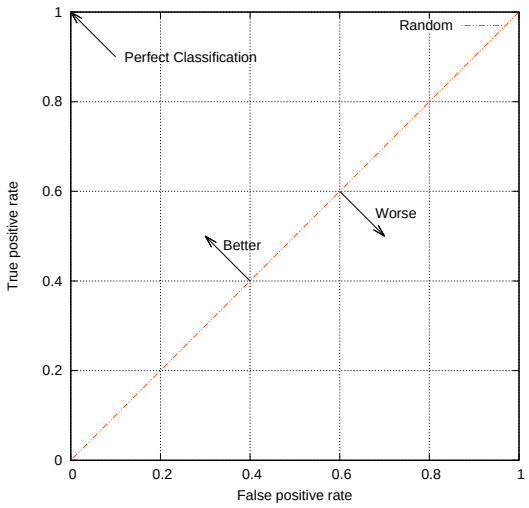


Figure 4.2: The ROC space and its interpretation. The ROC curve of a classifier randomly guessing lies somewhere along the line from (0,0) to (1,1). All values above reflect a classifier better than random.

Accuracy denotes the fraction of true classification results (either correctly classified positives or correctly classified negatives) among all possible examples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.14}$$

Precision, *recall*, F_β and *accuracy* optimize some of the requirements just described. However, it is difficult to apply these measures to highly skewed classes, i. e., datasets where one or very few classes dominate the other classes significantly in terms of the amount of training (and test) examples. In the spam domain, many applications have a lot more spam elements than they have legitimate elements.
The problem with the "traditional" measures is that they depend on both portions, i. e., on the positive and negative class fractions. For example *precision* calculates the number of true positives in reference to the number of all positives identified including false positives which contain examples of the negative class.
The AUC value, in contrast, is based on a composition of two evaluation measures which consider either only the positive or the negative class, namely the TP rate and the FP rate. The TP rate corresponds to the recall defined above (see Equation 4.11):

$$TP\ rate = \frac{TP}{TP + FN} \tag{4.15}$$

The FP rate is defined as

$$FP\ rate = \frac{FP}{FP + TN} \tag{4.16}$$

A so-called ROC (receiver operator characteristic) curve depicts the TP rate on the y-axis and the FP rate on the x-axis. Many classifiers (for example the ones described above) not only predict a new instance's class, but yield a probability value showing how confident they are about their decision. One can order these instances considering such confidence scores. By traversing the list of ordered instances, for each instance one can produce a

point in the ROC space by computing the TP rate and the FP rate at that position. The points can then be connected and form the so-called ROC curve. The area under such a curve is the AUC value which is also called the ROCA (ROC Area).
As each axis is scaled in the range of $[0, 1]$ forming a unit square, the AUC value is also in the range of $[0, 1]$, where a higher AUC value reflects a better classifier. AUC values can be interpreted as a probability indicating how probable it is that "a classifier will rank a randomly collected positive instance higher than a randomly collected negative instance." [Fawcett, 2004]
Figure 4.2 illustrates this interpretation. The line from (0,0) to (1,1) represents a random classifier. All values above reflect a classification better than random, all values below are worse than random. Classifiers with prediction values below 0.5 should convert their classification and predict the negative class. The steeper the ROC curve is at the beginning, the bigger the area under the ROC curve is leading to better AUC values. Therefore, the classifiers whose lists reflect a better ordering of positive and negative instances get a better grade.

4.3 Characterizing Social Spam

At first glance, spam in social media applications does not differ significantly from spam in the rest of the digital world. The motivation for distributing spam in social media applications is the same and the content (text, graphics) is often similar. Many of the classification methods applied in other domains can therefore be used to detect spammers in the Web 2.0.
The difference with other spam areas lies in the way of distribution, the shortness of spam messages these systems allow and the additional features social media applications offer to detect spam. In the following, we will review the specialities of social spam in social media applications and present state-of-the-art approaches to detect spamming attempts in different media applications.

4.3.1 Spam in Social Bookmarking Systems

Spam has become a major problem for tagging systems. As technical barriers are low and the possibility to reach a big audience are high, a high percentage of new posts in social tagging systems consist of spam. Depending on the system these can be bookmarks (social bookmarking systems), videos (video sharing sites) or photos. In this section, we focus on spam in social bookmarking systems as our spam detection experiments in Chapter 7 are conducted on data of a social bookmarking system and most related, relevant projects considering spam detection in social tagging systems have been introduced based on a social bookmarking system. An overview of related work is also presented in Ivanov et al. [2012].

Spamming incentives

The major motivation for spamming is financial [Markines et al., 2009a]. Spammers want to attract as many users as possible to a specific website, photo or video promoting mainly commercial, but also political or religious content. Since most internet users employ search engines to find information on the Web, spammers aim at getting high rankings

for specific query terms in search engines [Chen et al., 2009a]. An important factor search engines consider in their ranking is the link structure of the specific website, e. g., how well a specific site is connected. The more websites link to it, the more important it is. This principle has been explored in the PageRank algorithm [Page et al., 1999] (see Section 2.5.1), which is employed by search engines such as Google. Social tagging systems offer a cheap and easy way to create such links. As tagging systems are crawled by search engines, spammers post links at marginal cost hoping to gain a better ranking in search engines [Krause et al., 2008c].
By posting spam content, not only search engines are lead to web spam sites, but also other participants of the social tagging system. Users often browse and search the system's content by clicking on tags they added to their own posts, or tags of the tag cloud or by simply following other users in the system. When spammers use popular or very general tags, their chance to lead many visitors to their sites therefore increases.

Consequences for social media applications

Besides allocating network and storage resources which could be used for legitimate purposes, spam destroys the system's usability and quality. Social web applications make their living from the interaction and participation of their users. If those participants are distracted by spamming activities, they might stop using the system. Also, the speciality of tagging systems – their inherent semantics, which can be leveraged for all kinds of applications – can be destroyed if no spam filtering activities are employed.

Spam Filtering Methods

Heymann et al. [2007] categorized existing methods for fighting spam into three main groups: prevention-based, demotion-based and detection-based.

Prevention-based Methods Prevention-based methods make the posting of spam entries on a system more difficult. Such methods are mainly small features implemented into the system. Both legitimate users and spammers are confronted by these features when interacting with the system. The objective is to exclude spam robots. The difficulty is, however, to not make it too difficult for loyal, legal users who might otherwise stop using the system. Prominent examples for prevention-based methods are:

- CAPTCHAs
- Hiding / personalization of interfaces

Heymann et al. [2007] mentioned further financial barriers such as the introduction of account fees or the obligation to pay per action. As most social media systems offer free services to enable as many users as possible to participate, theses measures are also counterproductive for legitimate users.

Demotion-based Methods Demotion-based methods refer to the devaluation of spam entries compared to entries of legitimate users. For example, the order of ranking results according to a specific query can be influenced by degrading those results which are possible spam entries.

Detection-based Methods Detection-based methods focus on identifying spam and – depending on the system – deleting, hiding or marking it as spam. The core of those systems is a classifier which estimates whether a post entry or user is spam or not with the help of available information. The range of possibilities to identify spam is huge - from manual approaches to the implementation of machine learning techniques. The approaches relevant for spam detection in this thesis have been introduced in Section 4.2.

Spam classification methods

One of the first authors mentioning the problem of spam in social tagging systems were Cattuto et al. [2007] when they detected anomalies in their study of network properties in folksonomies. Heymann et al. [2007]; Koutrika et al. [2007] were the first to deal with spam in tagging systems explicitly. The authors identified anti-spam strategies for tagging systems and constructed and evaluated models for different tagging behaviour. In contrast to their approach using an artificial dataset, this thesis presents a concrete study using machine learning techniques to combat spam on a real-world dataset (see Chapter 7.3 and [Krause et al., 2008c]). After the publication of a BibSonomy dataset in the scope of the ECML/PKDD discovery challenge 2008 [Hotho et al., 2008] further studies were published. The experiments of participants of the ECML/PKDD discovery challenge 2008 are introduced in Section 7.4.4. Follow-up publications using the dataset include Bogers and Van den Bosch [2009]; Neubauer and Obermayer [2009]; Sung et al. [2010]; Yang and Lee [2011a]. The results, however, are still difficult to compare, as the pre-processing of the dataset is different. For example, Sung et al. [2010] reduce the tag size from about 400000 tags to about 20000 tags by excluding terms that appeared only once in a post or removing "noisy" tags such as numbers or tags composed of only two letters.
Several publications deal with the exploration of appropriate features to describe social bookmarking system users and therefore better distinguish between a spammer and a non-legitimate user. Markines et al. [2009a] construct six features which address tag-, content- and user-based properties. They evaluate their approach on a modified dataset of the SpamData08 dataset, changing the class distribution so that spam and non-spam class proportions are less skewed. Yazdani et al. [2012b] introduce 16 features based on tag popularity and user activities. Their most prominent feature computes the probability that the tags applied by a user are only used by legitimate users. The authors of M.Gargari and Oguducu [2012] introduce a new set of features which consider different temporal aspects. For instance, they observed that spammers show a different posting behaviour than legitimate users. For instance, they register several user accounts in a short period of time and post the same resource under different accounts. Such *short range resource bombardments* can be found by analysing the timestamps of each post. Users are classified as spam, when they exhibit high bombardment activities. Using the SpamData08 dataset they can show, that their method outperforms previous approaches, especially in terms of reducing the false positive rate.
While most spam detection approaches operate on a user level, i. e., users are classified based on their posts, Liu et al. [2009]; Sung et al. [2010]; Yang and Lee [2011a] propose methods to classify spam on a post level. In Liu et al. [2009], for each post, an information value is computed which is the average of each tags information value. A tag's information value is the proportion of the frequency, the tag is assigned to a resource by different

users, to the sum of the frequencies of all tags assigned by different users to this resource. The authors of Sung et al. [2010] compute scores for the tags of a post. Based on these scores, a post is considered spam or not. The scores are derived from tag usage and co-occurrence information of tags applied in posts of spammers and legitimate users. The authors experiment with different combinations of tag scores. For example, they identify so called "white tag" scores. Such tags are frequently used by non-spammers. Spammers pick them up to make their posts appear as legitimate posts. Yang and Lee [2011a] measure the semantic similarity between a tag and a web page. They related the keywords of the web site with the tags of a post using self-organizing maps.
Some works do not focus on developing spam classification methods, but find other effective approaches or present interesting insights when investigating data from social tagging system for other tasks. In Sakakura et al. [2012] the authors use a supervised learning scenario, but cluster users based on their sharing of bookmarks. As in M.Gargari and Oguducu [2012] the authors claim to better detect users, registering several accounts and bookmarking the same resource. Noll et al. [2009] explore the identification of experts in social tagging systems. They show that their algorithm is more resistant to spammers than more traditional methods such as the HITS algorithm [Kleinberg, 1999a]. When analysing tagging behaviour in social bookmarking systems, Dellschaft and Staab [2010] show that spamming behaviour deviate significantly from the behaviour of legitimate users. In Navarro Bullock et al. [2011b] we analyse different spam features with respect to their degree of privacy conformance and accuracy (see Section 8.3).

4.3.2 Spam Detection in other Social Media Applications

Microblogs

One of the most popular micro-blogging services in the world is Twitter [2]. Founded in 2006, it enables users to publish short messages of up to 140 characters, called tweets. The messages are read by followers or retrieved through search systems. With its popularity, Twitter is one of the most useful systems to receive real-time information about current events, opinions or news. Spammers take advantage of Twitter in many different ways. The authors of Thomas et al. [2011] performed an analysis of typical spam activities in Twitter on a dataset containing 1.8 billion tweets whereby 80 million were published by spammers. Using one or more accounts, spammers publish malicious links (for example, links to websites containing malware) or hijack popular topics. The authors also mention the growing market for (illegal) spammer operated software selling Twitter accounts, or URL-shorteners to disguise spam. Another study [Almaatouq et al., 2014] distinguishes two major classes of spamming activities: The first, mainly contains fraudulent accounts, which rather follow other users than beiing followed. The second class of spammers shows more similarities to legitimate users. The study's author assume that such accounts may be compromised. The hijacking of user accounts is also analysed in Thomas et al. [2014]. Based on a dataset containing 13 million of hijacked accounts, they study the likeness to become comprised, the dominant way of hyjacking accounts and the social consequences of users which have been compromised.
One of the major challenges to reduce micro-blogging spam activities is to capture the specific spam behaviour so that spam classifiers can work properly. Several studies analyse

[2] http://www.twitter.com

the performance of spam detection methods. Kwak et al. [2010] found that a simple filter based on excluding users who have used Twitter less than a day or tweets which contain three or more popular topics helps to distinguish between spammers and non-spammers. In Benevenuto et al. [2010] the authors used tweets considering three popular topics from 2009. They manually identified spam and non-spam users and extracted various features, among them content attributes such as the number of numeric characters appearing in a tweet, the number of popular spam words or the fraction of tweets which are reply messages. Additionally, they considered user behaviour attributes such as the number of followers per number of followees or the age of the user account. Similarly, Wang [2010] present an automatic spam detection approach using graph-based and content-based characteristics of twitter spammers. Ghosh et al. [2011a] study the dynamics of re-tweeting activities. They establish two features based on the time intervals between successive tweets and the number of times a user retweets a certain message. Other studies investigate link farming and how to prevent such farms [Ghosh et al., 2012] or look at correlations of URL redirect chains [Lee and Kim, 2013].

Blogs

Blog spam (also known as splogs) can be seen as a type of web spam where the author of the blog is a spammer. Such splogs are often created to attract traffic from search engines [Zhu et al., 2011].
First spam detection approaches in this field include Mishne et al. [2005], who identified comment spam in blogs by applying a language model and Kolari et al. [2006a], who derive textual features using the blog text, anchors and URLs and use a SVM to evaluate them. They extend their work showing that such features are more successful than link-based features such as incoming or outgoing links to classify spam blogs [Kolari et al., 2006b]. More features were suggested by Lin et al. [2007], including temporal, content, and link self-similarity properties. Yoshinaka et al. [2010] present a personalized blog filter helping to handle instances of the "gray" zone where system participants differ on their evaluation. Zhu et al. [2011] monitor online search results and filter blogs from those results by analyzing their temporal behaviour.

Wikipedia

Wikipedia is a popular website to publish all kinds of information. It can be used by any participant and has no entry barrier. Spamming attacks (often called vandalism in Wikipedia) can be defined as "any addition, removal, or change of content in a deliberate attempt to compromise the integrity of Wikipedia." [Javanmardi et al., 2011] According to West et al. [2011], "common forms of attack on Wikipedia include the insertion of obscenities and the deletion of content".
The authors of Geiger and Halfaker [2013] describe Wikipedia's vandalism detection network as a multi-layered system. Most malicious edits are removed by autonomous bots. More subtle vandalism is handled by humans (partly tool-assisted). Finally, specific scripts play a role in the combat against spam. Most research projects focus on improving bots by applying standard machine learning approaches [see for example Potthast et al., 2008; Smets et al., 2008]. In the scope of the PAN 2010 vandalism detection competition [Potthast et al., 2010] several authors proposed automatic detection mechanisms. A further

study outperforms the winning team on the same dataset by integrating a variety of features which they categorize as either metadata, text, language characteristics or reputation [Adler et al., 2011].
Among the malicious edits are link spam edits of the same kinds as introduced in the context of social bookmarking spam (see Section 4.3.1). Such links are often removed by Wikipedia's existing spam detection mechanisms. West et al. [2011], however, present a spam model to detect spam links more quickly by concentrating on popular articles, an eye-catching presentation of links, automated mechanisms of entry generation and the distribution of hosts.

Chapter 5

Data Privacy

In order to better understand data privacy concepts and the legal situation in Europe, especially in Germany, this chapter briefly reviews the basic privacy terminology, privacy principles and the European and German legislation. Section 5.3 then presents current research around data privacy in the Social Web. In Chapter 8 we will build on this foundation to discuss data privacy on social bookmarking systems. In the light of current events such as the extensive surveillance of citizens by the National Security Agency (NSA) [Marcel Rosenbach, 2014], data privacy in the digital world has come to the fore in media, politics and legislation. Current events in context of our research will therefore be briefly discussed in the conclusion in Section 11.3.

5.1 Basic Concepts

The term *privacy* generally refers to the protection of personal data (in Germany the so-called "Datenschutz"). More precise notions are *data privacy* or *information privacy*, though the two composites do not adequately capture the subject of privacy. According to Roßnagel [2007] the data itself does not need to be protected, but rather the person the data relates to. This concept of privacy has been clarified in the right to informational self-determination which will be briefly discussed in the following section. Afterwards, basic terms of privacy laws will be defined and principal guidelines how to implement privacy (laws) presented.

5.1.1 Privacy as the Right to Informational Self-Determination

Westin [1970] defined privacy in his fundamental work as the

> claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.

This concept of privacy was taken up by the German Federal Constitutional Court. In 1984, the Court formulated the right to information self-determination as a fundamental, personal right when canceling the Population Census Act, which was set up by the German government a year before. The so-called Census Decision ("Volkszählungsurteil"),

with the introduction of the basic right of informational self-determination, can be considered as a fundamental decision in the history of German data protection [Hornung and Schnabel, 2009]. Briefly, the right to informational self-determination is an individuals right to "determine what personal data is disclosed, to whom, and for what purposes it is used" [Fischer-Hübner et al., 2011]. As soon as a third party – be it a public authority or a private company – wants to process an individuals personal data without being authorized by this person or by exceptional cases explicitly stated in the law, the right to informational self-determination is violated.

5.1.2 Guidelines and Definitions

Guidelines by different organizations have been published with the goal of encouraging their member states to implement privacy laws confirming to specific principles. Among them are the OECD's Guidelines on the Protection of Privacy and Transborder Flows of Personal Data from 1980 [OECD, 1980] or the UN guidelines Concerning Computerized Personal Data Files from 1990 [Assembly, 1990]. In the European Union, the Data Protection Directive [Directive, 1995] (officially called the Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data) is the foundation for regulations with respect to data protection. As the Directive is a framework law, it needs to be implemented in the EU Member States by means of national legislations. In Germany, the (main) implementation is the German Federal Data Protection Act, which is further discussed in Section 5.2.
In the following section, we will introduce basic privacy terms in order to provide a common understanding. Thereby, we will follow the definitions of the Data Protection Directive.

Personal Data In general, the goal of the Data Protection Directive is to protect the right to privacy of natural persons relating in particular to the processing of their personal data. *Personal data* comprehends "any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity (95/46/EC Article 2 a)" [Navarro Bullock et al., 2011b].

Processing *Processing* personal data is further explained by Article 2 of the EU Data Protection Directive: Processing refers to "any operation or set of operations which is performed upon personal data, whether or not by automatic means, such as collection, recording, organization, storage, adaptation, or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, blocking, erasure or destruction".

Controller The Directive differentiates between the data controller and the data processor. The data *controller* is "the natural or legal person, public authority, agency or any other body which alone or jointly with others determines the purposes and means of the processing of personal data" (95/46/EC Article 2 d).

Processor The data *processor* relates to "a natural or legal person, public authority, agency or any other body which processes personal data on behalf of the controller".

While the controlling entity determines the purpose of data processing, the real processing can be done by another entity (95/46/EC Article 2 e). However, the processor acts on the behalf of the controller and needs to "implement appropriate technical and organizational measures to protect the data against loss, destruction, unauthorized disclosure or access, and other unlawful processing" [Noorda et al., 2010].

5.1.3 Privacy Principles

Privacy principles have been published by several organizations to trigger the discussion about privacy and privacy requirements. In many countries, including Germany, they serve as normative instructions which are implemented in different laws to preserve data privacy. The list below summarizes the essential principles. Similar summaries can be found in [Fischer-Hübner, 2001; Gutwirth et al., 2009; Kosta et al., 2011].

Principle of fair and lawful processing The data must be processed in a transparent way for the data subject and only as allowed by law or by consent of the concerned. This principle can be seen as a primary requirement which generates the other principles of data protection laws.

Principle of purpose specification and purpose binding Personal data should only be collected and used for the purposes specified. Such purposes can either be legally provided or agreed upon in advance by the data controller and the considered person. The permission for personal data processing only applies for the specified purpose. When the purpose is changed, the legal requirements must be checked again [Navarro Bullock et al., 2011b].

Principle of consent The processing of data is only permitted if the person concerned or a legislative authority approves its processing. The permission only holds for a specific purpose and set of data.

Principle of transparency Individuals should be provided with detailed information about what personal data is being collected, by which means it is collected, how it is processed, who the recipients are or how the data collector assures the confidentiality and quality of the data collected. The goal of providing transparency can be realised by information and notification rights and the possibility to change, delete or block one's data in case it is incorrect or illegally stored [Fischer-Hübner, 2001].

Principle of data minimization "As little personal data as possible are to be collected and processed" and have to be deleted or anonymized at the earliest [Navarro Bullock et al., 2011b].

Principle of information quality and information security Providing information quality refers to keeping personal data correct, relevant and up to date. The data processor needs to implement security mechanisms which assure the confidentiality, availability and integrity of personal data [Fischer-Hübner, 2001].

5.2 Legal Situation in Germany

Concrete requirements about how to deal with personal data can be found in the German Federal Data Protection Act (Bundesdatenschutzgesetz, BDSG) and in further laws handling specific data privacy regulations. Regulations for online services are specified in the German Telemedia Law (Telemediengesetz, TMG). Therefore, when discussing privacy regulations for online services such as Web 2.0 applications, both acts need to be considered. Depending on the service provider's company seat, further State (Landes-) laws may apply. For example, a service provider in Hesse needs to conform to Hessian state law as well.
In 2001, a major adaption of the BDSG was made in order to conform to European standards. In 2009, three further amendments were introduced. The following subsection will concentrate on the basic concepts of the German Federal Protection Act necessary in order to understand the analysis of Chapter 8 and 10.

5.2.1 German Federal Protection Act

The BDSG defines *personal data* as "any information concerning the personal or material circumstances of an identified or identifiable natural person ('data subject')"(§ 3 Abs. 1 BDSG). Such data includes information directly referring to a natural person such as the name, but also information which can be linked to factual circumstances relating to the person such as a comment or a photograph [Commission et al., 2010]. In order to decide whether data should be classified as personal, one needs to determine, whether it is possible to identify a human being from the data or from the data in conjunction with other information the data controller might be aware of.
Personal data can be divided into three categories: Content data, inventory and usage data ("Inhaltsdaten, Bestands- und Nutzungsdaten"). While the content data is regulated in the BDSG, inventory and usage data are considered in the TMG (German Federal Telemedia Act), which will briefly be explained in the next paragraph.
Similar to the European Data Directive, the BDSG differentiates between the collection, processing and use of personal data ("Erhebung, Verarbeitung und Nutzung", § 11 Abs. 1 BDSG). Processing comprehends activities such as recording, altering, disclosing, blocking and erasing of data (§ 3 Abs. 4 BDSG). The legitimacy to deal with personal data needs to be checked for each of those activities.
The BDSG applies to the public and the private sector. Public authorities comprehend institutions responsible for federal issues. Authorities which take care of federal state administration need to consider individual federal state (Länder) data protection legislation [Fischer-Hübner, 2001]. Public and private sectors are not treated the same. Some of the rules in the BDSG apply only to the public or private sector, while some have different requirements for both of them [Commission et al., 2010]. Additionally, special exceptions can be found. For example, in § 40, BDSG, special rules for processing personal data for scientific purposes are defined.
The BDSG and further specific laws define basic principles which conform to the ones discussed in the previous Section (5.1.3). In general, the collection, processing and use of personal data is prohibited. The only exceptions are an explicit legal permission or if the data subject has agreed to the processing of his or her personal data. The data must be obtained directly from the data subject rather than from third parties ("Direkterhebung", §

4 Abs. 2 BDSG). Only if a legal provision exists or the collection of data from the data subject involves a "disproportionate effort" can the data be obtained from other sources.
One of the core concepts is the *principle of purpose-limitation* ("Zweckbindung", § 14 BDSG) which implies that each handling of data needs a specific purpose which – at the best – has been documented in advance. Furthermore, the *principle of data avoidance and data minimisation* of § 3a has been introduced as an amendment in 2000. It refers to "the requirement that in any context, no more than the minimum amount of personal data may be collected; and that, whenever possible, personal data must be anonymized or pseudonymized"[Commission et al., 2010].
Anonymization and Pseudomization is defined in § 3 Abs. 6 BDSG. Data can be seen as anonymized when they can no longer, or only with a disproportionate effort with respect to time, costs and working forces be linked to a data subject. Pseudomization refers to the replacement of names or other identifiers by some kind of other attribute which prevents or significantly complicates the identification of the data owner.
The *principle of transparency* refers to an individuals right to know who collects the data, as well as where, when and for what purpose (§ 4 Abs. 3 BDSG). Transparency requires comprehensive information and access rights. Finally, the *principle of necessity* only allows the processing of personal data if it is absolutely necessary and no other (fair) means exist to achieve the intended purpose.

5.2.2 German Federal Telemedia Act

The Telemedia Act applies to all service providers which "operate under German law and offer services via the Internet, but exclude telecommunication or related services" [Sideridis and Patrikakis, 2010] (i. e., services which consist of signal distribution via telecommunications networks and broadcasting).
The main concepts in the Telemedia Act are:

Inventory data is regulated in § 14, TMG:

> The service provider may collect and use the personal data of a recipient of a service only if it is needed to establish, carry out or amend a contractual relationship between the service provider and the recipient on the use of telemedia (inventory data).

Such data can be the name, address, phone number, user name, password, e-mail, birthday or credit card number.

Usage data is regulated in § 15, TMG:

> The service provider may collect and use the personal data of a recipient of a service only to the extent necessary to enable and invoice the use of telemedia (data on usage). Data on usage are in particular
>
> 1. characteristics to identify the recipient of the service
> 2. details concerning the beginning, end and scope of the respective usage
> 3. details of the telemedia used by the recipient of the service.

Similar to the BDSG, the TMG requires a service provider not to process any personal information beyond what is necessary for service delivery unless it is explicitly allowed by the TMG or another provision considering telemedia services or a user has authorized the processing (§ 12 Abs. 1 TMG). The authorisation is bound to a specific purpose. A change of purpose needs to be legitimized by law or again by the user's consent. Data retention is therefore not legitimate. A user's approval can be asked for electronically when the approval process meets the following conditions (§ 13 Abs. 2 TMG):

1. the user has consciously and unambiguously given his authorisation
2. a record of the approval is kept
3. the recipient of the service can access the content of the approval at any time
4. the recipient of the service can revoke the approval at any time with effect for the future.

The purpose-binding and data minimisation principles have to be applied to inventory and usage data as well. This means that – though it is very easy for service providers to collect such data through forms and log files – as few personal inventory and usage data as possible should be collected and stored. Furthermore, after being used they need to be deleted or anonymized. In practice, when developing information systems, such legal limitations should be considered in advance in order to avoid expensive amendments. A provider should select the technical way of using as little data as possible or they need to inform their users in detail about the collection and processing of their personal information. Users can then decide in a self-determined way what kind of data processing they are comfortable with.

5.3 Data Privacy in the Social Web

With the growth of Web 2.0 systems, especially social networking services, the violation of privacy has come to the attention of researchers. In contrast to former privacy issues, where people refused to share sensitive data with a company or the government, many Internet user publish their private information voluntarily. Most of them, however, are aware of the privacy risk, but do not apply this concern to their own usage. In literature, this has been termed as the *privacy paradox* [Barnes, 2006]. Several works explore the motivation and extent of information disclosure and discuss risks with respect to this revelation [see for example Gross and Acquisti, 2005; Krishnamurthy and Wills, 2008; Krishnamurthy et al., 2008; Schrammel et al., 2009a,b; Stutzman, 2006].
In general, sensitive data is shared widely without limiting standard private settings (which are often tailored to publishing data). For example, Gross and Acquisti [2005] analysed the information revealed by 4000 Carnegie Mellon University students on Facebook and found that 61 percent of the profile pictures disclosed could be used for identifying a user. In 21 percent of the cases, the information available (including published phone numbers and addresses) would make it possible to stalk someone. A few years later, Farahbakhsh et al. [2013] analysed 479K Facebook profiles in terms of the kinds of profile attributes users tend to publish. On average, users make about 4 attributes publicly available, whereby the friend-list is the most often published attribute. Yang et al. [2012] show that web

users can be identified from only small pieces of information published online by using state-of-the-art search techniques and online repositories.
Schrammel et al. [2009b] explored the information disclosure behaviour of different online communities by analyzing demographic, context and usage variables and their correlation to the willingness of disclosing information. They assume that the "actual usage purpose and goal of a user when interacting with a community is the main driving factor behind the information disclosure behaviour." Taddicken [2014] found that " the higher the people rate the social relevance of the Social Web as important, and the more they focus on the use of specific Social Web applications, the more information they disclose. Social Web users tend to self-disclose more personal and sensitive information when their friends and acquaintances also use it."
Gürses and Berendt [2010] argue that viewing data privacy solely as a concept of confidentiality is not enough. The authors highlight different aspects of privacy including the construction of identities as a result of constant negotiations what data to disclose or hide. Similar to us (see Section 8) they argue for a combination of legal and technical measures to preserve privacy.
Different approaches considering data privacy issues in online social networks have been proposed. These include *privacy enhancing technologies* such as anonymization techniques [Bhagat et al., 2009; Zhou et al., 2008a] and *privacy aware engineering* such as the design of privacy policies (for example [Danezis, 2009; Fang and LeFevre, 2010]).
A first analysis of privacy in social bookmarking systems considering German law has been conducted by Lerch et al. [2010] and Krause et al. [2010]. A detailed legal analysis considering the legal situation of private data handling in social bookmarking systems is given in Doerfel et al. [2013]. This includes considerations about social peer reviews, spam detection and the responsibilities of service providers. Eecke and Truyens [2010] analyse legal issues raised by the application of the EU Data Protection Directive to social networks. Hoeren [2010]; Stadler [2005] deal with problems in respect to the processing of usage data in order to fight misuse.
The analysis of privacy in social bookmarking systems [Krause et al., 2010; Lerch et al., 2010] will be presented as part of this thesis in Chapter 10. This includes an interdisciplinary analysis of what kind of user data from a social network is necessary to efficiently conduct data mining and what kind of data protects the privacy rights of users. The methodology and experiments will be presented in Chapter 8.

Part II

Methods

Chapter 6

Social Information Retrieval in Folksonomies and Search Engines

6.1 Introduction

In Chapter 3 we have characterized social information retrieval in general. One, social approach of retrieving digital information is the usage of social bookmarking systems. Over the last years, a significant number of resources has been collected in these systems, offering a new form of searching and exploring the Web [Krause et al., 2008a]. To many folksonomy users, this personalized, community driven search has become an alternative to search engine information retrieval.

The major differences between a folksonomy and a search engine concern the interface and content-creation aspects. Folksonomies allow users to organize and share web content. By contrast, classical search engines index the Web and offer a straightforward user interface to retrieve information from this index [Benz et al., 2009b]. The index itself is created by automatically crawling the Web, while the content of a folksonomy emerges from the explicit tagging by its users. As a consequence, users, not an algorithm, decide about relevance in a folksonomy. The perception of users can be integrated into search engine rankings as well. One prominent example is the integration of user feedback as extracted from log files of users' click history in order to improve rankings (see Section 3.2.2).

In this chapter, we search for similarities and differences between information retrieval on the Web and in social bookmarking systems. We hereby consider four different aspects of social search as identified in the introduction in Section 1.2.1:

Usage behaviour and system content: We will conduct an analysis of the systems as they are, taking a look at user interactions and content in folksonomies and search engines. We will show that, indeed, search engines and folksonomies are used in a similar way and also cover similar content.

Structure: We will construct a folksonomy like structure (a *logsonomy*, see definition in Section 3.2.4) out of search engine clickdata and compare the topological and semantic properties to the ones of the well-known folksonomy Delicious. By looking at a logsonomy graph's components (degree distribution, disconnected components, shortest path length, clustering coefficient), we find that logsonomies can be broken down into more disconnected components than folksonomies. By contrast, small world properties considering the shortest path length and the clustering coefficient, as compared to random graphs

and Delicious are present.
Semantics: We explore whether similar semantic structures evolve from query logs and tagging systems. Considering relatedness measures (co-occurrence, tag context, resource context), logsonomies show slightly different characteristics. For example, the co-occurrence measure tends to reconstruct compound expressions.
Integration: We present an initial analysis about how folksonomy data can be of use for search engines. We concentrate on the fact that folksonomies provide explicit feedback of what users find relevant for specific topics. Posts in a social bookmarking system are therefore interpreted as a form of feedback (this resource is relevant for a specific tag). Different strategies to manipulate rankings based on such feedback are tested and evaluated for a learning-to-rank scenario. Overall, the strategy using the *FolkRank* algorithm (see Section 6.5) shows promising results.
The chapter is organized as follows. Section 6.2 provides a description of all datasets used in the different experiments. Section 6.3 presents the analysis of search and tagging behaviour of users and compares the content of tagging systems and ranking results. Section 6.4 deals with structural and semantic aspects of search systems and folksonomies. Section 6.5 presents a first approach how to integrate data of folksonomies and search engines by using tagging data as implicit feedback to create training data for learning-to-rank algorithms. Finally, Section 6.6 provides a summary of all findings.
The chapter is based on work published in Benz et al. [2009b, 2010a,b]; Krause et al. [2008a,b].

6.2 Datasets

A variety of datasets were used to compare social bookmarking systems and search engines. This section describes these datasets.

6.2.1 Overview of Datasets

Delicious complete In November 2006 the research team of the Knowledge Engineering Group of the University of Kassel crawled Delicious to obtain a comprehensive social bookmarking set with tag assignments from the start of the system up to October 2006 [Cattuto et al., 2007].

Delicious May only Based on the time stamps of the tag assignments, it is possible to produce snapshots. We use a snapshot from May 2006 for the comparison of tagging behaviour in social bookmarking systems to the query behaviour in search engines (Section 6.3.1).

Delicious 2005 This snapshot contains all posts which were created before July, 31st 2005. The first 40,000 tags of the latter dataset served as queries in our search engine crawls. This dataset was also used to represent folksonomy data for the topological comparison of folk- and logsonomies (Section 6.4.2).

Delicious 2008 In Section 6.5 a dataset of Delicious provided by Wetzker et al. [2008] is used for inferring implicit feedback from tagging data. The dataset contains the public bookmarks of about 980,000 users retrieved between Sep. 2007 and Jan. 2008. The dataset is similar to the *Delicious complete* dataset, but URLs were retrieved

Table 6.1: Input datasets from the social bookmarking system Delicious and the search engines MSN, AOL and Google.

Dataset Name	Date	Words/Tags	URLs
Delicious 2005	until July 2005	430,526	2,913,354
Delicious May only	May. 06	377,515	1,612,405
Delicious complete	until Oct. 06	2,741,198	17,796,405
Delicious 2008	until Dec. 2007	6,933,179	54,401,067
MSN click data	May 06	2,224,550	4,970,635
MSN crawl	Oct. 06	29,777	19,215,855
AOL click data	March - May 06	1,483,186	1,620,034
Google crawl	Jan. 07	34,220	2,783,734

until a later time. The retrieval process resulted in about 142 million bookmarks or 450 million tag assignments that were posted between Sep. 2003 and Dec. 2007.

MSN crawl and Google crawl A crawl from each MSN and Google are used to compare the content of search engines with social bookmarking systems in Section 6.3.1. For both systems we submitted the 40,000 most popular tags of the Delicious dataset as queries. We retrieved 1,000 URLs for each query in the MSN data set, and 100 URLs for each query in Google.

MSN click data We obtained a click dataset from Microsoft for the period of May 2006. The MSN dataset consists of about 15 million queries submitted in 7,470,915 different sessions which were tracked from the MSN search engine users in the United States in May 2006. The dataset was provided as part of the award "Microsoft Live Labs: Accelerating Search in Academic Research" in 2006 [1]. The data was used to analyse behavioural aspects (Section 6.3.1), to construct the logsonomy data for the analysis of structural and semantic aspects in Section 6.4 and as a source of implicit feedback in the comparison of implicit feedback strategies in 6.5.

MSN ranking data We obtained a ranking dataset from Microsoft collected in May 2006.[2] The dataset consists of about 1,6 million rankings having up to 50 ranked URLs each. The rankings were used for the learning-to-rank scenario in Section 6.5.

AOL click data The data was collected from March 1st to May 31st 2006. The dataset consists of 657,426 unique user IDs, 10,154,742 unique queries, and 19,442,629 click-through events [Pass et al., 2006]. It was used in the experiments of Section 6.3.1 and Section 6.4.

To make the click datasets (MSN and AOL click data) comparable to tags, we decomposed a query into single query words, removed stop words and words containing only one letter. All query words and all tags were converted to lowercase.
The different datasets are summarized in Table 6.1.

[1] http://research.microsoft.com/ur/us/fundingopps/RFPs/Search_2006_RFP.aspx

[2] http://research.microsoft.com/ur/us/fundingopps/RFPs/Search_2006_RFP.aspx

Table 6.2: Folksonomy and logsonomy datasets for the structural comparison of the tripartite structures built from social bookmarking and search engines.

Dataset Name	$\|T\|$	$\|U\|$	$\|R\|$	$\|Y\|$
Delicious host only URLs	430,526	81,992	934,575	14,730,683
Delicious complete URLs	430,526	81,992	2,913,354	16,217,222
AOL complete queries	4,811,436	519,250	1,620,034	14,427,759
AOL split queries	1,074,640	519,203	1,619,871	34,500,590
MSN complete queries	3,545,310	5,680,615	1,861,010	10,880,140
MSN split queries	902,210	5,679,240	1,860,728	24,204,125

6.2.2 Construction of Folk- and Logsonomy Datasets

For the experiments comparing folksonomies and logsonomies (Section 6.4.2), the two click datasets *MSN click data* and *AOL click data* had to be transformed to represent the tripartite structure of a logsonomy (see Section 6.4 for a more detailed description of a logsonomy's structure). In the dataset *MSN complete queries*, the set of tags is the set of complete queries, the set of users is the set of sessions and the set of resources is the set of clicked URLs. For the second dataset, *MSN split queries*, we decomposed each query t at whitespace positions into single terms $(t_1, \ldots, t_k)$ and collected the triples (u, t_i, r) (for $i \in 1, \ldots, k$) in Y instead of (u, t, r). This splitting better resembles the tags added to resources in folksonomies, which typically are single words.
The AOL data was transformed into the two datasets *AOL complete queries* and *AOL split queries* analogous to the MSN datasets. We used unique user IDs for the AOL dataset because session IDs were not included in the AOL dataset.
The AOL data was available with truncated URLs only. To make the MSN (and the Delicious) data comparable to the AOL data, we reduced the URLs to host-only URLs, i. e., we removed the path of each URL leaving only the host name. Then we transformed both the MSN and the AOL data to two logsonomies each as described above, resulting in the logsonomies that are described in the four last lines of Table 6.2. As we removed stopwords for the 'split queries' datasets, a minor fraction of users (1,665) and URLs (97) disappeared in the MSN case because of their relation to a query consisting only of stopwords.
Folksonomy data was represented by the *Delicious 2005* data set (see Table 6.1) containing posts from $81,992$ users up to July 31st 2005. From this we created two datasets: one consisting of full URLs to be comparable to prior work on folksonomies and one reduced to only the host part of the URL to be comparable to the logsonomy datasets. The sizes of the created folksonomies are presented in the first two lines of Table 6.2. The frequency distributions of the folk- and logsonomy datasets generated will be shown in Section 6.4.1.
In Section 6.4.3, in order to study semantic aspects of a logsonomy, we used the *Delicious host only URLs*. We restricted the dataset to the 10,000 most frequent tags and to the resources/users that have been associated with at least one of those tags. The dataset was also used in the analysis of tag relatedness [Cattuto et al., 2008].
For the logsonomy representation we used the click dataset from the *AOL split queries*. Once again, we constructed a logsonomy, this time with the restriction of only using the 10,000 most frequent query words of the dataset. The resulting sizes of the datasets are shown in Table 6.3.

Table 6.3: Folksonomy and logsonomy datasets for the comparison of semantic aspects of the tripartite structures built from social bookmarking and search engines.

dataset	$\|T\|$	$\|U\|$	$\|R\|$	$\|Y\|$
Delicious reduced	10,000	476,378	12,660,470	101,491,722
AOL split queries reduced	10,000	463,380	1,284,724	26,227,550

Taking only the 10000 most frequent tags might raise the question if information is not lost in the process. More rarely used tags might provide a higher information content. However, the sparsity of such tags makes them less useful for the study of the semantic measures applied [Cattuto et al., 2008].

6.2.3 Construction of User Feedback Datasets

For our experiments in Section 6.5, we combine three different kinds of data: Ranking data (*MSN ranking data*), click data (*MSN click data*) and social bookmarking data (*Delicious 2008*).
For about 700,000 queries from the MSN click dataset we have the same queries with a set of ranked URLs in the MSN ranking dataset and with at least one URL clicked in the MSN click dataset.
To be comparable to the ranking and click datasets, for the *Delicious 2008*, we only consider posts before end of May 2006 [3]. Tags are normalized by splitting all queries into single, lower case terms and all characters expect the letters and numbers are removed. Furthermore, only those posts are filtered which match a query-doc pair in the click dataset. We therefore normalize the queries in the same manner as done with the tags and filter the posts which have the same URL and contain at least all query terms as tags. Overall, we get 36,830 queries with rankings where at least one URL in the ranking has been tagged in Delicious together with the corresponding query terms. This results in 263,171 users, 11,264,441 resources and 1,390,878 tags.

6.3 Comparison of Searching and Tagging

In this section we study the user behaviour and content of search and tagging systems. We will concentrate on three aspects: Are query words and tags used in a similar way (6.3.1)? Is tagging and search behaviour correlated over time (6.3.1)? And, is the content of both systems similar (6.3.1)?

6.3.1 Analysis of Search and Tagging Behaviour

First, we will compare the behaviour of searchers and taggers. Search behaviour is described by the query terms submitted to a search engine. We use the number of occurrences of a term in the queries over a certain period of time as an indicator for the users'

[3] The experiments could have been conducted with the dataset *Delicious complete*. As *Delicious 2008* even reduced was bigger than *Delicious complete*, we decided to use this one and get as much overlap as possible for the query and tag terms.

Table 6.4: Statistics of item frequencies in Delicious and MSN in May, 2006

	MSN	**Delicious**	**MSN - Del.**
items	31,535,050	9,076,899	—
distinct items	2,040,207	375,041	96,988
average frequency	15.46	24.20	—
frequent items $\geq$ 10 occurrences	115,966	39,281	18,541
frequent items containing "_"	90	1,840	1
frequent items containing "-"	1,643	1,603	145
frequent items cont. "www.", ".com", ".net" or ".org"	17,695	136	30

interests. The interests of taggers in social bookmarking systems, on the other hand, are described by the tags they assign to resources over a certain period of time.
We start with a comparison of the overlap of the set of all query terms in the *MSN click data* with the set of all tags in the dataset *Delicious May only* (see Section 6.2). This comparison is followed by an analysis of the correlation of search and tagging behaviour in both systems over time. Query log files were not available for the bookmarking systems, hence we only study the tagging (and not the search) behaviour.

Query Term and Tag Usage Analysis

By comparing the distribution of tags and query terms we gain first insights into the usage of both systems. The overlap of the set of query terms with the set of tags is an indicator of the similarity of the usage of both systems. We use the *Delicious May only* to represent social bookmarking systems and the *MSN click data* to represent search engines.
Table 6.4 shows statistics about the usage of query terms in MSN and tags in Delicious. The first row reflects the total number of queried terms and the total number of tags used in Delicious. The following row shows the number of distinct items in all systems. As can be seen, both the total number of terms and the number of distinct terms is significantly larger in MSN when compared to the total number of tags and the number of distinct tags in Delicious. Interestingly, the average frequency of an item is quite similar in all systems (see third row). These numbers indicate that Delicious users focus on fewer topics than search engine users, but that each topic is, on average, equally often addressed.
Figure 6.1 shows the distribution of items in both systems on a log-log scale. The x-axis denotes the count of items in the data set, while the y-axis describes the number of tags that correspond to the term/tag occurrence number. We observe a power law in both distributions.
A power law in this case means that the vast majority of terms only appears once or very few times, while only a few terms are used frequently (see details in 2.3). This effect also explains the relatively small overlap between the MSN query terms and the Delicious terms, which is given in the 2nd row/3rd column of Table 6.4. In order to analyse the overlap for the more frequent terms, we restricted both sets to query terms/tags that showed up in the respective system at least ten times.[4] The resulting frequencies are given in the first line of the second part of Table 6.4. It can be seen that the sizes of the reduced MSN and Delicious datasets thereby become more equal and that the relative overlap increases.

[4]The restriction to a minimum of 5 or 20 occurrences provided similar results.

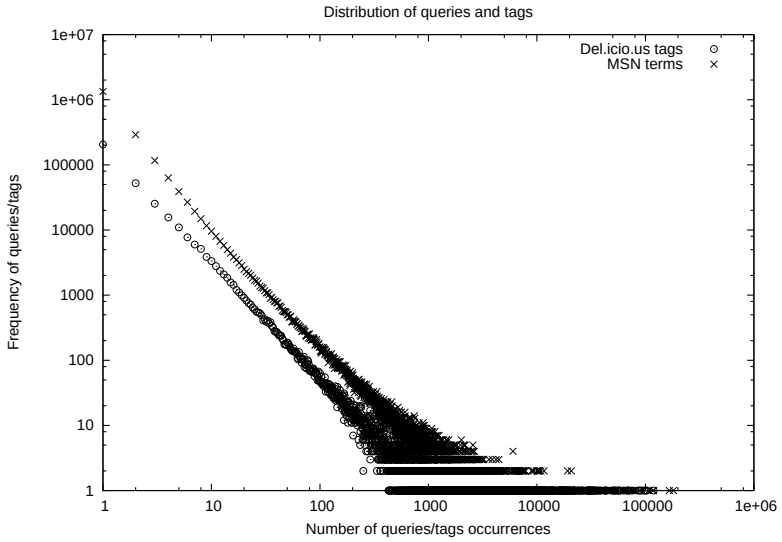


Figure 6.1: Distribution of items in Delicious and MSN on a log-log scale. The x-axis denotes the count of items in the data set, the y-axis describes the number of tags that correspond to the term/tag occurrence number. We observe a power law in both distributions.

When browsing both reduced data sets we observed that the non-overlapping parts result very much from the different usages of both systems. In social bookmarking systems, for instance, people frequently encode multi-word lexems by connecting the words with either underscores, hyphens, dots, or no symbol at all. (For instance, all of the terms 'artificial_intelligence', 'artificial-intelligence', 'artificial.intelligence' and 'artificialintelligence' show up at least ten times in Delicious). This behaviour is reflected by the second and third last rows in Table 6.4. Underscores are basically only used for such multi-word lexemes, whereas hyphens occur also in expressions like 'e-learning' or 't-shirt'. Only in the latter form do they show up in the MSN data.

A large part of the query terms in MSN that are not Delicious tags are URLs or part of URLs (see the last row of Table 6.4). This indicates that users of social bookmarking systems prefer tags that are closer to natural language, and thus easier to remember, while users of search engines (have to) anticipate the syntactic appearance of what they are looking for.

The top five tags of Delicious and the top five terms of MSN in May 2006 can be seen in Table 6.5 together with their frequencies. One can see that Delicious has a strong bias towards computer science related terms. Eleven of the 20 top tags are computer terms (such as web, programming, ajax or linux). The top terms in MSN are more difficult to interpret. "yahoo" and "google" may be used when people have the MSN search interface as a starting point in their internet explorer, or when they leave Microsoft related programs such as hotmail, and want to use another search engine. "county" is often part of a composed query such as "Ashtabula county school employees credit union" or "county state bank". We lack a good explanation for the high frequency of this term. This might result from the way Microsoft extracted the sample (which is unknown to us).

Table 6.5: The top five items and their frequencies of Delicious and MSN in May 2006. Delicious has a strong bias towards computer science related terms. For MSN no specific topic-relation can be found.

Tags Del	Frequency	Query terms MSN	Frequency
design	119,580	yahoo	181,137
blog	102,728	google	166,110
software	100,873	free	118,628
web	97,495	county	118,002
reference	92078	myspace	107,316

Correlation of Search and Tagging Behaviour over Time

Up to now we have considered both data collections as static. In the next section we analyse if and how search and tagging behaviour are correlated over time. Again we use the MSN query data and the Delicious data of May 2006. Each data set has been separated into 24-hour bins, one for each day of May 2006. As the unit of analysis we selected those tags from Delicious that also appeared as a query term in the MSN click data. In order to reduce sparse time series, we excluded time series which had fewer than five daily query or tagging events. In total, 1003 items remained.

For each item i, we define two time series. The Delicious time series is given by $X_i^d = (x_{i,1}^d, ..., x_{i,31}^d)$, where $x_{i,t}^d$ is the number of assignments of tag i to some bookmark during day $t \in \{1, \ldots, 31\}$. For MSN, we define $X_i^m = (x_{i,1}^m, ..., x_{i,31}^m)$, where $x_{i,t}^m$ is the number of times this term was part of a query on day t according to the MSN data.

The data was normalized in order to reduce seasonal effects. We chose an additive model for removal of seasonal variation, i. e., we estimated the seasonal effect for a particular weekday (e. g., Monday) by finding the average of each weekday observation minus the corresponding weekly average and subtracted this seasonal component from the original data [Chatfield, 2004]. The model underlies the assumption that no substantial (i. e., long-term) trend exists which otherwise would lead to increasing or decreasing averages over time. As our time period is short, we assume that long term trends do not influence averages. We also smoothed the data using simple average sine smoothing [Ivorix, 2007] with a smoothing window of three days to reduce random variation. Other smoothing techniques were also tested but they delivered similar results.

In order to find out about the similarity of the two time series of an item i we used the correlation coefficient between the two random variables $x_{i,t}^d$ and $x_{i,t}^m$ which is defined as $r = \frac{\sum_t (X_{i,t}^d - \mu(X_i^d))(X_{i,t}^m - \mu(X_i^m))}{\sigma(X_i^d)\sigma(X_i^m)}$ where $\mu(X_i^d)$ and $\mu(X_i^m)$ are the expected values and $\sigma(X_i^d)$ and $\sigma(X_i^m)$ are the standard deviations.

We applied the t-test for testing significance using the conventional probability criterion of .05. For 307 out of 1003 items we observed a significant correlation. We take this as indication that tagging and searching behaviour are indeed triggered by similar motivations. The highest correlation has the item 'schedule' ($r = 0.93$), followed by 'vista' ($r = 0.91$), 'driver', 'player' and 'films'. While both 'schedule' time series are almost constant, the following item 'vista' has a higher variance, since a beta 2 version of Microsoft's Vista operating system was released in May 2006 and drew the attention of searchers and taggers. The 'vista' time series are given in the left of Figure 6.2. Another example where the peaks

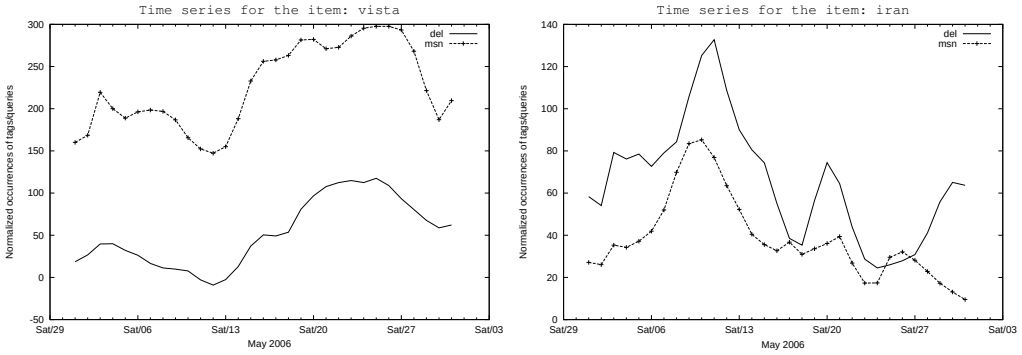


Figure 6.2: Time series of two highly correlated items, "vista" and "iran". The straight line shows Delicious, the dashed line MSN. For each day in May 2006, the normalized occurrences of the item are shown.

in the time series were triggered by an information need after a certain event is "iran" ($r = 0.80$), which has the 19th highest correlation of all tags. The peaks show up shortly after the confirmation of the United States White House that Iran's president sent a letter to the president of the USA on May 08, 2006. The two curves are strongly correlated. A similar peak for 'iran' can be observed in Google Trends[5] which show Google's search patterns in May 2006. These examples support the hypothesis that popular events trigger both search and tagging during the time period close to the event.

Coverage of Delicious with MSN, Google and AOL

In this section we shift our focus from query terms and tags to the underlying resources, i. e., the URLs. Considering the size of the Web, both search engines (in particular the part we can crawl) and folksonomies constitute only a small fraction of the Web. An interesting question is thus if there is any significant overlap between the URLs provided by both systems.

To compare the coverage of the different data sets, we compute the overlaps between *MSN crawl*, *Google crawl*, *AOL click data* and the *Delicious complete* data sets (see Section 6.2). As we had no access to the index of the search engines, we crawled all search engines with 1,776 queries to obtain comparable datasets. These queries were determined by taking the 2000 most popular tags of the *Delicious 2005* dataset and intersecting them with the set of all AOL items.

In order to see whether Delicious contains these URLs that were considered relevant by the traditional search engines, we computed a kind of "recall" for folksonomy-URLs on the other data sets as follows. First we cut each of the 1,776 rankings of each search data set after the first 25, 50, 75 and 100 URLs. For each ranking size we computed the intersection with all Delicious URLs. As the AOL log data consist of domain names only (and not of full URLs), we also pruned the URLs of the other systems in a second step to only include the domain names.

Table 6.6 shows the results. The first number in each cell is the average number of overlaps for the original URLs, the second for the pruned URLs. Google shows the highest overlap

[5] `http://www.google.com/trends?q=Iran&geo=all&date=2006-5`

Table 6.6: Averages of overlapping URLs computed over 1776 rankings of all Delicious URLs from the *Delicious complete* dataset with the search datasets, i. e., *MSN crawl*, *Google crawl*, *AOL click data*. The rankings were cut after 25, 50, 75 and 100 URLs. The first number in each cell is the average number of overlaps for the original URLs, the second the pruned URLs. Google shows the highest overlap with Delicious.

Dataset	**top 25**	**top 50**	**top 75**	**top 100**
Google	19.91 / 24.17	37.61 / 47.83	54.00 / 71.15	69.21 / 85.23
MSN	12.86 / 20.20	22.38 / 38.62	30.93 / 56.47	39.09 / 74.14
AOL	— / 19.61	— / 35.57	— / 48.00	— / 57.48

with Delicious, followed by MSN and then AOL. For all systems, the overlap is rather high. This indicates that, for each query, both traditional search engines and folksonomies focus on basically the same subset of the Web. The values in Table 6.6 will serve as upper bounds for the comparison of ranking overlaps in folksonomies and search engines in Section 6.3.2.
Furthermore, the top rankings show more coverage: While in average 24.17 URLs in the top Google 25 ranking are represented in Delicious, only 85.23 are represented in the top 100 URLs in average. This indicates that the top entries of search engine rankings are – compared to the medium ranked entries – also those which are judged more relevant by the Delicious users.

Conclusions of Section 6.3.1

The collection of all Delicious tags is only about a quarter of the size of the MSN queries, due to a very high number of very infrequent items in both systems (Section 6.3.1, Table 6.4). Once the sets are reduced to the frequent items, the relative overlap is higher. The remaining differences are due to different usage, e. g., to the composition of multi-word lexems to single terms in Delicious, and the use of (parts of) URLs as query terms in MSN.
We have seen that, for a relatively high number of items, the search and tagging time series were significantly correlated. We have also observed that important events trigger both search and tagging without significant time delay, and that this behaviour is correlated over time.
Considering the fact that both the available search engine data and the folksonomy data cover only a minor part of the WWW, the overlaps of the sets of URLs of the different systems (as discussed in Section 6.3.1) are rather high, indicating that users of social bookmarking systems are likely to tag web pages that are also ranked highly by traditional search engines. The URLs of the social bookmarking system over-proportionally match the top results of the search engine rankings. A likely explanation is that taggers use search engines to find interesting bookmarks.

6.3.2 Analysis of Search and Tagging System Content

In the previous section we compared the user interaction in social bookmarking systems and search engines and the coverage of URLs from folksonomies in search engines. In

this section we focus on ranking algorithms. Are overlapping results different when we introduce a ranking to the folksonomy structure? Are important URLs in search engines the same ones as important URLs in social bookmarking systems? Is the ranking order within the overlap the same? These questions will be answered below.

For the commercial search engines, we rely on our crawls and the data they provided, as the details of their ranking algorithms are not published (beside early papers like Page et al. [1998]). To rank URLs in social bookmarking systems we used two well-known ranking approaches: the traditional vector space approach with TF-IDF weighting and cosine similarity and FolkRank [Hotho et al., 2006a], a link-based ranking algorithm similar to PageRank [Page et al., 1998], which ranks users, resources or tags based on the tripartite hypergraph of the folksonomy (a description of these algorithms can be found in Section 2.5.1).

Overlap of ranking results

To compare the overlap of rankings we start with an overview of the average intersection of the top 50 URLs calculated for all datasets. In this case we based the analysis on the normalized URLs of the same datasets as used in Section 6.3.1. Table 6.7 contains the average overlap calculated for the sets of normalized URLs and the TF, TF-IDF and FolkRank rankings of the Delicious data. We see that the overlap of Delicious Oct. 2006 with the result sets of the three commercial search engines is low. The average overlap of the MSN and Google crawl rankings, however, is considerably bigger (11.79) – also when compared to the AOL results, which are in a similar range with the Delicious data. The two major search engines therefore seem to have more in common between them than folksonomies with search engines.

The TF and TF-IDF based rankings show a surprisingly low overlap with Google, MSN and AOL, but also with the FolkRank rankings for Delicious. This indicates that – for web search – graph-based rankings provide a view about social bookmarking systems that is fundamentally different to pure frequency-based rankings.

Even though the graph-based ranking on Delicious has a higher overlap with the search engine rankings than TF-IDF, it is still very low when compared to the potential values one could reach with a 'perfect' folksonomy ranking, e. g., an average overlap of 47.83 with the Google ranking, as shown in Table 6.6. The remaining items are thus contained in the Delicious data, but FolkRank ranked them beyond the top 50.

To investigate this overlap further, we have extended the Delicious result sets to the top 100 and top 1,000 URLs, resp..

Table 6.8 shows the average overlap of the top 100 and the top 1,000 normalized URLs from the FolkRank computations with Delicious data from Oct. 2006 to the top 50 normalized URLs in the Google crawl, MSN crawl and AOL log data. This can be seen in the middle column of Table 6.7. For Google, for instance, this means that the relative average overlap is $\frac{6.65}{50} \approx 0.13$ for the top 50, $\frac{9.59}{100} \approx 0.10$ for the top 100, and only $\frac{22.7}{1000} \approx 0.02$ for the top 1000. This supports our finding from Section 6.3.1, that the similarity between the FolkRank ranking on Delicious and the Google ranking on the Web is higher for the top than for the lower parts of the ranking.

Table 6.7: Average overlap of top 50 normalized URLs from 1,776 rankings of MSN, AOL and Google with the corresponding normalized URLs from the TF, TF-IDF and FolkRank rankings of the Delicious data. We see that the overlap of Delicious rankings with the result sets of the three commercial search engines is low. The average overlap of the MSN and Google crawl rankings is considerably bigger (11.79). The two major search engines therefore seem to have more in common between them than folksonomies with search engines.

	Google	**MSN**	**Del FolkRank**	**Del TF-IDF**	**Del TF**
AOL	2.39	1.61	2.30	0.30	0.21
Google		11.79	6.65	1.60	1.37
MSN			3.78	1.20	1.02
Del FolkRank				1.46	1.79
Del TF-IDF					49.53

Table 6.8: Average overlap with top 100/1,000 normalized Delicious URLs

	Google top 50	**MSN top 50**	**AOL top 50**
Del 100	9.59	5.00	1.65
Del 1000	22.72	13.43	5.16

Correlation of rankings

After determining the coverage of folksonomy rankings in search engines, one further question remains: Are the rankings obtained by link analysis (FolkRank) and term frequencies / document frequencies (TF-IDF) correlated to the search engine rankings? Again, we use the rankings of the 1,776 common items from Section 6.3.1. As we do not have interval-scaled data, we select the Spearman correlation coefficient $r_s = 1 - \frac{6 \sum d^2}{n(n^2-1)}$, where d denotes the difference of ranking positions of a specific URL and n the size of the overlap.[6]

In Section 6.3.2 we showed that the overlap of the rankings is generally low. We therefore only compared those rankings having at least 20 URLs in common. For each such item, the Spearman coefficient is computed for the overlap of the rankings. Table 6.9 shows the results. The AOL comparisons to Delicious (using the link-based method as well as TF-IDF) do not show sufficient overlap for further consideration. The Google and MSN comparisons with the link-based FolkRank ranking in Delicious yield the highest number of ranking intersections containing more than 20 URLs (Google 361, MSN 112). Both Google and MSN show a large number of positive correlations. For instance, in Google we have 326 positive correlations, where 176 are significant. This confirms our findings from Section 6.3.2.

From the results above we derive that, if overlap exists, a large number of rankings computed with FolkRank are positively correlated with the corresponding search engine rankings. In order to find on which topics the correlation is high, we extracted the top ten correlations of the Delicious FolkRank with Google and MSN, resp., (see Table 6.10). We

[6] In Bar-Ilan et al. [2006], enhancements to Kendall's tau and Spearman are discussed to compare rankings with different URLs. These metrics are heavily influenced if the intersection between the rankings is small. Because of this we stick to the Spearman correlation coefficient.

Table 6.9: Correlation values and number of significant correlations from rankings having at least 20 URLs in common. The Google and MSN comparisons with the FolkRank ranking in Delicious yield the highest number of ranking intersections containing more than 20 URLs (Google 361, MSN 112).

Datasets	# overlap > 20)	Avg. correlation	Avg. of significant correlations	# correlated rankings	# significant correlated rankings
			pos/neg	pos/neg	pos/neg
Google/FolkRank	361	0.26	0.4/-0.17	326/37	176/3
Google/TF-IDF	17	0.17	0.34/0	15/2	5/0
MSN/FolkRank	112	0.25	0.42/-0.01	99/13	47/1
MSN/TF-IDF	6	-0.21	-/-	2/4	0/0
AOL/FolkRank	1	0.25	-/-	1/0	0/0
AOL/TF-IDF	1	0.38	0.38/-	1/0	1/0

Table 6.10: Intersections and correlations for the top 10 correlations of the Delicious FolkRank rankings with Google rankings (left) and MSN rankings (right). The ranking's contained 100 URLs. Most items in this set are IT related.

Item	**Inters.**	**Correlation**
technorati	34	0.80
greasemonkey	34	0.73
validator	34	0.71
tweaks	22	0.68
metafilter	24	0.67
torrent	29	0.65
blender	22	0.62
torrents	30	0.62
dictionaries	21	0.62
timeline	21	0.62

Item	**Inters.**	**Correlation**
validator	21	0.64
subversion	22	0.60
furl	23	0.59
parser	27	0.58
favicon	28	0.57
google	25	0.57
blogosphere	21	0.56
jazz	26	0.56
svg	23	0.55
lyrics	25	0.54

found that most items in this set are IT related. As a major part of Delicious consists of IT related contents, we conclude that link-based rankings for topics that are specific and sufficiently represented in a folksonomy yield results similar to search engine rankings.

Conclusions of Section 6.3.2

In Section 6.3.2 we have seen that a comparison of rankings is difficult due to sparse overlaps of the data sets. It turned out that the top hits of the rankings produced by FolkRank are closer to the top hits of the search engines than the top hits of the vector based methods. Furthermore, we could observe that the overlap between Delicious and the search engine results is larger in the top parts of the search engine rankings.
We also observed that the folksonomy rankings are more strongly correlated to the Google rankings than to those of MSN and AOL, whereby the graph-based FolkRank is closer to the Google rankings than TF and TF-IDF. Again, we assume that taggers preferably use search engines (and most of all Google) to find information they then proceed to tag. A qualitative analysis showed that the correlations were higher for specific IT topics, where

Delicious has a particularly good coverage.

6.3.3 Discussion

In this section we conducted an exploratory study to compare social bookmarking systems with search engines. We concentrated on information retrieval aspects by analyzing search and tagging behaviour as well as ranking structures. We were able to discover both similar and diverging behaviour in both kinds of systems, as summarized in the conclusions of Sections 6.3.1 and 6.3.2.
A question still open is whether, with more data available, the correlation and overlap analyses could be set up on a broader basis. However, a key question to be answered first is: what is to be considered a success? Is it desirable that social search tries to approximate traditional web search? Is Google the measure of all things?
Computing overlap and comparing correlations helped us finding out about the similarities between the systems. However, we have no information about which approach offers more relevant results from a user's perspective. A user study in which users create a benchmark ranking and performance measures would be of benefit. Further investigation also has to include a deeper analysis of where URLs show up earlier as well as the characteristics of both system's URLs that are not part of the overlap.

6.4 Properties of Logsonomies

In this section, we look at structural and semantic characteristics of the tripartite structure of click data, i. e., logsonomies and compare it to the tripartite structure of folksonomies. First, we look at the degree distribution of users, tags and resources in folk- and logsonomies. Afterwards, in Section 6.4.2, we discuss the topological structure of logsonomies. The section is part of the work presented in Krause et al. [2008b]. Section 6.4.3 summarizes semantic aspects. A complete analysis can be found in Benz et al. [2009b].

6.4.1 Degree distribution

In Section 6.3.1, we have seen that the tag distribution of the Delicious folksonomy follows a power law, and that the MSN terms follow a similar distribution. We will analyse now, if this holds also for the logsonomies. We will consider all three dimensions of the folksonomies, i. e., tags, resources, and users.
When considering a folk-/logsonomy as a hypergraph, the count of an item equals its degree in the graph. We will therefore also use the notion of *degree distribution*.
The distributions of tags, resources, and users are plotted in Figures 6.3, 6.4, and 6.5, resp. These plots are similar to the one in Figure 6.1, with the following differences:

- As we now consider logsonomies, we restrict ourselves in the search engine data to those resources which actually have been clicked by some user, and to the corresponding query words and users. The plot in Figure 6.1, on the other hand, is based on *all* tags, as our aim in Section 6.3.1 was to compare the content of the systems rather than the user behaviour.

- The y-axis displays now relative rather than absolute values, to make the curves better comparable. In all plots, all curves start thus in the upper left corner (which

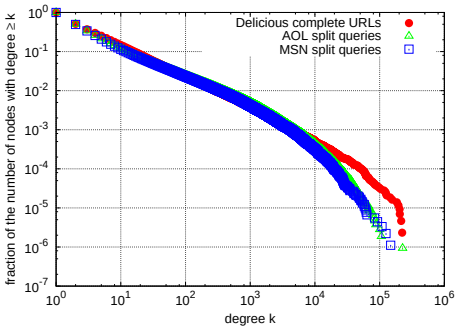


Figure 6.3: Degree distribution of tags/query words/queries. The degree k is plotted against the fraction of nodes with at least this degree.

is stating the trivial observation that 100 % (=10^0) of all items in all datasets have a degree of ≥ 1).

- The plot now shows the *cumulated* degree distribution. While in Figure 6.1 the y-axis showed the number of items with a degree of exactly k, it now shows the number of items with a degree $\geq k$. The non-cumulated version has the advantage that each data point refers to exactly one item (e. g., the query word 'yahoo' being the rightmost box in Figure 6.1), but the very few items at the right end of the curve 'mess up' the diagram, making it more difficult to read. The cumulated plot smoothes the right end of the curves, without loosing any information.[7]

- For MSN, we use now the 'host only' version of the URLs, to be comparable with the AOL data, as discussed in Section 6.2.2. For Delicious, only the 'host only' curve is relevant for comparison with the other systems. Its 'complete URLs' curve was plotted only to verify that the restriction to the host part does not bias the results.

Term Distributions

The distributions of the terms (i. e., tags of Delicious and query words/whole queries of the search engines) are plotted in Figure 6.3. We observe that all datasets except the AOL and MSN complete queries datasets have a very similar behaviour, and differ only for the frequently used terms (i. e., those with very high degree). The stronger decrease of the distributions for the two latter results from the fact that it is less likely for different users to share full queries than to share single query words, and that very many queries are submitted only once. We attribute the relatively higher amount of frequent queries in MSN compared to AOL to the larger size of the MSN sample, which makes the frequent response to the same queries more likely.
The frequency of usage of query words in search engines[8] and of tags in Delicious is very similar, as the other distributions show. For the small and medium degrees, they

[7] In the cumulated version, the power law is called *Pareto law* Adamic [2002]. There is a $1:1$ correspondence between the non-cumulated and the cumulated version. The only difference is that the slope of the line is increased by 1, as the cumulated version results from the non-cumulated one by integration, which increases the degree of the exponent of a power by 1.

[8] Note that in a logsonomy 'usage of a query word' means the frequency of clicking on hits that were returned for that query word.

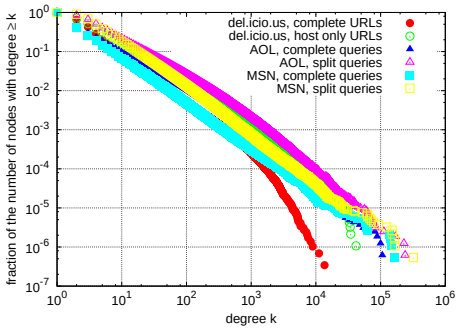


Figure 6.4: Degree distribution of resource nodes. The x-axis shows the degree k, the y-axis the fraction of resources with at least this degree.

are remarkably similar, and differ only for the very frequent terms. While the Delicious distribution follows a power law, the AOL and MSN curves are concave; indicating that they have a lesser amount of very frequent terms than expected. We conclude that the agreement on the most central terms is less strong among the search engine users, but that the distributions of moderately used terms is very similar.
The very small difference between the two Delicious curves indicates that we do not disturb the distribution by considering only the host part of the URLs — which was necessary to be comparable with the AOL data.

Resource Distributions

For all datasets (except the non-competitive Delicious complete URLs dataset), the resource distributions are surprisingly similar to each other (cf. Figure 6.4). Not only do they all form straight lines and are thus clearly power law, but additionally all show the same slope. We consider this as a strong indicator that all datasets reflect the same underlying distribution of the interests of Web users — even though the coverage of the search engines is broader than the coverage of Delicious.
N.B.: The strong deviation of the distribution for the 'Delicious complete URLs' was to be expected, since reducing URLs to their host aggregates their frequencies to the frequency of the host name, which is thus expected to be higher.

User Distributions

Figure 6.5 shows the distributions of users for the different datasets. They are all concave, and have thus more moderately active users and less very active users than expected for a power law. Although they still follow the overall scheme — the majority of users is very inactive, while only very few users are very active — one cannot claim the existence of a power law any more. This indicates that models which generate power law distributions — like preferential attachment Barabasi and Albert [1999] — are not adequate for describing the user activity in folksonomies nor in logsonomies.
The slopes of the user distributions have a high variance. Delicious shows the highest relative amount of very frequent users, followed by AOL and — with a significant distance — MSN. The latter is likely to be due to the nature of sessions representing the users in this dataset: though long-term cookies to track users exist in MSN, sessions have a shorter life

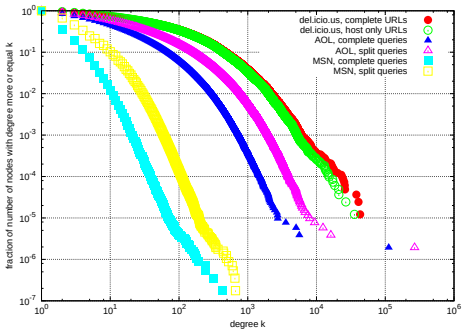


Figure 6.5: Degree distribution of user nodes. The x-axis shows the degree k, the y-axis the fraction of users with at least this degree.

time as opposed to unique, timeless user IDs as present in the AOL and Delicious datasets. The probability of being strongly interlinked is therefore lower. The difference between AOL and Delicious indicates that the activity of the very active users is significantly higher in the bookmarking system.
For both AOL and MSN, the curves for the complete queries are below the curves for the split queries. This difference is systematic, because it is less likely to reuse complete queries than to reuse single query words.
N.B.: The very small difference between the two Delicious curves indicates again that we do not disturb the distribution by considering only the host part of the URLs.

6.4.2 Structural Properties

Folksonomies exhibit small world characteristics (see Section 2.3): they contain a graph topology for which the clustering coefficient is higher than the one of a random graph but the average shortest path length is almost as small as that of a random graph [Watts and Strogatz, 1998]. These characteristics are one explanation for the popularity of social bookmarking systems: on the one hand, resources fulfilling a specific information need are clustered together in the folksonomy, while on the other hand, users can reach most of the contents within a few clicks.
In the following sections we investigate to which extent these characteristics hold for logsonomies. We created six folk- and logsonomy datasets from the available Delicious, MSN and AOL data as described in Section 6.2.2.
We followed the experiments of Cattuto et al. [2007] who compared folksonomies to random graphs in order to be comparable to former findings regarding folksonomy properties. In these experiments, binomial and shuffled (hyper-)graphs of the same size as the original folksonomy were selected to compare the original graph to random graphs. For a given folksonomy (U, T, R, Y), a *binomial* random graph is a folksonomy $(U, T, R, \hat{\mathbf{Y}})$ where $\hat{\mathbf{Y}}$ consists of $|Y|$ randomly drawn tuples from $U \times T \times R$. A *shuffled* random graph is then a folksonomy $(U, T, R, \check{Y})$ where $\check{Y}$ is derived from Y by randomly shuffling all occurrences of tags in Y, followed by shuffling all occurrences of the resources. (For a complete shuffling, it is sufficient to shuffle any two of the three dimensions.) The binomial graph has thus the same number of tag assignments as the original graph, while the shuffled graph has additionally the same degree distribution.

Table 6.11: Average shortest path lengths, cliquishness and connectedness of each dataset and the corresponding random graphs.

	ASPL			cliquishness			connectedness		
dataset	raw	shuffled	binomial	raw	shuffled	binomial	raw	shuffled	binomial
Delicious, complete URLs	3.59	3.08	3.99	0.86	0.55	0.20	0.85	0.37	0.00
Delicious, host only URLs	3.48	3.06	3.67	0.75	0.51	0.05	0.83	0.32	0.00
AOL, complete queries	4.11	3.81	5.76	0.85	0.66	0.32	0.33	0.03	0.00
AOL, split queries	3.62	3.20	3.90	0.70	0.43	0.04	0.66	0.10	0.00
MSN, complete queries	5.43	4.10	8.78	0.87	0.75	0.47	0.42	0.03	0.00
MSN, split queries	3.94	3.42	5.48	0.85	0.50	0.23	0.70	0.11	0.00

Average Shortest Path Length

The average shortest path length (ASPL) denotes the mean shortest distance between any two nodes in the graph. In a tripartite hypergraph, a path between any two nodes is a sequence of hyperedges that lie between them. The shortest path is a path with the minimum number of hyperedges connecting the two nodes.

For complexity reasons, we approximated the average shortest path length as follows. For each of the datasets, we randomly selected 4,000 nodes and calculated the shortest path length of each of those nodes to all other nodes in its connected component.

Table 6.11 shows the average shortest path length of each dataset together with the values for the corresponding random graphs. Comparing the two Delicious datasets, the average shortest path length does not vary greatly when considering host only URLs (3.48 for the host-only-graph versus 3.59 for the graph with complete URLs). The average shortest path length of the AOL and MSN datasets with split queries are smaller than those of the datasets with complete queries. This can be explained by the higher overlap which occurs when splitting the queries. As a side effect, this also leads to a mixing of contents, e. g., the word *java* in *java programming language* and *java island* will link to different topics. However, such wording issues also exist in folksonomies.

Compared to Delicious, all four datasets from MSN and AOL show larger path lengths. Capturing the intuition of serendipitous browsing, it takes longer to reach other queries, users, or URLs within a logsonomy than it takes to jump between tags, users and resources in a folksonomy. In particular, the high values for MSN are likely to result from the fact that a user cannot bridge between different topics if he searched for them in different sessions.

Small world properties are still confirmed by the shortest path length: when comparing each logsonomy to the corresponding binomial and random graphs, the path lengths differ only slightly.

Clustering Coefficient

The clustering coefficient characterizes the density of connections in the environment of a node. It describes the cliquishness, (i. e., *are neighbor nodes of a node also connected among each other*) and the connectedness of a node, (i. e., *would neighbor nodes stay acquainted if the node was removed*). In a tripartite graph, one needs to consider these two characteristics separately. In Cattuto et al. [2007], two measures were proposed, which are summarized in the following paragraphs.

Cliquishness. Consider a resource r. Then the following sets of tags T_r and users U_r are said to be connected to r: $T_r = \{t \in T \mid \exists u \in U\colon (t,u,r) \in Y\}$, $U_r = \{u \in U \mid \exists t \in T\colon (t,u,r) \in Y\}$. Furthermore, let $tu_r := \{(t,u) \in T \times U \mid (t,u,r) \in Y\}$, i. e., the (tag, user) pairs occurring with r. If the neighborhood of r was maximally cliquish, all of the pairs from $T_r \times U_r$ would occur in tu_r. So we define the cliquishness coefficient $\gamma_{cl}(r)$ as:

$$\gamma_{cl}(r) = \frac{|tu_r|}{|T_r| \cdot |U_r|} \quad \in [0,1] \ . \tag{6.1}$$

The cliquishness is defined likewise for tags and users.
Connectedness. Consider a resource r. Let $\widetilde{tu_r} := \{(t,u) \in tu_r \mid \exists \tilde{r} \neq r\colon (t,u,\tilde{r}) \in Y\}$, i. e., the (tag, user) pairs from tu_r that also occur with some other resource than r. Then we define:

$$\gamma_{co}(r) := \frac{|\widetilde{tu_r}|}{|tu_r|} \quad \in [0,1] \ . \tag{6.2}$$

γ_{co} is thus the fraction of r's neighbor pairs that would remain connected if r were deleted. It indicates to what extent the surroundings of the resource r contain "singleton" combinations, i. e., *tag - user* that only occur once. The connectedness is defined likewise for tags and users.
The results in Table 6.11 show that the average cliquishness and connectedness coefficients of the original AOL, MSN and Delicious graphs are in general higher than the ones of the corresponding random graphs. This indicates that there is some systematic aspect in the search behaviour which is destroyed in the randomized versions. Comparing the two logsonomies to the folksonomy, however, one can conclude that the clustering coefficients of the folksonomy exceeds those of logsonomies. This is probably due to the higher variety of topics in the logsonomy datasets – whereas Delicious is very focused on computer related terms. Additionally, users in folksonomies tend to add similar tags to a resource, while resources in web search engines will be retrieved by many different queries. This relates to the issues described in Section 3.2.4: The process in which tags are created in logsonomies and folksonomies is different. Therefore, logsonomies show a tendency towards reflecting the connectedness and small world properties which have been found in folksonomies. Due to their diverse topical structure and the different process of gathering the data (from different search rankings and their clicks), however, the tripartite network of clickdata is less connected and cliquish than the one built from tagging data.

6.4.3 Semantic Properties

The previous section revealed that folksonomies and logsonomies show similar structural characteristics, e. g., small world properties. These findings support the idea of enabling some kind of browsing facilities in search engines. Another exciting property of folksonomies is the inherent semantic which occurs with the process of tagging. We now present experiments from Benz et al. [2009b], who investigate to which extend a logsonomy allows the extraction of semantics emerging from the "collaborative" process of searching for similar information and being interested in the same resources. Again, results are compared to results of the same experiments conducted on folksonomy data. The analysis is based on the datasets *Delicious reduced* and *AOL split queries reduced* as presented in Section 6.2.2.

Relatedness Measures

In order to find semantic similarities, one needs to define measures which allow for extracting related items. Several relatedness measures have been introduced in Cattuto et al. [2008] to extract semantic similarities between query parts from folksonomies. They can be applied to logsonomies in a similar way:
Given a logsonomy (U, T, R, Y), one can define the *query word co-occurrence graph* as a weighted, undirected graph whose set of vertices is the set T of query words. For all users $u \in U$ and resources $r \in R$, let T_{ur} be the set of query words within one query, i. e., $T_{ur} := \{t \in T \mid (u, t, r) \in Y\}$. Two query words t_1 and t_2 are connected by an edge, iff there is at least one query (u, T_{ur}, r) with $t_1, t_2 \in T_{ur}$. The *weight* of this edge is given by the number of queries that contain both t_1 and t_2, i. e.,

$$w(t_1, t_2) := \mathrm{card}\{(u, r) \in U \times R \mid t_1, t_2 \in T_{ur}\} \ . \tag{6.3}$$

Co-occurrence relatedness (Co-Occ) between query words is given directly by the edge weights. For a given query word $t \in T$, the tags that are most related to it are thus all the tags $t' \in T$ with $t' \neq t$ such that $w(t, t')$ is the maximum value.
Three distributional measures of query-word-relatedness that are based on three different vector space representations of query words have been introduced in Benz et al. [2009b]. The difference between the representations is the feature space used to describe the tags, which varies over the three dimensions of the logsonomy.
Specifically, for $X \in \{U, T, R\}$ we consider the vector space $\mathbb{R}^X$, where each query word t is represented by a vector $\mathbf{v}_t \in \mathbb{R}^X$ as described below.
The *Tag Context Similarity* (TagCont) is computed in the vector space $\mathbb{R}^T$, where, for tag t, the entries of the vector $\mathbf{v}_t \in \mathbb{R}^T$ are defined by $v_{tt'} := w(t, t')$ for $t \neq t' \in T$, where w is the co-occurrence weight defined above and $v_{tt} = 0$. The reason for giving a zero weight between a node and itself is that two tags should be considered related when they occur in a similar context and not when they occur together.
The *Resource Context Similarity* (ResCont) is computed in the vector space $\mathbb{R}^R$. For a tag t, the vector $\mathbf{v}_t \in \mathbb{R}^R$ is constructed by counting how often a tag t is used to annotate a certain resource $r \in R$: $v_{tr} := \mathrm{card}\{u \in U \mid (u, t, r) \in Y\}$.
The *User Context Similarity* (UserCont) is built similarly to ResCont by swapping the roles of the sets R and U: For a tag t, the vector $\mathbf{v}_t \in \mathbb{R}^U$ is defined as $v_{tu} := \mathrm{card}\{r \in R \mid (u, t, r) \in Y\}$.
In all three representations, we measure vector similarity by using the cosine measure [Singhal, 2001], as is customary in Information Retrieval. The *FolkRank algorithm* (see Chapter 2.5.1) allows for computing a ranking of the most similar tags to a specific term t.
To compute a tag ranking in the logsonomy, we assigned high weights to a specific query term t in the random surfer vector. The final outcome of the FolkRank is then, among others, the ranked list of tags which FolkRank judges as related to t.

First Insights

Table 6.12 provides a few examples of the related tags returned by the measures under study. A first observation is that the co-occurrence relatedness seems to often "restore" compound expressions like *news channel*, *guitar tabs*, *brain tumor*. This can be attributed

Table 6.12: Examples of most related tags for each of the presented measures.

rank	tag	measure	1	2	3	4	5
37	news	*co-occurrence*	channel	daily	fox	paper	newport
		folkrank	channel	fox	daily	newspaper	county
		tag context	news.com	newspaper	weather	obituaries	newspapers
		resource context	news.com	arrested	killed	accident	local
		user context	county	center	edging	state	city
399	guitar	*co-occurrence*	tabs	chords	tab	free	bass
		folkrank	tabs	chords	lyrics	tab	music
		tag context	banjo	drum	piano	acoustic	bass
		resource context	tabs	tab	tablature	chords	acoustic
		user context	chords	tabs	tab	guitars	chord
474	gun	*co-occurrence*	smoking	paintball	parts	laws	control
		folkrank	guns	rifle	paintball	parts	sale
		tag context	guns	pistol	rifles	rifle	handgun
		resource context	smoking	pistol	rifle	handgun	guns
		user context	safes	guns	pistol	holsters	pellet
910	brain	*co-occurrence*	tumor	stem	injury	symptoms	tumors
		folkrank	cancer	symptoms	tumor	blood	disease
		tag context	pancreas	intestinal	liver	thyroid	lungs
		resource context	tumor	tumors	syndrome	damage	complications
		user context	stem	feline	tumor	acute	urinary

to the way the logsonomy was constructed, namely by splitting queries (and consequently also compound expressions) using whitespace as a delimiter. Another observation which is identical to the folksonomy data is that co-occurrence and FolkRank relatedness seem to often return the same related tags.

The tag context relatedness seems to yield substantially different tags. Our experience from folksonomy data (where this measure discovered preferentially synonym or sibling tags) seems to also prove true for logsonomy data: The most similar tags by tag context similarity often refer to a type of synonym[9] (e. g., *gun – guns*, *news – news.com*), whereas the remaining tags can be regarded as "siblings": For example, for the tag *brain* it gives other organs of the body, whereas for the tag *guitar* it gives other music instruments. When we talk about "siblings", we mean that these tags could be subsumed under a common parent in some suitable concept hierarchy; in this case, e. g., under *organs* and *music instruments*, respectively. In our folksonomy analysis, this effect was even stronger for the resource context relatedness – a finding which does not seem to hold for logsonomy data based on this first inspection. The resource context relatedness does exhibit some similarity to the tag context relatedness, however, in general it gives a mixed picture. User context relatedness is even more blurred – an observation is again in line with the folksonomy side. These first results suggest that despite the reported differences, especially the tag context in a logsonomy seems to hold a similar semantic information to the one we found in folksonomy data.

Semantic Grounding

Next, we look up the tags in an external, structured dictionary of word meanings. Within these structured knowledge representations, there are often well-defined metrics of seman-

[9] Please note that we do not use the term 'synonym' in a linguistically precise way; we regard two words as being synonyms when they basically refer to the same concept. This also includes e. g., singular / plural forms of a noun.

tic similarity. Based on these, one can infer which type of *semantic* relationship holds between the original and the related tags.

We use WordNet [Fellbaum, 1998b], a semantic lexicon of the English language. The core structure we exploit is its built-in taxonomy of words, grouped into synsets, which represent distinct concepts. Each synset consists of one or more words and is connected to other synsets via the *is-a* relation. The resulting directed acyclic graph connects *hyponyms* (more specific synsets) to *hypernyms* (more general synsets).

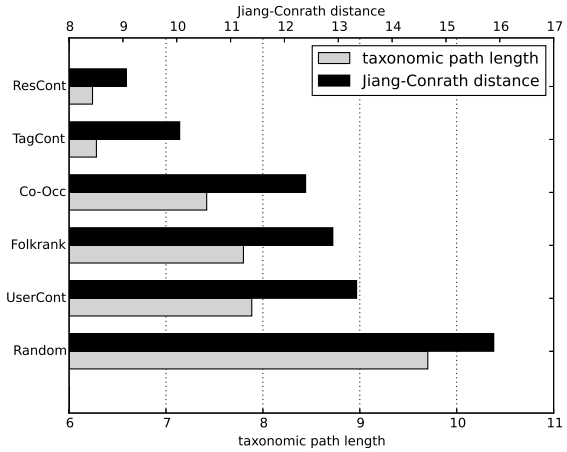
Based on this semantic graph structure, several metrics of semantic similarity have been proposed [Budanitsky and Hirst, 2006]. The most simple one is counting the number of nodes one has to traverse from one synset to another one. We adopted this *taxonomic shortest-path length* for our experiments. In addition, we use a measure of semantic distance introduced by Jiang and Conrath [1997], which combines the taxonomic path length with an information-theoretic similarity measure. The choice of this measure was guided by a work of Budanitsky and Hirst [2006], who showed by means of a user study that the Jiang-Conrath distance comes most closely to what humans perceive as semantically related.

Following the pattern proposed in Cattuto et al. [2008], we carried out a first assessment of our measures of relatedness by measuring – in WordNet – the average semantic distance between a tag and the corresponding most-closely-related-tag according to each of the relatedness measures under consideration. For each tag of our logsonomy, we find its most closely related tag using one of our measures. If we can map this pair to WordNet (i. e., if both tags are present), we measure the semantic distance between the two synsets containing these two tags. If any of the two tags occurs in more than one synset, we use the pair of synsets which minimizes the path length.

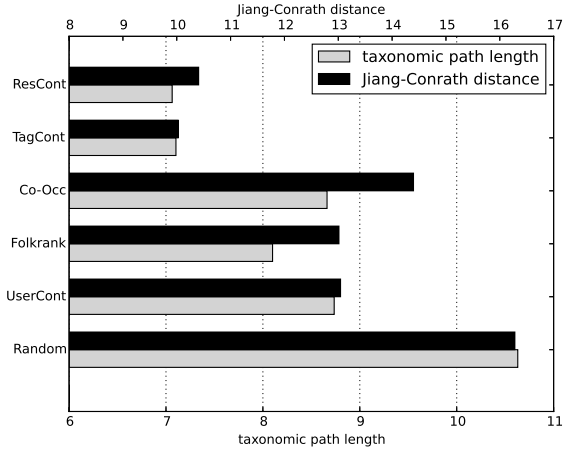
Figure 6.6 reports the average semantic distance between the original tag and the most related one, computed in WordNet by using both the taxonomic path length and the Jiang-Conrath distance. Overall, the diagrams are quite similar with respect to structure and scale. In both cases, the random relatedness (where we associated a given tag with a randomly chosen one) constitutes the worst case scenario.

Similar to our prior results for folksonomies (i. e., those shown in Figure 6.6(a)), for the logsonomy, the tag and resource context relatedness measures yield the semantically most closely related tags. In the logsonomy case, the distances between related tags for the context resource relatedness are larger than in the folksonomy case. We attribute this to the way the logsonomy is built: When users *implicitly* tag a certain URL by clicking on it, they are probably not as aware of the actual content of this page as a user who *explicitly* tags this URL in a social bookmarking system.

Another remarkable difference compared to the folksonomy data is that the co-occurrence relatedness yields tags whose meanings are comparatively distant from the one of the original tag. This can be attributed to the fact that co-occurrence often “reconstructs” compound expressions as already mentioned in Section 6.4.3. The finding is a natural consequence of splitting queries and consequently splitting compound expressions as we did. Our results confirm the intuitive assumption that the semantics of isolated parts of a compound expression usually are semantically complementary.



(a) Delicious folksonomy



(b) AOL logsonomy

Figure 6.6: Average semantic distance, measured in WordNet, from the original tag to the most closely related one. The distance is reported for each of the measures of tag similarity discussed in the main text (see 6.4.3). The corresponding labels are on the left. Grey bars (bottom) show the taxonomic path length in WordNet. Black bars (top) show the Jiang-Conrath measure of semantic distance.

6.4.4 Discussion

In this section we analysed the graph structure and semantic aspects of the logsonomies MSN and AOL. We found similar user, resource and tag distributions, whereby the split query datasets are closer to the original folksonomy than the complete query datasets. We could show that both graph structures have small world properties in that they exhibit relatively short shortest path length and high clustering coefficients. In general, the differences between the folksonomy and logsonomy model mentioned in Section 3.2.4 did not affect the graph structure of the logsonomies. Minor differences are triggered by the session IDs, which do not have the same thematic overlap as user IDs. Also, full queries show less inherent semantics than the split datasets do.

To analyse semantic aspects, we used different relatedness measures and WordNet. Due to the fact that queries were split up into single terms, we found that most co-occurrence related measures restored compound expressions. Interestingly, applying the resource-context-relatedness to logsonomies is much less precise for discovering semantically-close terms when compared to a folksonomy. We attribute this mainly to the incomplete user knowledge about the content of a page link they click on, leading e. g., to "erroneous" clicks. The behaviour of the tag context measure is more similar to the folksonomy case, which recommends it as a candidate for synonym and "sibling" term identication.
In future work, a more thourough analysis of these differences could lead to better understanding about why they occur. Would the inclusion of full URLs lead to different structural results? How can we avoid the splitting of compound expressions?
Overall, the results support our vision of merging the search engine and folksonomy worlds into one system. While some search engines already allow the storage of search results, they do not provide folksonomy-like navigation or the possibility to add or change tags. From a practical point of view, the following considerations are further arguments for a logsonomy implementation and its combination with a folksonomy system. Some of those points have recently been introduced into search engines.

- Users could enrich visited URLs with their own tags (besides the automatically added words from the query) and the search engine could use these tags to consider such URLs for later queries — also from other users. Thus, those tags could improve the general quality of the search engine's results.
- Search engines typically have the problem of finding new, unlinked web pages. Assuming, users store new pages in the folksonomy, the search engine could better direct its crawlers to new pages. Additionally, those URLs would have been already annotated by the user's tags. Therefore, even without crawling the pages it would be possible to present them in result sets.
- Folksonomies can help spot trends in society. Many social bookmarking users can be viewed as trend setters or early adopters of innovative ideas — their data is valuable for improving a search engine's diversity and novelty.
- Bookmarked URLs of the user may include pages the search engine can not reach (intranet, password-protected pages, etc.). These pages can then be integrated into personalized search results.

However, privacy issues are very important when talking about search engine logs. They provide details of a user's life and often allow the identification of the users themselves [Adar, 2007]. Certainly, this issue requires attention when implementing a logsonomy system.

6.5 Exploiting User Feedback

The last part of this chapter explores how the information collected in tagging systems can be of use for traditional search. One possibility is to use tagging data as a source of implicit feedback for learning-to-rank algorithms. Section 6.5.1 introduces this general idea and describes the different possibilities used in the experiments presented in Section 6.5.3 to

derive information from tagging data. Results are summarized in Section 6.5.4. The work has been presented in Navarro Bullock et al. [2011a].

6.5.1 Implicit Feedback from Tagging Data for Learning-to-Rank

In Section 3.2.3 the general concept of learning-to-rank was introduced. The idea is to learn a ranking function in order to sort search results. The training data consists of queries and documents matching the query ordered according to their relevance to the query. A query-document pair is usually represented by feature vectors with features such as the frequency of the query term in the document's summary or the length of the document.

Different approaches exist for solving the learning-to-rank task: pointwise, pairwise and listwise approaches [Dong et al., 2009]. In the experiments presented here we focus on the pairwise approach: A binary classifier is learned which classifies one of two documents to be more relevant than the other one (Section **??** presents details about the algorithm).

The training data consists of relevance scores assigned to query-document pairs. The scores for the training data are either derived by exploiting user search behaviour such as click data (for example in Dou et al. [2008]; Joachims [2002]; Macdonald and Ounis [2009]) or by asking experts to manually assess search results. The human evaluation of ranking results gives explicit relevance scores but is expensive to obtain. Clickdata can be logged from the user interaction with a search engine, but the feedback is noisy, as a click does not always indicate relevance. Sometimes, people are not satisfied with the clicked result, reformulate their information need or click on another resource.

The process of storing and annotating web links in a social bookmarking system can also be seen as an expression of relevance: by tagging a specific URL, a user judges this resource to be of importance. The resource's tags indicate what it is relevant for. Mostly, tags describe a topic, the resource's context or the user's reason for tagging the resource.

While search queries express a specific information need and there is no evidence as to whether a clicked URL fulfills this need or not, tags serve as a description or categorization for the specific resource. It would therefore be helpful for generating training- and test data for learning-to-rank, if, of course, one could use tagging data as a further source of implicit feedback.

In order to explore the practicability of this approach, we compare implicit feedback generated from tagging data to implicit feedback generated from clickdata. Given a search query and the ranking of a search engine, we match the query and URLs with tags and resources of a social bookmarking system. We thereby assume that a URL in ranking list is important if it has been tagged with the query terms (or similar tags). At the same time, we assume that the URL is relevant if it has been clicked on after the submission of the specific query.

In Section 6.5.2 we present different strategies for modeling implicit feedback. To compare the feedback type's performance, ranking models are learned using training data, where the relevance scores are generated from a specific feedback type (for example from tagging data). The models are tested by predicting relevance scores generated from other feedback types (for example click data). The experimental results, described in Section 6.5.3, show that ranking models generated from both tagging and click data are comparable in terms of the rankings they produce.

Table 6.13: Example of a mapping for the MSN search query *Social Web*, user clicks in the MSN ranking and resources tagged with *Social Web* in Delicious.

query: *Social Web*		
MSN click	Delicious resource	MSN ranking
x	x	http://www.socialweb.net/
		http://de.wikipedia.org/wiki/Web_2.0
x		http://www.socialweb.siteblob.com
	x	http://www.extensions.joomla.org/extensions/social-web

6.5.2 Mapping click and tagging data to rankings

Given a search query and the ranking of a search engine, one can match the query and ranked URLs with tags and resources from the social bookmarking system. Different possibilities exist to derive such a match: a link in the social bookmarking system can be seen as an indicator for relevance if it contains one of the query terms as tag, all query terms as tags or tags that are similar to the query terms. The experiments in this paper consider all social bookmarking links as relevant when they contain all of the tags. Table 6.13 depicts an example for such a match. The query submitted to the search engine is "Social Web". The first column shows which URLs in the ranking were clicked by MSN users. The second table shows which URLs in the ranking also appear in the folksonomy with "Social Web" as tags.

Preference Strategies

By mapping click- or tagging data to query rankings, we know which URLs of a ranking have been clicked on or tagged with the same tag as the query in the social bookmarking system. One can then extract preference pairs from the different lists l_q in the form of tuples $\{(q, r_i, r_j) \mid r_i \succ r_j\}$, which means that document r_i is more relevant than document r_j for the query q. Different strategies can be defined to extract the relevance pairs considering the order of resources, the number of times a resources was clicked on or the co-occurrence of resources. In the following paragraphs, strategy names with the suffix *tag* refer to a strategy based on tagging data, without the suffix to a strategy based on clickdata.

Binary relevance (binary, binarytag): A preference pair (q, r_i, r_j) is extracted if an arbitrary user clicked on or tagged a resource r_i while no user clicked on or tagged the resource r_j. The click pairs in Table 6.13 would then be $\{(r_1, r_2), (r_1, r_4), (r_3, r_2), (r_3, r_4)\}$.

Heuristic rules: In [Joachims, 2002] different heuristic rules were proposed to infer preference statements from clickdata (see Section 3.2.2).

- *Skip above pair extraction (sa, safull, satag, satagfull):* For the ordered ranking list and a URL at position i which was either tagged or clicked on, all unclicked/untagged results ranked above i are predicted to be less relevant than the result at position i. URLs after i are ignored (*sa*, *satag*) or also considered as non-relevant (*safull, satagfull*). Example pairs in Table 6.13 for *satag* are: $\{(r_4, r_2), (r_4, r_3)\}$, for *satagfull*: $\{(r_4, r_2), (r_4, r_3), (r_1, r_2), (r_1, r_3)\}$.

- *Skip above reverse pair extraction (sar, sarfull, sartag, sartagfull):* Search engine

users normally scan a result list from the top to the bottom. After clicking on one document, their information need is either satisfied or they get back to the list and continue scanning. Resources clicked at later positions have therefore been seen after the clicks at earlier positions. They can be considered as more likely to be relevant than the clicked documents before. Example pairs in Table 6.13 for *sar* are: $\{(r_3, r_1)\}$, for *sarfull*: $\{(r_3, r_2), (r_3, r_1)\}$. Example pairs for *sartag* are: $\{(r_4, r_1)\}$, for *sartagfull*: $\{(r_4, r_1), (r_4, r_2), (r_4, r_3)\}$.

Popularity information (popularity, poptag): When aggregating click-through or tagging data over different users, one can get information about the popularity of the resource. The more often a resource was tagged, the more relevant it might be for the tags/queries in question. Hence, we can extract a preference pair if r_i was more often clicked on or tagged than r_j by counting clicks over all sessions of the query log and all posts of the folksonomy. This strategy does not consider multiple clicks for the same query within one session as they are often caused by spammers. In folksonomies, this is not a problem, as users can tag resources only once.
FolkRank (folkrank): In Hotho et al. [2006c], the well-known PageRank algorithm was adapted to folksonomies (see Section 2.5.1). Resources which appear in the folksonomy ranking as well as the search engine's ranking list are preferred over those which were not ranked highly by the FolkRank algorithm.
Based on the described strategies, pairwise preferences can be extracted and ordered into (partial) ranking lists. For example, if three URLs of a ranked list have been clicked on, those URLs would be ordered according to their popularity by means of the *popularity* strategy, while all other URLs in the ranked list would be set to non-relevant (e. g., receive a rank score of 0).

6.5.3 Experimental Setup

This section describes the experiments' datasets and the general experimental setting.

Setting

Since no ground truth for the rankings exists, we compare the performance of different strategies against each other. The basic idea is to learn a ranking model using training examples with preference scores generated from one of the strategies (see Section 6.5.2) and thereby predict new rankings with this model. The predicted rankings can then be compared to preference scores derived from other strategies. If click and tagging data based strategies generate similar results, they can be considered both as valuable sources for implicit feedback. We construct training examples using the information (document title, summary, URL) given by the *MSN ranking* dataset (see Section 6.2.3). Features similar to those proposed in the LeToR 4.0 benchmark dataset [10] are computed to represent a query-document pair. The features include term frequency, the length, tf-idf values, BM25 and different language models of the document fields (body, anchor, title, URL or entire document). As the MSN dataset offers only summary snippets as descriptions of a website, we use those snippets as body text and skip all features based on anchor text.

[10] http://research.microsoft.com/en-us/um/beijing/projects/letor/LETOR4.0

Overall, each query-document pair consists of 46 features (see Appendix A.1 for a full list). The relevance scores of each query-document pair are inferred from one of the strategies proposed in 6.5.2. For example, the *binarytag* strategy would set all query-document pairs as relevant which have been tagged with the appropriate query terms.
In Dou et al. [2008], the click entropy was proposed to classify queries. Queries with lower click entropies are navigational queries (for example "yahoo" or "facebook"). As the result lists of those queries are normally less diverse and easier to predict we filter queries with high click entropies. The measure itself is defined as

$$Click - Entropy(q) = \sum_{d \in D(q)} -P(d|q) log_2 P(d|q) \tag{6.4}$$

where $Click - Entropy(q)$ is the click entropy of query q. In our settings, we have either clicks or posts including the tagged resources and the query as their tags. $D(q)$ is then the collection of documents clicked on or tagged for query q. $P(d|q)$ is the percentage of clicks for document d among all clicks for q or the percentage of posts with document d among all posts considering q as tags. In our experiments, we consider queries with a click or tag entropy lower than 0.5. Furthermore, query rankings with less than five clicks or five posts are not considered.
We use ranking SVM [Joachims, 2002] to learn the different models (see Section 3.2.3).

Comparison of different feedback methods

We first compare the similarity of ranking lists derived from the different strategies in Section 6.5.2 by means of a correlation analysis. In a second step, the performance of models derived from different preference scores are compared in Section 6.5.3.

Correlations

The similarity of ranking lists derived from the different strategies can be compared by analyzing their correlations. As a correlation measure, the Kendall tau-b (τ_b) is used, which measures the degree of correlation between rankings by considering the number of concordant and discordant pairs. In contrast to Kendall tau, the measure additionally considers ties [Dou et al., 2008]. A Kendall tau-b of 1 yields a perfect correlation, while a correlation of -1 reveals an inverse ranking.
Table 6.14 shows the correlations of each ranking list with all other ranking lists. Each ranking strategy is perfectly correlated with itself. Reverse strategies such as *sa* and *sar* are perfectly inversely correlated. The strategies *sarfull* and *safull* correlate as strongly as the full rankings. This is due to the fact that the two strategies also consider pairs derived from non-clicked documents and are therefore very similar. The feedback generated from *satas* correlates strongly with *rank* and *sa*, as no reordering takes place. The correlation of the *folkrank* and *poptag* strategies with feedback generated from clickdata (for example, *sa, sar*) is mostly positive but low. The *popularity* strategy yields the highest correlation (0.20 / 0.279). Similar feedback ranking lists seem to be generated from the *folkrank* and *poptag* strategies (0.88). Overall, one can find a positive correlation between feedback lists generated from click and tagging data. However, the correlation is not as high as feedback generated from strategies using the same feedback data type.

Table 6.14: Correlation of each ranking list with all other ranking lists derived from the different strategies presented in Section 6.5.2. Each ranking strategy is perfectly correlated with itself. Reverse strategies such as *sa* and *sar* are perfectly inversely correlated. There is a positive correlation between feedback lists generated of click and tagging data, but the correlation is low.

	rank	safull	sa	sar-full	sar	popu-larity	safull-tag	satas	sarfull-tag	sar-tag	poptag	folk-rank
rank	1.0	0.984	1.0	0.982	-1.0	0.263	0.971	1.0	0.965	-1.0	0.150	0.130
safull		1.0	1.0	0.998	-1.0	0.263	0.952	0.886	0.947	-0.886	0.182	0.140
sa			1.0	-1.0	-1.0	0.263	0.803	1.0	0.570	-1.0	0.226	0.158
sarfull				1.0	1.0	-0.263	0.949	0.794	0.944	-0.794	0.171	0.136
sar					1.0	-0.263	-0.803	-1.0	-0.570	1.0	-0.226	-0.158
popularity						1.0	0.285	0.483	0.191	-0.483	0.279	0.203
safulltag							1.0	1.0	0.994	-1.0	0.150	0.350
satag								1.0	-1.0	-1.0	0.150	0.134
sarfulltas									1.0	1.0	-0.150	0.301
sartag										1.0	-0.150	-0.138
popularitytag											1.0	0.880
folkrank												1.0

Table 6.15: Prediction errors made by models derived from training data using different strategies (rows) and tested on test sets of a specific strategy and a corresponding random test set. The error of models tested on those random test examples is close to 0.5. The error of models tested on examples with preference scores derived from the corresponding strategy is smaller.

training / test	binary	rand_binary	binarytag	rand_binarytag	folkrank	rand_folkrank
binary	**0.26**	0.48	0.37	0.49	0.33	0.49
binarytag	0.31	0.51	**0.33**	0.50	**0.31**	0.50
folkrank	**0.26**	0.50	**0.33**	0.51	0.34	0.51
popularity	**0.26**	0.50	0.37	0.50	0.33	0.51
popularitytag	0.31	0.48	**0.33**	0.50	**0.31**	0.49
rank	0.29	0.51	0.41	0.50	0.36	0.51
sa	**0.26**	0.49	0.37	0.49	0.33	0.49
safull	0.28	0.51	0.40	0.51	0.35	0.52
safulltag	0.30	0.48	0.36	0.49	0.32	0.49

Error

As a performance measure we consider the training error, which is defined by the number of missclassified pairs divided by the number of total pairs.

Table 6.15 depicts the errors made by models derived from training data using different strategies (rows) and tested on test sets of a specific strategy and a corresponding random test set. The random test sets contain the same number of preference pairs as the test set from a specific strategy, but preferences are uniformly sampled from all possible preference pairs. The error of models tested on those random test examples is close to 0.5, which is the probability that a random selected pair is swapped. The error of models tested on examples with preference scores derived from the corresponding strategy is smaller. For example, the model generated from the *folkrank* strategy (3rd row) has an error of 0.26 when tested on test preference pairs generated from the *binary* strategy, while the random test set (*rand_binary*) reveals an error of 0.5. The error difference between non-random and random models demonstrates that non-random models approximately predict document orders according to the relevance of the documents. The complete test set results (without random test sets) are shown in Figure 6.16.

Table 6.16 shows the prediction errors obtained over all different test sets (without random

Table 6.16: Prediction errors obtained over all different test sets (without random counterparts). Again, the rows indicate the strategy used for generating preference scores for the training examples, while the columns represent the strategies for the preference scores used for testing. The best-performing models for each column are marked in bold.

	binary	binarytag	folkrank	popularity	poptag	rank	sa	safull	safulltag
binary	**0.26**	0.37	0.33	**0.27**	0.37	0.43	**0.27**	0.42	0.45
binarytag	0.31	**0.33**	**0.31**	0.31	**0.33**	0.45	0.32	0.44	0.45
folkrank	**0.26**	**0.33**	0.34	**0.27**	0.34	**0.42**	**0.27**	**0.41**	**0.43**
popularity	**0.26**	0.37	0.33	**0.27**	0.37	0.43	**0.27**	0.42	0.45
popularitytag	0.31	**0.33**	**0.31**	0.31	0.34	0.45	0.31	0.44	0.45
rank	0.29	0.41	0.36	0.29	0.41	**0.42**	0.29	**0.41**	0.45
sa	**0.26**	0.37	0.33	**0.27**	0.38	0.43	**0.27**	0.42	0.45
safull	0.28	0.4	0.35	0.28	0.4	**0.42**	0.28	**0.41**	0.45
safulltag	0.3	0.36	0.32	0.3	0.36	0.43	0.3	0.42	**0.43**

counterparts). Again, the rows indicate the strategy used for generating preference scores for the training examples, while the columns represent the strategies for the preference scores used for testing. The best-performing models for each column are marked in bold.
Although the model learned from a strategy often performs best when tested against the rankings inferred from the specific strategy (for example *binary* or *binarytag*), this is not always the case (for example *poptag*). The *rank* strategy, derived from the original MSN ranking, performs worse than the other strategies in the majority of cases. Comparing strategies derived from the clickdata and those derived from the tagging dataset, one can find that clickdata-derived strategies perform better on the clickdata and tagging-data-derived strategies on tagging data. Only models derived from the *folkrank* strategy perform well on both kinds of data. As the *folkrank* strategy does not match query terms to tags, but rather computes a ranking using the entire folksonomy, the results can be seen as an indicator to test more elaborate matching approaches than matching folksonomy resources to rankings only when the query terms appear as tags. However, further experiments are required to better understand the results and to better reduce noise.

6.5.4 Discussion

The last section presented a comparison of implicit feedback strategies for a learning-to-rank scenario. Analogously to previous works proposing strategies for extracting preference scores from clickdata, we proposed different methods to infer feedback from tagging systems. By learning models with training examples from one of the strategies and predicting the outcome of other strategies, we could analyse similarities and differences between click- and tagging data. While the *folkrank* strategy predicts feedback from both types of data reasonably well, the other strategies perform better when predicting feedback of examples generated from their corresponding dataset.
In future work it would be interesting to develop more sophisticated strategies by considering the time of a post, the activity and specific interests of users as well as by matching not only posts containing the same tags as query terms, but also similiar tags. As our evaluation method only compares strategies but cannot show a preference for one of them, we need to create a ground truth dataset by manually labeling examples. Furthermore, transfer learning methods can be an interesting future research direction, as the transfer of a model

learned on a specific dataset to another dataset allows the evaluation of the strategies on existing datasets (such as the LeToR datasets).

6.6 Summary

This chapter presented experiments and analyses answering the thesis research question about how information retrieval in folksonomies compares to traditional search engines. Four different aspects were considered:

- **Usage behaviour and system content**: How do tags differ from query terms? Do users tag and search in the same way? Do users click on the same resources as they tag?
- **Structure & Semantics**: Is there a folksonomy like structure inherent in query log files? Can we detect similar semantic connections in logsonomies?
- **Integration**: How can folksonomies be of use for traditional search?

Concerning user behaviour, it could be shown that both tagging and query systems present a long tail of infrequent items which reduce overlap between both systems. Considering only frequent items shows a higher overlap. Major differences arise from different usage of tags and search: e. g., to the composition of multi-word lexems to single terms in Delicious and the use of (parts of) URLs as query terms in MSN (see Section 6.3.1). It could be shown, however, that queries and tags are correlated over time (see Section 6.3.1), suggesting that ongoing, relevant events and topics are considered in both systems. As overlaps between the search and folksonomy system were rather high when compared to the size of the Internet and the limited dataset we used, it seems likely that users of social bookmarking systems tag web pages ranked highly by traditional search engines. Ranking comparisons resulted in the observation that folksonomy rankings based on the FolkRank algorithm correlated best with Google rankings, especially for specific IT topics. This also indicates that taggers prefer to use search engines (and most of all Google) to find information, i. e., the prominent resources in both systems overlap.
In order to compare the structure and semantics of folksonomies and search engines, we transformed the click data file of a search engine (MSN) into a folksonomy-like structure, a logsonomy. Using short path lengths and clustering coefficients in order to compare small world properties, it could be demonstrated that logsonomies do present a folksonomy like structure. Differences consist in the notion of user: while folksonomies store bookmarks from registered users, logsonomies track the interests in form of SessionIDs, which are not as coherent (see Section 6.4.2).
In terms of emergent semantics as found in folksonomy systems [Cattuto et al., 2008], logsonomies show slightly different characteristics. As tags in a logsonomy consist of split query terms, co-occurrence measures reconstructed compound expressions. While tag context measures show similar results (the detection of synonyms and siblings), the results from resource context relatedness return less precise semantically related terms which can be explained by the different process of search: users do not know in advance whether the retrieved page they click on really reflects the search need.
One possibility of integrating the user knowledge inherent in folksonomies into search is to use folksonomy data to derive implicit feedback and use this to improve rankings. A

comparison of different strategies to infer implicit feedback for a learning-to-rank scenario has shown that strategies tend to perform better when the same data (either tagging or click data) is used to generate feedback and to predict feedback. The best results when mixing tagging and click data for learning and evaluation are obtained from the strategy based on the *FolkRank* algorithm (see Section 6.5).

The analysis of this chapter contributes to an understanding of differences and similarities in the usage and structure of folksonomy and search engine systems. The observed similarities in click and tagging behaviour suggest a combination of both systems could be used to enhance a user's search experience. Search engine companies have started to follow the trend of integrating users into the search process. The search engine Google, for example, released its own social platform where users can register and connect to friends and other associates. The ranking results of a specific search in this system also include content published or liked by a user's friends [Heymans, 2009]. Social bookmarking systems, on the other hand, profit from the technologies and methods of search algorithms (for example the FolkRank algorithm).

Chapter 7

Spam Detection in Social Tagging Systems

7.1 Introduction

This chapter deals with the research question how to best detect and eliminate spam in social bookmarking systems (see 1.2.2). The success of social bookmarking systems and other tools of the Social Web depends on powerful spam fighting mechanisms. Without them, the systems would be invaded by malicious posts, and lose their legitimate users. In social bookmarking systems, manual spam fighting approaches such as the provision of captchas do not prevent human spammers from posting. Therefore, system providers need spam fighting methods which automatically identify malicious posts in the system.

As discussed in Section 4.2, the problem can be considered as a binary classification task. Based on different features which describe users and their posts, a model is built from training data to classify unknown examples (on a post or user level) either as "spam" or "non-spam" (ham). As we consider "social" systems in which legitimate users mainly interact with each other because they can benefit from other user's content and present themselves (or their interests / knowledge), an exclusion of non-spammers from publishing is a severe error which might prevent the user from further participation. Therefore, similar to other spam detection settings, the problem of generating too many false positive errors after classification needs to be carefully considered when implementing spam algorithms.

The adaptation of existing classification algorithms to the task of detecting spam in social bookmarking systems consists of two major steps. The first one is the selection of features to describe the users as accurately as possible. The second step is the selection of an appropriate classifier. In this chapter we will discuss both steps.

First, we will present different features which can be used for spam classification (7.3.1). These features are then evaluated with well-known classifiers (SVM, Naive Bayes, J48 and logistic regression). A deeper understanding of the features describing spam and non-spam groups is provided by an analysis of local patterns. Which feature values and their combinations describe typical spam patterns? The analysis of local patterns in the BibSonomy spam dataset is a collaborative work with the Data Mining and Information Retrieval Group of the University of Würzburg.

Algorithm adaptation and tuning will be briefly discussed by presenting the results of the ECML/PKDD discovery challenge 2008. In this event, a dataset of the social bookmarking

system BibSonomy was published and used by the challenge's participants to design and test spam classification algorithms. Several approaches concentrating on algorithm tuning as well as feature engineering competed to be the challenge's winner.
This chapter is organized as follows. In Section 7.2 the datasets used are introduced. Section 7.3 presents and evaluates possible features for spam detection in social bookmarking systems. Section 7.4.4 describes different spam classification approaches developed in the scope of the ECML/PKDD discovery challenge. Finally, spam features are considered in the light of local patterns. Section 7.6 summarizes the results.

7.2 Datasets

The datasets used for the spam experiments of this chapter have been created in the course of the years 2007 and 2008 from the social bookmarking system BibSonomy (see Section 2.6 for a description of the system). The process of how spammers were identified is briefly described in the next section. The resulting datasets are presented in Section 7.2.2.

7.2.1 Dataset Creation

In order to prevent spammers from publishing, the system administrators created a simple interface which allowed authorized users (mainly the system administrators and some researchers) to flag users as spammers. The interface will be presented in more detail in Section 9.4.
The flagging of spammers by different evaluators is a very subjective process. There were no official guidelines, just common sense as to what distinguishes users from spammers based on the content of their posts. To narrow down the set of potential spammers, the evaluators normally looked at a user's profile (e. g., name, e-mail address), the composition of posts (e. g., the semantics of tags, the number of tags) before assessing the content of the bookmarked websites. Borderline cases were handled from a practical point of view. BibSonomy intends to attract users from research, library and scholarly institutions. Therefore, entries referring to commercial advertisements, Google Ad clusters, or the introduction of specific companies are considered as spam. The marked spammers are shown on the administration interface and can be unflagged by all authorized users. Evaluators, however, rarely cross-checked the evaluations. A certain amount of noise in the classifications is therefore probable.
If users are flagged as spammers, their posts are no longer visible to the other users. As a consequence, general pages, frequently visited by BibSonomy users, such as the home page or a page showing popular entries are cleaned of malicious posts. Spammers, however, can still see and manage their own posts on their own user page, but are not able to use the API BibSonomy offers. New users having the same IP address as a spammer can not register.

7.2.2 Dataset Descriptions

Two datasets have been used for investigating spam in social bookmarking systems. The first one, created in 2007, is the predecessor of the second one, the official ECML/PKDD discovery challenge dataset. Both datasets are presented in the following section.

Table 7.1: Sizes of users, tags, resources and TAS of the SpamData07. The training data contains all TAS until end of November 2007, the test data contains all TAS of December 2007.

	$\|U\|$	$\|T\|$	$\|R\|$	$\|Y\|$
Overall	20,092	306,993	920,176	8,717,510
Train	17,202	282,473	774,678	7,904,735
Test	2,890	49,644	153,512	804,682

Table 7.2: User ratios of spam and non-spammers in SpamData07 training and test dataset. The non-spam ratio in the test set is slightly smaller than the non-spam ratio in the training dataset.

	Overall	Train	Test
Overall Users	20,092 (100.0 %)	17,202 (100.0 %)	2,890 (100.0 %)
Spam Users	18,681 (92.98 %)	15891 (92.38 %)	2790 (96,54 %)
Non - Spam Users	1,411 (7.02 %)	1311 (7,62 %)	100 (3,46 %)

Spam Dataset 2007 (SpamData07)

The Spam Dataset 2007, in the following referred to as SpamData07 is comprised of users, tags, resources and the user profile information of all BibSonomy users until the end of 2007. The different sizes of users, tags, resources of spammers and non-spammers are shown in Table 7.1. As at that time nearly all spammers posted bookmarks, the dataset disregards the publication posts of users. Considering only bookmarks, the datasets consists of 1,411 legitimate users and 18,681 users who were flagged as spammers.
The above data was used to generate a training and test set. Thereby, instances were split chronologically so that a prediction of spam for the next month/week/day could be evaluated. The training set comprehends all instances until the end of November 2007, the test set all instances of the month December 2007 (see Table 7.2).

Spam Dataset 2008 (SpamData08)

The Spam Dataset 2008 (in the following named SpamData08) was created in the scope of the ECML/PKDD challenge 2008. The public dataset [1] has been used by the challenge participants and various other researchers to explore features and algorithms for detecting social bookmarking spam (see Section 7.4).
Analogously to SpamData07, it contains information about users, tags and resources. Overall, seven files list users as well as spam and non-spam entries of TAS, BIBTEX and bookmarks (see Table 7.3). The files represent tables of the BibSonomy database. They have been created with the `mysqldump` program which dumps tables into simple text files which can be easily imported again into a MySQL database by using the `LOAD DATA INFILE` command.

[1] `http://www.kde.cs.uni-kassel.de/ws/rsdc08/dataset.html`

Table 7.3: The tables of the SpamData08 training dataset. Each table is stored as a MySQL dump in a file which has the same name as the table.

table	description	#rows
`tas`	Tag Assignments: fact table; who attached which tag to which resource (non-spam)	816,197
`tas_spam`	Tag ASsignments: fact table; who attached which tag to which resource (spam)	13,258,759
`bookmark`	dimension table for bookmark data (non-spam)	176,147
`bookmark_spam`	dimension table for bookmark data (spam)	1,626,560
`bibtex`	dimension table for BIBTEX data (non-spam)	92,545
`bibtex_spam`	dimension table for BIBTEX data (spam)	245
`user`	mapping of non-spammer/spammer for each user	31,715

Table 7.4: The distribution of spam/non-spam posts and users among bookmark/BIBTEX posts in the SpamData08 training dataset. Note that for the users the sum of the bookmark and BIBTEX columns is not equal to the overall number of users, since users can have both types of resources.

	overall	bookmark	BIBTEX
#posts	1,895,497 (100.0 %)	1,802,707 (95.10 %)	92,790 (4.90 %)
#regular posts	268,692 (14.18 %)	176,147 (9.29 %)	92,545 (4.88 %)
#spam posts	1,626,805 (85.82 %)	1,626,560 (85.81 %)	245 (0.01 %)
#users	31,715 (100.0 %)	31,033 (97.85 %)	1,329 (4.19 %)
#regular users	2,467 (7.78 %)	1,811 (5.71 %)	1,211 (3.82 %)
#spam users	29,248 (92.22 %)	29,222 (92.14 %)	118 (0.37 %)

In contrast to the SpamData07 dataset, information which might help to identify a user in the dataset is hidden. Usernames have been replaced by numbers and all user profile information (such as e-mail or full names) have been excluded.

The dump for the training dataset includes all posts from BibSonomy up to and including March 31st 2008, but excluding 1,017,162 posts from the user *dblp*[2] since this user is a representation of all publication metadata available from the DBLP computer science bibliography.[3]

Table 7.4 shows the number of posts in the dataset. By separating bookmark- and BIBTEX-posts and spam/non-spam posts, one can see that mainly the bookmark posts are affected by spam, where the majority of posts (more than 85 %) are spam.

The test data for the spam detection task consists of all posts which were stored between May 16th 2008 and June 30th 2008 (46 days).

Both datasets, SpamData07 and SpamData08, exhibit a highly skewed class distribution, i. e., there are many more spammers than non-spammers. This needs to be taken into account when selecting an appropriate evaluation method (see Section 7.3.2).

[2] `http://www.bibsonomy.org/user/dblp`
[3] `http://www.informatik.uni-trier.de/~ley/db/`

Table 7.5: The distribution of spam/non-spam posts and users among bookmark/BIBTEX posts in the SpamData08 test data.

	overall	bookmark	BIBTEX
#posts	207,012 (100.0 %)	141,173 (68.20 %)	65,839 (31.80 %)
#regular posts	67,191 (32.46 %)	1,399 (0.68 %)	65,792 (31.78 %)
#spam posts	139,821 (67.54 %)	139,774 (67.52 %)	47 (0.02 %)
#users	7,205 (100.0 %)	7,124 (98.88 %)	135 (1.87 %)
#regular users	171 (2.37 %)	102 (1.42 %)	99 (1.37 %)
#spam users	7,034 (97.63 %)	7,022 (97.46 %)	36 (0.50 %)

7.3 Feature Engineering

In this section we describe our experiments to identify and evaluate appropriate spam features for spam classification. The work was published in Krause et al. [2008c] and conducted on the dataset SpamData07.

7.3.1 Feature Description

An automatic classification of spammers requires features describing the user, so that legitimate users can be distinguished from malicious ones. In this section we describe the features we have chosen in detail. Overall we considered 25 features which can be classified into four different feature categories. Tables 7.6–7.9 summarize all features.

Profile features comprehend all information in the user's profile.

Location based features refer to the location a user publishes bookmarks from, or which is given as the domain in his or her e-mail address.

Activity based features concern the interaction of the user with the system.

Semantic features consider characteristics hidden in the choice and usage of tags.

A user instance in the training or test set consists of a vector where each entry corresponds to a feature value. Each feature is normalized over the total set of users by dividing a user's feature value minus the minimum value by the difference of the maximum and the minimum value of this specific feature.

Profile features

The profile features are extracted from a user's data, which is revealed when they request an account in BibSonomy. Table 7.6 shows the features corresponding to a user's profile. Most of the fields to fill in at registration are not obligatory, however, users need to indicate a name and a valid e-mail address. Spammers often differentiate from normal users in that they use names or e-mail addresses with many numbers. For instance, typical spam names could be "styris888" or "painrelief2".
Figure 7.1 shows the histogram of the spam/non-spam distribution and the number of digits in the username and the e-mail address (*namedigit*, *maildigit*). As can be seen, besides the peak at 0 numbers, spammers show a further peak at the two-digit category. The *namelen*,

Table 7.6: Description of the profile features

Feature name	Description
namedigit	name contains digits
namelen	length of name
maildigit	e-mail address contains digits
maillen	length of mail address
realnamelen	length of realname
realnamedigit	realname contains digits
realname2	two realnames
realname3	three realnames

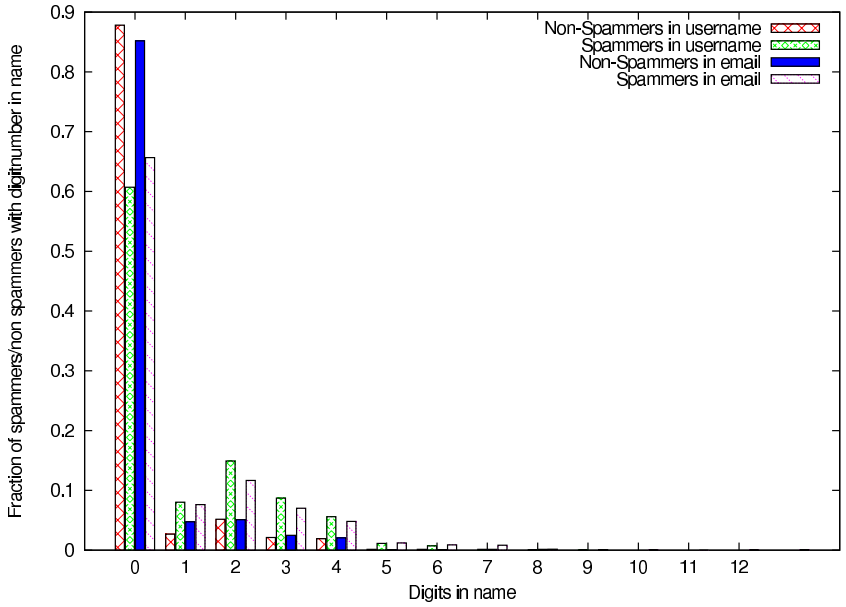


Figure 7.1: Histogram of the number of digits in the username and e-mail of spam vs. non-spam users. The x-axis shows the number of digits in either the username or the e-email address of spam and non-spammers. The y-axis shows the fraction spam vs. nonspam users containing the specific number of digits. Most users have no digits. Spammers, in general, use more digits than non-spammers.

maillen and *realnamelen* features refer to the length of the usernames, e-mail addresses and realnames. The *realname2* and *realname3* features are binary and set to one if the user has indicated two or three names. The features were derived from the observation that legitimate users often register with their full names.

Location based features

Location based features refer to those describing the user's location and domain. Table 7.7 summarizes the location based features.
Often, the same spammer uses several accounts to publish the same content. These ac-

Table 7.7: Description of the location based features

Feature name	Description
domaincount	number of users in the same domain
tldcount	number of users in the same top level domain
spamip	number of spam user with this IP

counts show the same IP address when they are registered. Thus, if one user with a specific IP is already marked as a spammer, the probability that other users with the same IP are also spammers is higher (*spamip*). When considering the users in the training dataset, 6,637 of them have at least one IP address in common with a spammer. Out of these, 6,614 users are marked as spammers. The same phenomenon holds for users of specific domains (*domaincount, tldcount*). The probability that a user who is from a rare domain which hosts many spammers is also a spammer is higher than average (and vice versa). For instance, 16 users have registered with the domain "spambob.com" and 137 with the domain "rhinowebmail", all of which were classified as spammers.

Activity based features

Activity based properties (Table 7.8) consider different kinds of user interactions with the social bookmarking system. While normal users tend to interact with the system directly after their registration (e. g., by posting a bookmark), spam users often wait a certain time after they submit their first post. This time lag can be considered when characterizing spam (*datediff*).

Furthermore, the number of tags per post varies (*tasperpost*). Spammers often add many different tags to a resource, be it to show up more often in searches or to confuse spam detection mechanisms by including "good" tags. Considering the BibSonomy dataset, spammers add on average eight tags to a post while non-spammers add four. The average number of TAS (see Definition 2.2.1) is 470 for spammers and 334 for users (*tascount*).

Table 7.8: Description of the activity based features

Feature name	Description
datediff	difference between registration and first post
tasperpost	number of tags per post
tascount	number of total tags added to all posts of this account

Table 7.9: Description of the semantic features

Feature name	Description
co(no)spamr	user co-occurrences (related to resources) with (non) spammers
co(no)spamt	user co-occurrences (related to tags) with (non) spammers
co(no)spamtr	user co-occurrences (related to tag-resources pairs) with (non) spammers
spamratio(r/t/rt)	ratios of spam/non spam co-occurrences
grouptag	number of times 'group=public' was used
spamtag	ratio of spam tags to all tags of a user

Semantic features

Semantic features (Table 7.9) relate to the usage and content of the tags which serve as an annotation for a bookmark.
There are several "simple" properties which were found when manually cleaning the system from spam. For instance, 1,916 users added "$group=public" as a tag or part of a tag for a resource. Out of these 1,914 users are spammers (*grouptag*). This specific tag is used by a software to generate spam in social bookmarking systems. We also have a blacklist of tags which contains keywords that are very likely to describe a spam post. For instance, "pornostars", "jewelry" or "gifts" are contained in this list. One feature calculates the ratio of such spam tags to all tags published by a specific user (*spamtag*).
Another set of features are based on co-occurrence information considering the sharing of tags and resources of users. Such information can be extracted by building different weighted undirected graphs whose set of vertices is the set of users U and two users (u_1 and u_2) are connected by an edge, if

1. they share at least one tag - resource combination, i. e., there are at least two tag assignments $(U_{rt}, t, r) \in Y$ with $u_1, u_2 \in U_{rt}$ (cospamtr / conospamtr). The edge weights are then computed as follows:

$$w(u_1, u_2) := |\{(r, t) \in R \times T \mid u_1, u_2 \in U_{tr}\}|. \tag{7.1}$$

2. they share at least one tag, i. e., there are at least two tuples $(U_t, t, r) \in Y$ with $u_1, u_2 \in U_t$ (cospamt / conospamt). The edge weights are then computed as follows:

$$w(u_1, u_2) := |\{(t) \in T \mid u_1, u_2 \in U_t\}|. \tag{7.2}$$

3. they share at least one resource, i. e., $(U_r, t, r) \in Y$ with $u_1, u_2 \in U_r$ (cospamr / conospamr). The edge weights are then computed as follows:

$$w(u_1, u_2) := |\{(r) \in R \mid u_1, u_2 \in U_r\}|. \tag{7.3}$$

For our feature calculation, we considered each graph (resource, tag or tag-resource co-occurrence graphs) twice: In the spam case, a link between u_1 and u_2 is only set if u_2 is a spammer, in the second a link is set if u_2 has been marked as a non-spammer.

The final measure, called *co(no)spam(r/t/tr)*, for a user u_i is then computed as the sum of the weight edges of all (n) users u_j connected to user i. For instance, $cospamr(u_i)$ is then defined as:

$$cospamr(u_i) = \sum_{j=1}^{n} w(u_i, u_j) \tag{7.4}$$

The assumption is that spammers show high values in the spammer co-occurrence graphs, as they use the same vocabulary and resources other spammers submit; non-spammers show higher values in the non-spammer case.
We also computed the ratio of each spam and non-spam pair (*spamratiot, spamratior, spamratiotr*).

$$\begin{aligned} spamratior(u_i) &= \frac{cospamr(u_i)}{cospamr(u_i) + conospamr(u_i)} \\ spamratiot(u_i) &= \frac{cospamt(u_i)}{cospamt(u_i) + conospamt(u_i)} \\ spamratiotr(u_i) &= \frac{cospamtr(u_i)}{cospamtr(u_i) + conospamtr(u_i)} \end{aligned} \tag{7.5}$$

7.3.2 Experimental Setup

For our evaluation, we consider the F-measure and the area under a ROC curve (AUC). Both are defined in Section 4.2.3. An advantage of using ROC curves for evaluation is that these curves are independent of the underlying class distribution. Therefore, the skewed class distribution in our dataset is not considered. Another more practical reason for considering the ROC curve is that we want to turn the obvious decisions over to the classifier while controlling the suggested classification of the borderline cases before finalizing the decision. The firm cases (the classifier's decision) are those at the beginning of the ROC curve. The steeper the curve starts, the fewer miss-classifications occur. Once the curve becomes flatter, we have to control the outcome of the classifier.
The simplest baseline we can consider is to always predict the majority class in the data, in our case "spammer". In our skewed dataset, this would yield a precision of 0,965, and a F-measure of 0,982 (for the spam class). However, all non-spammers would also be classified as spammers.
A more substantial baseline is to consider the tags used to describe a resource as features and use a classifier that has been shown to deliver good results for text classification, such as Naive Bayes. Each user u can then be represented as a vector $\mathbf{u}$ where each dimension corresponds to a unique tag t. Each component of $\mathbf{u}$ is then assigned a weight. We consider two different settings. In the first case, the weight corresponds to the absolute frequency with which the tag t occurs for the user u. In the second case, each tag is assigned a tf-idf value. The tf-idf value for a specific tag t_i considering all posts $\mathbb{P}_u$ of user u is defined as

$$tf - idf(i, \mathbb{P}_u) = \frac{tf_{i\mathbb{P}_u}}{max\{tf_{j\mathbb{P}_u}\}} log\frac{|P|}{|P_i|} \tag{7.6}$$

where $tf_{i\mathbb{P}_u}$ denotes the tag frequency of the tag $t_{i\mathbb{P}_u}$ in all tag assignments of the personomy P_u, $max\{tf_{j\mathbb{P}_u}\}$ is the frequency of the most frequent tag t_j in this personomy, $|P|$ is the total number of posts, and $|P_i|$ the number of posts which contain the tag t_i.

Table 7.10: Confusion matrix of the baseline with all tags as features (frequency). The ROC area value for the frequency baseline is 0.801 and the F-measure is 0.286.

	Spam	Non-Spam
Spam	466	2324
Non-Spam	0	100

Table 7.11: Confusion matrix of the baseline with all tags as features (tf-idf). The misclassification of spammers slightly improves, so that more spammers are identified. The ROC area value is 0.794 while the F-measure is 0.319.

	Spam	Non-Spam
Spam	530	2260
Non-Spam	0	99

Tables 7.10 and 7.11 show the TP, FP, FN, TN values for the absolute frequencies and the tf-idf values.
When computing the baseline with the tf-idf measure, the misclassification of spammers slightly improves, so that more spammers are identified. The ROC area value for the frequency baseline is 0.801 and the F-measure is 0.286. For the tf-idf baseline, the ROC area value is 0.794 while the F-measure is 0.319. Figure 7.2 shows the ROC curve progression of the two baselines. The curves are similar at the beginning. The tf-idf-baseline curve shows a steeper progression, but is later exceeded by the frequency-baseline.

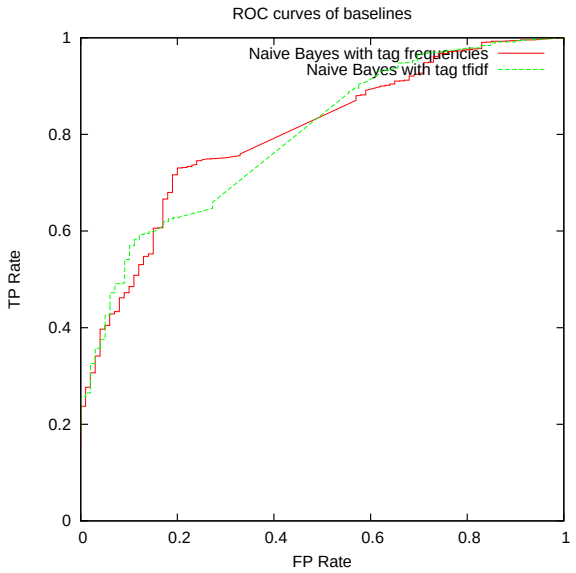


Figure 7.2: ROC curves of the frequency and tf-idf tag features. The x-axis shows the FP Rate while the y-axis shows the TP Rate. The tf-idf-baseline curve shows a steeper progression, but is later exceeded by the frequency-baseline.

Table 7.12: Evaluation values all features. The best classifier with an AUC of 0.936 is the SVM, followed by the logistic regression classifier. Even though the progression of the SVM's ROC shows that the false positive instances are the ones with less probability, 53 out of 100 non-spammers are ranked as spammers.

Classifier	ROC area	F1	FP	FN
Naive Bayes	0.906	0.876	14	603
SVM	0.936	0.986	53	23
Logistic Regression	0.918	0.968	30	144
J48	0.692	0.749	11	1112

7.3.3 Results

We selected different classification techniques to evaluate the features we introduced in the previous section. For the first three algorithms we used the Weka implementation [Hall et al., 2009] while for the SVM we used the LibSVM package [Chang and Lin, 2011].
Details about the used algorithms are presented in Section 4.2.2.
We tested different scenarios: Classification combining all features, classification of the feature groups and classification with costs. The results will be described in the following paragraphs together with the evaluation outcomes.

Classification combining all features

Table 7.12 shows the ROC area, F1 measure, and the absolute false positive values and false negative values for all algorithms based on all features. Figure 7.3 [4] depicts the ROC curves for all classifiers. The best classifier with an AUC of 0.936 is the SVM, followed by the logistic regression classifier. Even though the progression of the SVM's ROC shows that the false positive instances are the ones with less probability, 53 out of 100 non-spammers are ranked as spammers. Section 7.3.3 therefore introduces costs for misclassifying non-spammers. The AUCs of the two baselines (0.801 and 0.794) yield lower results.

Feature groups

In order to find out about the contribution of the different features, we have analysed each feature group separately. Thereby, the semantic features were split into two subgroups – co-occurrence features (and the ratios) and the spamtag/grouptag. Figures 7.4(a)–7.5(b) present the ROC curves and evaluation values for the different feature groups. The best results are given by the co-occurrence features (co(no)spamr, co(no)spamt, co(no)spamrt, spamratior, spamratiot, spamratiotr).
Table 7.13 shows, for each feature group, the evaluation values of the algorithm which optimizes the ROC area. Interestingly, there is not a single algorithm which performs best on all feature groups.

[4] We only included one baseline (tf-idf) to reduce the number of curves.

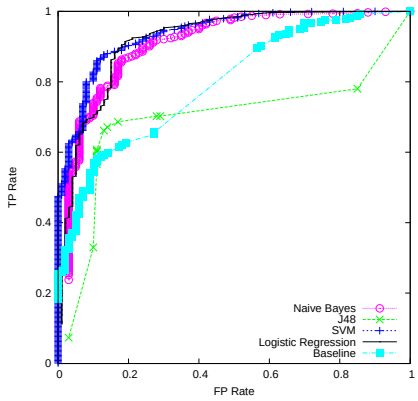


Figure 7.3: ROC curves of the different classifiers considering all features. The x-axis show the false positive rate and the y-axis the true positive rate. A random classifier would be a straight line from (0,0) to (1,1). The steepest progression shows the SVM classifier.

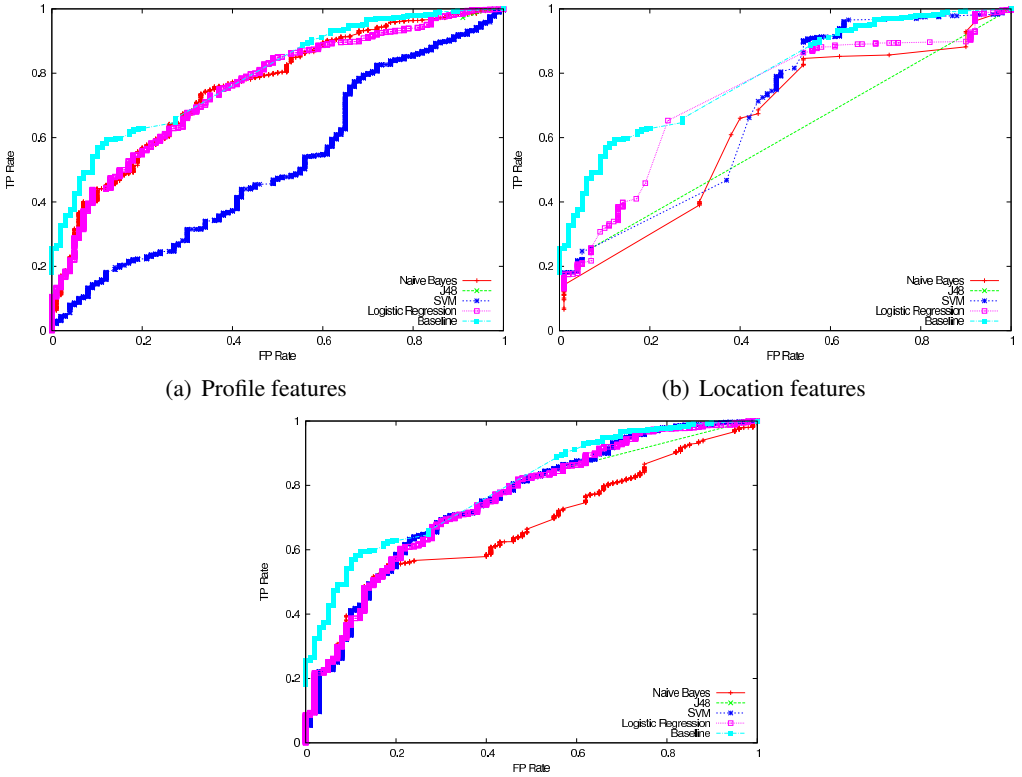
Table 7.13: Evaluation values of the feature groups computed by the algorithm which optimizes the ROC area. In respect to this measure, location features perform worst, while the best results are obtained from co-occurrence information.

Features	ROC area	F1
Profile features (log. reg.)	0.77	0.982
Location features (SVM)	0.698	0.407
Activity features (SVM)	0.752	0.982
Semantic features (J48)	0.815	0.981
Co-occurrence features (log. reg.)	0.927	0.985

Overall, none of the feature groups reaches the classification performance obtained when combining the features. This shows that in our setting, no dominant type of spam indicator exists. A variation of different kinds of information is helpful. The co-occurrence features describing the usage of a similar vocabulary and resources are most promising.

Costs

The ROC curves inherently introduce costs, as they order instances according to classification probabilities. However, most classifiers do not use cost information when building their models. As seen above, the SVM for the combination of all features nearly perfectly separates 40% of spammers from non-spammers. However, over half of the non-spammers are classified as spammers in the final result.
In order to penalize the wrong classification of non-spammers, we introduced cost sensitive learning [Hall et al., 2009]. Before a model is learned on the training data, the data is reweighted to increase the sensitivity to non-spam cases (i. e., the data consists of more non-spam classified instances than before). We experimented with different cost options and found that a penalty of ten times higher than the neutral value (one) delivered good results for the SVM. We also recalculated the other classifiers using cost options.



(a) Profile features

(b) Location features

(c) Activity features

Figure 7.4: ROC curves of the different feature groups (profile, location and activity). In each figure, the steepest curve is the baseline. No specific classifier performs better than the others in all scenarios.

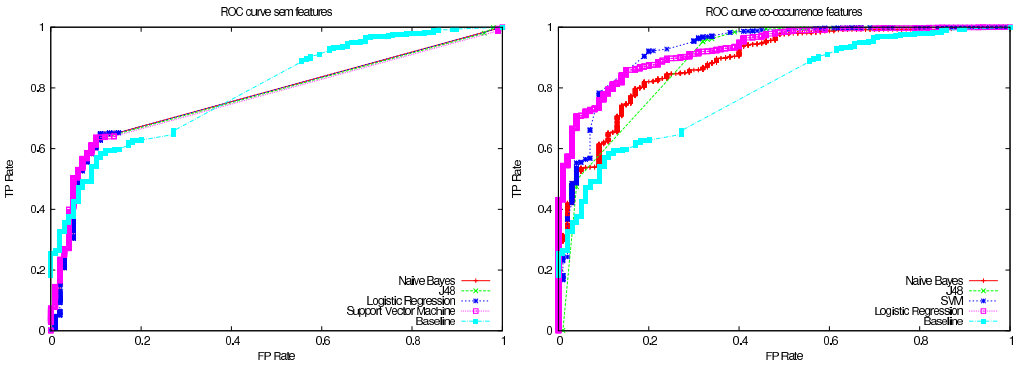
Table 7.14 shows an overview of the changed F1, false positive ratio and AUC values of classification using all features. As can be seen, cost-sensitive learning on all features with logistic regression returns the best results comparing the different classifiers. Except for the J48 classifier, the F1 degrades [5], while the false positive rates changes for the better. This shows that introducing costs help to reduce false positives at the expense of a worse performance in general.

7.3.4 Discussion

This section introduced a variety of features to fight spam in social bookmarking systems. The features were evaluated with well-known machine learning methods. Combining all features shows promising results exceeding the AUC and F1 measure of the selected baseline. Considering the different feature groups, co-occurrence features show the best ROC curves.

Our results support the claim of Heymann et al. [2007] that the problem can be solved with classical machine learning techniques — although not perfectly. The difference to web spam classification is the features applied: on the one hand, more information (e. g.,

[5] The same holds for the training dataset. For instance, the F1 measure for logistic regression is reduced from 0.991 to 0.041.



(a) Semantic features (grouptag, spamtag) (b) Semantic features (co-occurrences + ratios)

Figure 7.5: ROC curves of the two semantic feature groups. The best results yield the semantic features based on co-occurrence computations.

Table 7.14: Evaluation with a cost sensitive classifier using the test dataset. The values are compared to the F1, FP-rate and AUC values of the non cost-sensitive classifiers from Table 7.12. The cost-sensitive SVM, logistic regression and Naive Bayes classifiers show a reduced false positive rate, while their F1 measure deteriorates. The AUC value remains the same or increases slightly.

Classifier	SVM	J48	Logistic Regression	Naive Bayes
F1	0.924	0.794	0.927	0.855
F1 without costs	0.986	0.749	0.968	0.876
FP-rate	0.15	0.11	0.12	0.11
FP-rate without costs	0.53	0.11	0.30	0.14
ROC area	0.936	0.835	0.932	0.905
ROC area without costs	0.936	0.692	0.918	0.906

e-mail, tags) is given, on the other hand spammers reveal their identity by using a similar vocabulary and resources. This is why co-occurrence features tackle the problem very well.
Several issues considering our approach need to be discussed.

- A switch from the user level to the post level would be an interesting consideration. This would also facilitate the handling of borderline cases, as users, though some of their posts were flagged as spam, could still use the system.

- A consideration of a multiclass classification introducing classes in between “spam” and “non spam” or a ranking of classified instances may also help to identify those borderline cases a moderator needs to manually classify today.

- A further issue regards the evaluation method chosen. It would be interesting to consider more than one chronologically separated training/test set and to track the changes of the spam / non-spam ratio and the amount of user registrations over time.

- The large ratio between spam and non-spam users could be reduced by identifying spammers which have created several user accounts and therefore today are counted several times.

Overall, our contribution represents a first step towards the elimination of spam in social bookmarking systems using machine learning approaches. The spam detection framework presented in Chapter 9 uses many of the results of this section to classify spam in BibSonomy. Various papers of the ECML/PKDD 2008 discovery challenge presented in the following section build on the ideas of feature creation, algorithms and evaluation methods we introduced here and present new insights in one or more of these areas.

7.4 ECML/PKDD Discovery Challenge 2008

The ECML/PKDD discovery challenge 2008 [Hotho et al., 2008] offered participants two different competitions: spam detection or tag recommendation in social bookmarking systems. Both tasks asked for algorithms which had to be trained and tested on a dataset of BibSonomy. The presentation of the results took place at the ECML/PKDD discovery challenge workshop where the top teams were invited to present their approaches and results. The website with information about the dataset, the competition and results is still online [6].

In this section, we will present those results. Though the SpamData08 dataset consists of slightly different data (longer time period, no publication of registration information) than the dataset used for the experiments conducted in 7.3.3 (SpamData07), the results can be seen as comparable, as spam behaviour did not change during the period between them.

7.4.1 Task Description

The challenge's goal was to classify users in BibSonomy as spammers or non-spammers. In order to eliminate malicious users as early as possible, the model should be able to accurately distinguish spammers after the submission of only a few posts. The dataset, called *SpamData08*, is described in 7.2.2.

All participants could use the training dataset to build a model. The dataset contained flags identifying users as spammers or non-spammers. The test dataset with the same format could be downloaded two days before the end of the competition. Test users were those who registered and submitted posts in May 2008. All participants had to send a sorted file containing one line for each user composed of the user number and a confidence value. The higher the confidence value, the higher the probability that the user was a spammer. The highest confidence had to be listed first. An example of a result file is depicted in Table 7.15.

If no prediction was assigned to a user, it was assumed that the user was not a spammer. The evaluation criterion was the AUC (the Area under the ROC Curve) value (see 4.2.3). The submitted test user predictions of the participants were compared to the manual classifications of the BibSonomy administrators.

[6] `http://www.kde.cs.uni-kassel.de/ws/rsdc08/`

Table 7.15: Example of a result file for the ECML/PKDD spam challenge. Users are ordered according to their confidence values. The higher the confidence of the classifier, the more likely an instance can be categorized as spam and therefore the higher its position in the list.

user	spam
1234	1
1235	0.987
1236	0.765
1239	0
...	...

7.4.2 Methods

Thirteen solutions were submitted and evaluated by computing the AUC value. The proposed approaches of the challenge varied between those heavily reliant on feature engineering and approaches focusing on tuning machine learning methods (among them kNN, SVM and Neural Networks).

The winners, A. Gkanogiannis and T. Kalamboukis from Athens University, applied a linear classifier to classify users. The model was continuously refined by using a Rocchio-like relevance feedback technique. As classifier input, the authors unified all available information for a post, e. g., title, tags and description and considered it as a text document.

The second team, J. F. Chevalier and P. Gramme from Vadis Consulting designed a set of features characterizing tags and resources. Examples for automatically derived features include the number of tags or the number of bookmarks per user. Examples for manually derived features are the main tag category of a user or the total number of categories used.

C. Kim and K.-B. Hwang from Soongsil University ranked third by using a Naive Bayes classifier on a selected set of tags. The selection process was driven by mutual information and a restriction of tags to known tags from the test dataset.

Bogers and Bosch from Tilburg University, the Netherlands, assumed that similar system participants use a similar language which allows for a distinction of spammers and non-spammers. By means of language modeling, they identified the k most similar users or posts. Depending on the k nearest neighbour status (spam or non-spam) a new user received a score indicating the probability of being a spammer.

Krestel and Chen from the L3S Research Center at the University of Hannover focused on building co-occurrence networks of tags and resources. Kyriakopoulou and Kalamboukis from Athens University, Greece, used text clustering to compute a feature set which is used for text classification. Madkour et. al. from the IBM Cairo Technology Development Center investigated the applicability of semantic features similar to the ones introduced in 7.3. Neubauer and Obermayer from the Technical University of Berlin used co-occurrence, network and text features to set up an SVM model. All the approaches demonstrated the applicability of machine learning methods to solve the spam prediction task.

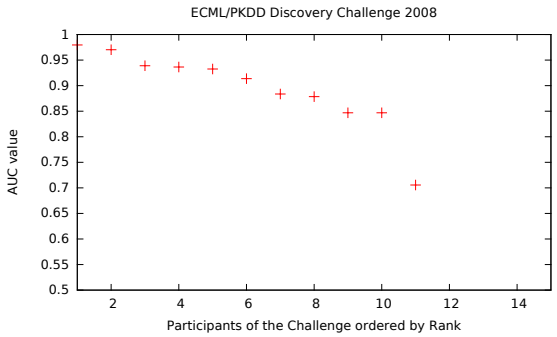
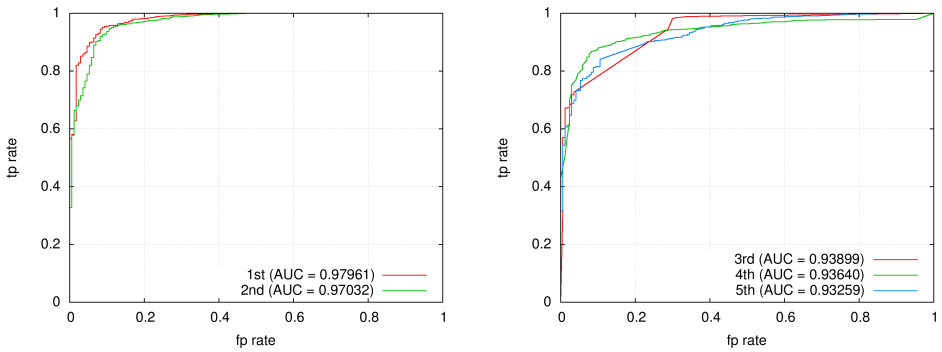


Figure 7.6: AUC values of different submissions. The x-axis shows the challenge's participants ordered by rank, the y-axis shows the AUC value. The scores range from 0.98 to 0.71.

7.4.3 Results

The AUC values of all participants are shown in Figure 7.6. As can be seen, most of the teams achieved an AUC score higher than 0.8. The best team, A. Gkanogiannis and T. Kalamboukis from Athens University, reached an AUC value of 0.98, followed by Gramme and Chevalier with 0.97 and Kim and Hwang with 0.94. Figures 7.7(a) and 7.7(b) depict the AUC curves of the five teams which ranked highest. As seen in Figure 7.7(a) the winning team has the steepest curve (at the beginning). As the ROC curve is plotted from a list ordered by confidence scores, one can see that their classifier accurately predicts spam instances. Only with instances having lower confidence scores does the classifier make mistakes. In contrast, the second and following teams have less steeper curves allowing for more misclassifications among their higher ranked instances.



(a) ROC curves of the first and second best team

(b) ROC curves of the 3rd-5th team

7.4.4 Discussion

In order to tackle the classification problem algorithmically, well established approaches such as SVMs or language modelling have been selected and tuned to solve the problem. It seems that the key for obtaining good classification results is the design of appropriate features. As the challenge's winning team focused on text classification, we can conclude

that interpreting a post as a text document and classifying such text snippets leads to reliable results. Other features such as co-occurrence features performed slightly worse, but are still helpful for classification.
In the experiments of Section 7.3.3, the best AUC value achieved with such an approach was 0.936, which would have been a fourth place in the challenge. Though the dataset is a slightly different one considering the time periods for training and testing, the results are still comparable. Personal information as used in this setting, such as the e-mail addresses or the IP address, do not necessarily lead to a better classification. This conclusion is affirmed in Chapter 8, where we explore different features and their suitability for preserving a user's privacy during the classification task.

7.5 Frequent Patterns

In the previous sections we have seen how important a good characterization of a legitimate social bookmarking system user is. Such characterizations can be used by a classification algorithm to derive appropriate features to distinguish between non-spammers and spammers. Another way to get a better understanding of users is the application of techniques from descriptive data mining, such as local pattern discovery.
Descriptive data mining – in contrast to predictive data mining – aims at finding "interesting, understandable and interpretable knowledge in order to discover hidden dependencies and characteristics of the data" [Atzmüller, 2007]. The idea of *concept description* is to better describe a specific population by finding descriptive patterns (also referred to as subgroups) within that population.
One can distinguish between two tasks in concept description: *Concept characterization*, which summarizes a given target population in terms of typical or characteristic features, and *concept/class discrimination*, which generates descriptions comparing the target population to one or more contrasting populations. In this way, both techniques describe the target population in complementary ways: Concept discrimination focuses on the differences between classes, while concept characterization focuses on the common or typical features of a certain class.
The two tasks can be realised by using techniques from *subgroup discovery*. Its goal is to identify relations between a dependent (target) variable, which represents the overall characteristic of a population and usually many independent variables describing the target appropriately. The relations are expressed by patterns describing a subgroup of the target variable in question. Instances whose feature values conform to the pattern are part of the subgroup.
Examples of such patterns – in the light of spam detection – could be the detection of the variable combination " users whose first post after registration yields a longer time span and who have no middle name indicate spammers [target variable]" or accordingly "users with a low number of tags and an IP in range X are usually non-spammers". Such revelations could be further used for spam classification or help to classify borderline cases manually.
The algorithms of subgroup discovery search for patterns in a given population (i. e., social bookmarking users) by optimizing a user-definable quality function. In the following section, we will discuss quality functions which can be applied to the spam detection task. The study results from a cooperation with Andreas Hotho, Florian Lemmerich and Martin

Atzmüller from the University of Würzburg. While they focused on the definition of the quality functions and conducted the experiments, the thesis' author prepared the dataset and helped to interpret the results. The next paragraphs will briefly introduce the quality function and present the experiment's results. They contain excerpts previously published in Atzmueller et al. [2009], however, they have been edited to fit this text.

7.5.1 Quality Functions for Discovering Frequent Spam Patterns

This section serves as background information in order to understand the experiments of Section 7.5.3. More details can be found in Atzmueller et al. [2009].
Frequent patterns of a dataset can be identified by combining different feature values (independent variables) and selecting those combinations which score highest with respect to the target variable and the applied quality function.
Let Ω_A be the set of all attributes. For each attribute $a \in \Omega_A$, a range $dom(a)$ of values is defined. Let V_A be the (universal) set of attribute values of the form $(a = v)$, where $a \in \Omega_A$ is an attribute and $v \in dom(a)$ is an assignable value. A subgroup description $sd = e_1, e_2, \ldots, e_n$ contains n individual selectors $e_i = (a_i, V_i)$, where $a_i \in \Omega_A$ is the attribute selected and $V_i \in dom(a_i)$ the attribute's value. Ω_E contains all possible selection expressions and Ω_{sd} the set of all possible subgroup descriptions.
A quality function with respect to a particular target variable $t \in \Omega_E$ can then be defined as $q : \Omega_{sd} \times \Omega_E \rightarrow R$. Subgroup descriptions $sd \in \Omega_{sd}$ can be sorted according to their quality scores: subgroups which receive low scores can be filtered by setting a threshold or by only accepting a certain amount (e. g., the top ten) of patterns. Thus, the definition of appropriate quality functions is essential in order to extract relevant patterns (i. e., variable combinations) for a target variable.
In literature, a variety of quality functions have been introduced to evaluate subgroup descriptions. Many of them consider the target share p, a subgroup's size n and the size of the total population. According to Atzmueller et al. [2009] the choice of an appropriate quality function, especially one applied to the discrimination setting, "is significantly dependent on the user requirements and the parameters of interest that are to be included into the quality function:" Since the previous experiments of this chapter already considered measures from information retrieval for evaluation purposes, (precision together with recall), a quality function based on these measures appeared to be a good choice.
In the scope of discovering spam with local patterns, the quality functions are based on the contingency table which has also been used for defining evaluation measures in Section 4.2:

> The category of true positives (tp) consists of all patterns which correctly predict the target variable, whereas the false positives (fp) represent all patterns for which the prediction is incorrect, and equivalently the false negatives (fn) and true negatives for the 'negation' of the rule, i. e., for the complement of the prediction [Atzmueller et al., 2009].

The first quality function defined is a pattern's *recall* which describes the relation of true positive users of a group to all existing positives, with $Pos = tp + fn$:

$$q_{TPR} = \frac{tp}{Pos} = \frac{tp}{tp + fn} \tag{7.7}$$

The quality function *recall* or equivalently the true positive rate (TPR) measures how many users of a specific group are actually covered by the pattern. It can therefore be used for concept characterization, i. e., for describing a subgroup (for example all spammers of a dataset) as clearly as possible: The better the recall, the more targeted users are contained in the subgroup. Subgroups having a large overlap with the target class instances are then selected; however, the (potentially large) overlap with non-target instances is not considered.
The second quality function defined is a pattern's *precision*, which measures the number of correctly extracted instances with respect to the size of the subgroup.

$$q_{PREC} = \frac{tp}{tp + fp} \tag{7.8}$$

In local pattern analysis, this function is order-equivalent to the relative gain quality function q_{RG} where $p = \frac{tp}{tp+fp}$ and p_0 is the relative frequency of the target variable of the total population [Atzmueller, 2007]:

$$q_{RG} = \frac{p - p_0}{p_0 \cdot (1 - p_0)}, n \geq \tau_{Cov} \tag{7.9}$$

The function does not account for a subgroup's size (n). Therefore, a minimum coverage threshold τ_{Cov} is introduced to make sure that a subgroup with a good q_{RG} covers a certain part of the target group. The function can be applied to the discriminative task which aims at distinguishing a given subgroup from the rest of the population. The higher the value of q_{RG}, the more precise the subgroup covers the target population.
The F-measure allows the combination of the pattern's *recall* and *precision*. It therefore enables the integration of concept characterization and discrimination. The F-measure in the scope of pattern analysis combines the two quality functions for characterization (q_c) and for discrimination (q_d). It is defined as follows:

$$F(q_c, q_d) = \frac{(1 + \beta^2) \times q_c \times q_d}{\beta^2 \times q_c + q_d} \tag{7.10}$$

Similarly to the F-Measure applied in IR experiments, "$F(q_c; q_d)$ measures the effectiveness of the concept description with respect to a user who attaches β times as much importance to q_c (characterization) as to q_d (discrimination)." [Atzmueller et al., 2009] With a $\beta = 1$ characterization and discrimination are equally weighted. The parameter provides "a convenient option for adaptations and for shifting the focus between characteristic and discriminative patterns." [Atzmueller et al., 2009]

7.5.2 Experimental Setup

For the case study we use the ECML/PKDD discovery challenge dataset *SpamData08* described in Section 7.2.2. The original data set contains 31715 cases (instances). After removing instances with missing values, the applied data set contained 31034 instances. The distribution of the classes in the applied dataset is highly unbalanced, with 1812 non-spammers and 29222 spammers, yielding default target shares of 5,8% non-spammers, and 94,2% spammers. In the following we will discuss both classes, i. e., spammers and non-spammers, as our target concepts using both characteristic and discriminative features/subgroups.

As attributes we selected 15 of the features already presented in Section 7.3.1. We thereby focused on socio-demographic features excluding semantic information such as co-occurrences. Table 7.16 summarizes the features used.

Feature Class	Feature Name	Description
Profile based	namedigit	name contains digits
	namelen	length of name
	maildigit	e-mail address contains digits
	maillen	length of mail address
	realnamelen	length of realname
	realnamedigit	realname contains digits
	realname2	two realnames
	realname3	three realnames
Location based	tld	top level domain
	domaincount	number of users in the same domain
	tldcount	number of users in the same top level domain
Activity based	datediff	difference between registration and first post
	grouppub	number of times '$group=public' was used
	tasperpost	number of tags per post
	tascount	number of total tags added to all posts of this account

Table 7.16: Summary of features describing spammers and non-spammers

For the spammer/non-spammer case study, we applied the q_{TPR} quality function measuring the true positive rate, or the recall of the patterns. For the discriminative setting we applied the relative gain quality function q_{RG} which is order equivalent to precision. Finally, for assessing the F-Measure, we utilized the classical recall and precision measures. We adjusted the measures slightly in order to control the simplicity of the patterns by favoring patterns with shorter descriptions (see Atzmueller et al. [2009] for more details).

7.5.3 Results

In this section we present the results using the above defined quality functions for concept characterization and discrimination of spammers and non-spammers.

Describing Non-Spammers

When comparing the attributes included in the patterns for concept characterization and discrimination of non-spammers, we find that *date_diff, grouppub, maildigit, maillen, namedigit, realname2, realname3, realnamelen, tld,* and *tldcount* are mainly used for characterization, while *date_diff, domaincount, maillen, namelen, realnamelen, tascount, tasperpost, tasperpost, tld*, and *tldcount* are mainly discriminative.
The used value ranges for the features are not always exclusive, which is explained by the general observation that characterizing features are also often observed for another class, while this is not true for the discriminative features. In general, profile information seems more important for characterization, while activity-based features seem more important for discrimination.
Figure 7.17 shows the results of applying concept characterization: The discovered subgroups are relatively large and (by construction) large areas of the target space are covered

Subgroup Description	Quality	Size	TP	Precision	Recall
grouppub=0	0.999	29095	1811	6.2%	99.9%
realname3=0	0.971	30712	1759	5.7%	97.1%
namedigit=0	0.855	19057	1550	8.1%	85.5%
maildigit=0	0.842	20956	1526	7.3%	84.2%
maillen=>17	0.754	26845	1366	5.1%	75.4%
realname2=0	0.611	15569	1107	7.1%	61.1%
grouppub=0 AND realname3=0	0.485	28792	1758	6.1%	97.0%
tld=com	0.462	24753	838	3.4%	46.3%
tldcount=>15092	0.462	24760	838	3.4%	46.3%
grouppub=0 AND namedigit=0	0.428	18044	1550	8.6%	85.5%
grouppub=0 AND maildigit=0	0.421	19708	1525	7.7%	84.2%
date_diff=>1104	0.417	20641	755	3.7%	41.7%
namedigit=0 AND realname3=0	0.415	18828	1504	8.0%	83.0%
realnamelen=0	0.408	8696	740	8.5%	40.8%
maildigit=0 AND realname3=0	0.407	20707	1476	7.1%	81.5%
realname2=>0	0.389	15465	705	4.6%	38.9%
maildigit=0 AND namedigit=0	0.386	16170	1398	8.7%	77.2%
grouppub=0 AND maillen=>17	0.377	25145	1365	5.4%	75.3%
maillen=>17 AND realname3=0	0.364	26569	1320	5.0%	72.9%
date_diff=8-1104	0.352	9486	638	6.7%	35.2%

Table 7.17: Concept Characterization of *non-spammers*. The table shows the top 20 subgroup descriptions for the target concept *class = non-spammer*; *Size* denotes the subgroup size, *Quality* is measured by the characteristic relative gain quality function q_{TPR}, *Precision* denotes the target share of the subgroup (precision), *TP* the number of *true positives* in the subgroup, and *Recall* the recall value of the subgroup pattern.

by the individual patterns. The most important attributes are *grouppub*, *realname3*, and *namedigit*, which characterize the non-spammer class with a value of zero very well. Especially the *namedigit* attribute makes intuitive sense: most spammers have numbers in their user names while non-spammers tend to select short user names without numbers.
Figure 7.18 shows the results of the discrimination task: As expected, the discriminative setting focuses on relatively small sections of the target space with high precision (target share), which is in contrast to the concept characterization results. The resulting patterns are significantly discriminative for the target class (non-spammer) and are readily available, e. g., for classification or explanation. One of the discriminative characteristics observable in the result list is the *number of tags per post*. The most discriminative values for this attribute are 2 and 3. This confirms the intuition that a non-spammer adds a smaller number of tags to a resource than a spammer. However, 4 and 5 is still a discriminative number and appears again in a few patterns. Another important attribute is *date_diff* with a value smaller than 7. Typically, non-spammers seem to register and submit their first post with only a small time difference, while spammers tend to register in BibSonomy and wait until they start to use the service they 'recently' discovered for their purposes. The *tld=de* pattern is also a discriminative feature. This can be explained by the fact that the system is very popular (for legitimate) users in Germany.
In general, while the concept characterization tasks produce descriptions which focus on demographic features such as the selection of the user name, a distinction between differ-

Subgroup Description	Quality	Size	TP	Precision	Recall
tldcount=61-70	12.58	40	30	75.0%	1.7%
tldcount=116-123	11.747	71	50	70.4%	2.8%
domaincount=89-90	9.13	116	65	56.0%	3.6%
tasperpost=4-5 AND tldcount=116-123	8.168	23	22	95.7%	1.2%
date_diff=0-7 AND tascount=0-1	8.044	35	33	94.3%	1.8%
tascount=2 AND tld=de	7.936	29	27	93.1%	1.5%
tldcount=1009-1312	7.874	916	450	49.1%	24.8%
namelen=0-3 AND tld=de	7.784	35	32	91.4%	1.8%
date_diff=0-7 AND domaincount=140-168	7.773	23	21	91.3%	1.2%
date_diff=0-7 AND tld=de	7.72	151	137	90.7%	7.6%
date_diff=0-7 AND namelen=0-3	7.589	28	25	89.3%	1.4%
date_diff=0-7	7.395	899	418	46.5%	23.1%
date_diff=0-7 AND domaincount=89-90	7.377	23	20	87.0%	1.1%
tasperpost=2 AND tldcount=1009-1312	7.303	101	87	86.1%	4.8%
tasperpost=0-1 AND tld=de	7.29	100	86	86.0%	4.8%
tascount=0-1 AND tld=de	7.264	42	36	85.7%	2.0%
namelen=0-3	7.239	149	68	45.6%	3.8%
date_diff=0-7 AND maillen=0-13	7.048	24	20	83.3%	1.1%
realnamelen=0 AND tldcount=116-123	7.048	24	20	83.3%	1.1%
date_diff=0-7 AND tascount=3-5	6.977	86	71	82.6%	3.9%

Table 7.18: Concept Discrimination of *non-spammers*. The table shows the 20 best subgroup descriptions for the target concept *class = non-spammer* using the discrimination setting. We applied the quality function q_{RG}, i. e., the relative gain quality function; for a description of the remaining parameters see Figure 7.17.

ent groups of non-spammers can be made with a combination of demographic and activity features. From this, we can learn that a good indicator for non-spammers is already given in the data they provide when registering; however, in order to reliably classify spammers we need further information about their system interaction.
Finally, Figure 7.19 shows the results of applying the F-Measure capturing both concept characterization and discrimination. Since the F-Measure combines both characterization and discrimination, the results show a balance between the other result tables: The focus of the patterns shifts towards more 'precise' but also more typical features. Considering the selected attributes, *date_diff* and *tldcount* appear most frequently. Considering the values of the attribute *tasperpost*, it is still important. However, it only comprises a smaller number of TAS (≤ 2), while the different subgroups implied by the condition TAS (> 2) seem to form a poor general description.

Describing Spammers

Considering the attributes used for concept discrimination, we see that specific values of *domaincount, grouppub, maildigit, namedigit, namelen, realname2, realnamelen, tasperpost, tldcount*, and certain top-level domains (*tld*) are very good indicators for spammers.

Subgroup Description	Quality	Size	TP	Precision	Recall
tldcount=1009-1312	0.33	916	450	49.1%	24.8%
date_diff=0-7	0.308	899	418	46.5%	23.1%
domaincount=0-3	0.215	3645	586	16.1%	32.3%
tasperpost=0-1	0.192	1588	326	20.5%	18.0%
tasperpost=2	0.17	2511	368	14.7%	20.3%
grouppub=0 AND tldcount=1009-1312	0.167	890	450	50.6%	24.8%
namedigit=0 AND tldcount=1009-1312	0.165	704	414	58.8%	22.9%
maildigit=0 AND tldcount=1009-1312	0.162	803	424	52.8%	23.4%
realname3=0 AND tldcount=1009-1312	0.161	901	436	48.4%	24.1%
tascount=3-5	0.156	3352	402	12.0%	22.2%
date_diff=0-7 AND grouppub=0	0.155	877	418	47.7%	23.1%
date_diff=0-7 AND namedigit=0	0.15	717	380	53.0%	21.0%
date_diff=0-7 AND realname3=0	0.15	882	404	45.8%	22.3%
namedigit=0	0.149	19057	1550	8.1%	85.5%
date_diff=0-7 AND maildigit=0	0.148	713	374	52.5%	20.6%
realnamelen=0	0.141	8696	740	8.5%	40.8%
maildigit=0	0.134	20956	1526	7.3%	84.2%
tascount=0-1	0.13	664	161	24.3%	8.9%
maillen=>17 AND tld=de	0.13	604	314	52.0%	17.3%
realname2=0	0.127	15569	1107	7.1%	61.1%

Table 7.19: Concept Description using the F-Measure. The table shows the top 20 subgroup descriptions for the target concept *class = non-spammer* (combined concept description setting).

For characterization, attributes like *date_diff, domaincount, grouppub, maildigit, maillen, namedigit, realnameX, tascount, tasperpost, tld* and *tldcount* are important, which is similar to the discriminative setting, but as expected, the value sets are often more general than the specific patterns used for discrimination.

Table 7.20 shows the results of the characterization of spammers[7]: While $grouppub \geq 0$ is a perfect feature for discrimination, $grouppub = 0$ is also a good feature for characterization, since there is also a large number of spammers with *grouppub=0*.

As expected, most spammers do not enter multiple names (*realname3=0*), however, they tend to choose long (*namelen*$\geq$ *9*) names, and long e-mail addresses (*maillen*$\geq$ *17*). Additionally, spammers often use digits in their names and e-mail (*namedigit*, *maildigit*). Our assumption is that they tend to number their created accounts at different sites. As a further characteristic, they often come from the .com domain, and use many TAS ($tascount \geq 33$) and TAS per post ($tasperpost = 5\text{-}11$).

Figure 7.21 shows the results of the discriminative description of spammers: While *date_diff* is not as important for discriminating spammers as discriminating non-spammers, the *tasperpost* attribute is also very important. As expected, and as also shown

[7]These results are also similar to the F-Measure results since spammers form the majority class and therefore the recall seems to dominate in this setting. Therefore we don't provide a detailed discussion of the F-Measure results.

Subgroup Description	Quality	Size	TP	Precision	Recall
realname3=0	0.991	30712	28953	94.3%	99.1%
grouppub=0	0.934	29095	27284	93.8%	93.4%
maillen=>17	0.872	26845	25479	94.9%	87.2%
tldcount=>15092	0.819	24760	23922	96.6%	81.9%
tld=com	0.818	24753	23915	96.6%	81.8%
date_diff=>1104	0.681	20641	19886	96.3%	68.1%
maildigit=0	0.665	20956	19430	92.7%	66.5%
namedigit=0	0.599	19057	17507	91.9%	59.9%
realname2=>0	0.505	15465	14760	95.4%	50.5%
realname2=0	0.495	15569	14462	92.9%	49.5%
tascount=>33	0.48	14447	14017	97.0%	48.0%
namelen=>9	0.466	14087	13621	96.7%	46.6%
grouppub=0 AND realname3=0	0.463	28792	27034	93.9%	92.5%
maillen=>17 AND realname3=0	0.432	26569	25249	95.0%	86.4%
grouppub=0 AND maillen=>17	0.407	25145	23780	94.6%	81.4%
realname3=0 AND tldcount=>15092	0.405	24507	23689	96.7%	81.1%
realname3=0 AND tld=com	0.405	24500	23682	96.7%	81.0%
namedigit=>0	0.401	11977	11715	97.8%	40.1%
tasperpost=5-11	0.378	11380	11055	97.1%	37.8%
domaincount=>4473	0.367	11200	10726	95.8%	36.7%

Table 7.20: Concept Characterization of *spammers*. The table shows 20 best subgroup descriptions for the target concept *class* = *spammer*. As for the non-spammer characterization, we applied the true positive rate q_{TPR} quality function. For a description of the parameters see Figure 7.17.

by the characterization findings, *realnamelen*, *maildigit* and *namedigit* provide typical features for spammers — usually having digits in their names and using longer names. Another very discriminative feature is *tld*. Spammers seem to heavily rely on domains such as *th*, *us*, *info*, and *biz* in addition to the already mentioned *com* domain. This complements the patterns observed for the non-spammers.

7.5.4 Discussion

The experiments above evaluated an approach for concept characterization and discrimination using local patterns that were discovered by applying subgroup discovery techniques. Suitable quality functions for the characterization and discrimination of spam and non-spam user groups could be found by relying on existing measures from the field of information retrieval.

The patterns provide interesting insights into the characteristics used to uncover spammers in social bookmarking. For example, it could be shown that the number of tags per post or the time lag between registration and the first post help to distinguish spammers from (non)spammers. The patterns make intuitive sense and help to better understand different user groups in the system. In future work, it would be of interest to explore how such patterns can be used to improve spam classification.

Subgroup Description	Quality	Size	TP	Precision	Recall
tld=th	1.062	29	29	100.0%	0.1%
grouppub=>0 1	.053	1939	1938	100.0%	6.6%
tld=us	1.011	708	706	99.7%	2.4%
tasperpost=>11	0.956	5515	5483	99.4%	18.8%
tldcount=666-1008	0.95	1464	1455	99.4%	5.0%
domaincount=91-139	0.894	651	645	99.1%	2.2%
tld=info	0.893	755	748	99.1%	2.6%
domaincount=2174-4473	0.716	6311	6191	98.1%	21.2%
tld=biz	0.716	105	103	98.1%	0.4%
namedigit=>0	0.664	11977	11715	97.8%	40.1%
domaincount=60-88	0.585	381	371	97.4%	1.3%
maildigit=>0	0.546	10078	9792	97.2%	33.5%
tasperpost=5-11	0.543	11380	11055	97.1%	37.8%
domaincount=169-2173 AND realnamelen=1-3	0.531	81	81	100.0%	0.3%
namelen=5 AND realnamelen=1-3	0.531	34	34	100.0%	0.1%
realname2=>0 AND realnamelen=1-3	0.531	83	83	100.0%	0.3%
realnamelen=1-3 AND tasperpost=>11	0.531	140	140	100.0%	0.5%
domaincount=4-54 AND tld=biz	0.531	25	25	100.0%	0.1%
realnamelen=13-15 AND tld=biz	0.531	27	27	100.0%	0.1%
tasperpost=>11 AND tld=biz	0.531	23	23	100.0%	0.1%

Table 7.21: Concept Discrimination of *spammers*. The table shows the top 20 subgroup descriptions for the target concept *class* = *spammer*, using the discrimination setting. We applied the quality function q_{RG}, i. e., the relative gain quality function. For a description of the remaining parameters see Figure 7.18.

7.6 Summary

This chapter analysed features and methods to detect spam in social bookmarking systems. First, a categorization of possible features and their suitability for spam classification was explored. It could be shown that semantic features are of great help to identify spammers. A combination of all features delivered the best results. From an algorithmic perspective, SVMs and logistic regression delivered good results, though no classification algorithm significantly outperformed others. Second, the ECML/PKDD discovery challenge 2008 was summarized. The results of the different participants were briefly introduced and compared to our previous work. The best results were obtained by treating a post as a text document and conducting text classification on the dataset. Finally, spam and non-spam patterns were discovered by means of subgroup discovery. The patterns help to understand the differences in user characteristics and behaviour.

Overall, classification algorithms and the features introduced in this chapter are suitable for spam detection. However, there is room for improvement. Over time, most of the social bookmarking systems have implemented more functionalities so that more user information is available. For example, BibSonomy allows users to build a social network by either explicitly building friend links or by following other users. These social networks could be used to extract features such as the number of spam / non-spam friends or the

number of users a participant is following.
An important information not considered in this work is the temporal perspective. Do spammers change over time? Are the same features which helped to detect spammers in the beginning still important to detect the new wave of spammers? Does a preselection of training instances tailored to a specific period help to improve spam detection results?
In the scope of a bachelor thesis [Borchert, 2011] the last question was analysed. Interestingly, it could be shown that there is a difference regarding how to treat spam and non-spam examples. While non-spam examples should always be included in a training dataset, spam examples from recent classification activities are sufficient to characterize spammers. This is also due to the skewed dataset leading to a very small amount of non-spammers. The bachelor thesis also analysed active learning techniques to better select spam examples. However, there was no significant improvement over selecting examples by chance. From this thesis we concluded that, as long as a certain amount of spam examples are contained in the training set, no other preselection methods are necessary. However, it is important to increase the amount of non-spam examples relatively to the amount of spam examples.
While we focused on analyzing different features, a deeper investigation of algorithmic details might be helpful to better tackle the problem. For example, the implementation of a kernel triggered by the problem of spam might improve spam classification. Keeping in mind that the best classifier of the ECML/PKDD discovery challenge used textual classification, string kernels [Amayri and Bouguila, 2010; Sonnenburg et al., 2007] could be interesting to look at. Also, the performance of SVM algorithms could be improved by implementing online updateable techniques [Sculley and Wachman, 2007]. Such approaches are especially useful for advancing the BibSonomy spam classification framework (introduced in Chapter 9) in order to make the framework more flexible to ongoing changes.

Chapter 8

Data Privacy in Social Bookmarking Systems

8.1 Introduction

The problem of identifying spam in social bookmarking systems has been discussed in the previous chapter. We showed how classical machine learning techniques can be used for this task. Most of these methods create a mathematical model from positive and negative user examples. New system users can then be classified with the generated model. The training examples consist of different features describing a user. The more descriptive the features, the better the performance of the classifier. Hence, a careful feature engineering is an important task to build efficient and effective spam classifiers.

Feature engineering involves the collection and storage of user data. Although the data may not be published or forwarded to other companies, the fact that user data is processed can be seen as an invasion of user data privacy. In the light of a growing awareness among internet users regarding the storage and usage of their private data, the protection of user privacy can be seen as a system's seal of quality and become a competitive advantage that is not to be underestimated. Depending on the system provider's country, such data collection can even be illegal. Thus, from a legal and user-friendliness point of view, a system provider should favor features built from non-private and publicly available data.

In order to balance performance and data privacy aspects in social spam detection, this chapter presents a privacy-aware feature engineering approach as published in Navarro Bullock et al. [2011b]. The paper is a conjoint work from authors with a technological and legal background and combines computing aspects with legal expertise. In contrast to the creation of privacy enhancing-technologies, including approaches such as data anonymization or the definition of user privacy policies, we start our privacy-aware consideration at the beginning of the data mining process – when data is collected and spam classification features are generated. Our contribution consists of an integrative pattern demonstrating how to evaluate classification features according to performance and privacy conditions. In order to show the practicability of our approach, we will conduct extensive classification experiments using features generated from different data sources of the social bookmarking system BibSonomy. The experiments will show that performance and privacy aspects must not be mutually exclusive.

The technical background necessary for the experiments has been introduced in Chapter 4.

Chapter 5 explained basic legal concepts. In the following sections, the legal background will be extended with a discussion about spam and privacy concepts (Section 8.2). Section 8.3 provides details about the experiments, discusses data privacy-levels and presents results. Section 8.4 summarizes our findings.

8.2 Legal Analysis of Spam Detection

Building models for spam detection requires the collection and processing of user data such as IP addresses, content and log data. Looking at the plain data, it is possible to associate the data with a natural person (the user). As a consequence, data privacy regulations and laws must be considered when implementing and running spam detection systems. The relevant data privacy law in Europe are national implementation acts of the European Data Protection Directive 95/46/EC [Krause et al., 2010].
As already mentioned in 5.1.2, 95/46/EC Article 2 a) defines personal data as

> any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity.

In order to identify a person, the controller (in our case system provider) of a social bookmarking system has a variety of information available: profile data such as age, sex or contact, content data such as his or her posts, and the usage behaviour. Even if the majority of data sources does not reveal a user's identity, often the combination of facts allows for an identification. As soon as a person is identified the rest of the user's data becomes personal as well. Thus, even if users post under a pseudonym, their data is private data as soon as they reveal some kind of personal information, for example when adding the tag "myown" to a publication with only one author in BibSonomy). It is therefore difficult (probably impossible) for service providers processing social bookmarking data to eliminate personal data in their data basis. Thus, all data must be considered as personal data.
The collection and processing of personal data for the task of spam detection can be justified legally. Article 7 b) of the Data Protection Directive states that "personal data may be processed only if processing is necessary for the performance of a contract to which the data subject is party [...]." The main task in order to fulfill the contract between user and social bookmarking service provider is to ensure that users can store their posts and use the system in the agreed upon way. Is spam detection part of this contract? In Navarro Bullock et al. [2011b] we conclude:

> This data has been originally stored for another purpose, such as the technical realisation of the usage or publication of contents. The usage of data for spam detection constitutes a change of purpose, so the lawfulness of the processing needs to be re-examined. The purpose, spam detection in a social bookmarking system, enables the fulfilment of a contract, namely the realisation of the service the operator offers the user in the context of the usage relationship. The lack of spam detection measures in systems without regulated access, like BibSonomy, quickly results in a degree of spam infestation that renders

> the system inoperative for the legitimate user. The user relationship, therefore, includes a certain minimum usability which cannot be provided without an effective spam detection scheme.

The design and provision of the service, however, must then confirm to the privacy regulations. Such data privacy laws specify the conditions under which personal data can be processed. In Section 5.1.3 we introduced privacy principles providing the basis for all regulations including national data privacy laws. Article 6 (1) a) 95/46/EC reflects those principles [Navarro Bullock et al., 2011b]:

> According to [the article], data processing must be carried out fairly and lawfully, i. e., in a transparent way for the data subject and solely on grounds of law or by consent of the concerned. The principles of purpose limitation, relevance and necessity apply, as well as the principle of data reduction. This means that personal data is solely to be collected and used in accordance with either specific, legally permissible or consented purposes, and only as long as it is relevant and necessary. In case of a change of purpose – meaning data collected for a specific purpose are to be processed for another one – the legal requirements have to be fulfilled anew for each processing stage. As little personal data as possible is to be collected and processed and must be erased or anonymized as soon as possible. Often the law calls for a weighting of the respective interests of the data controller and of the data subject, which only allows data processing if and when it falls in favour of the controller. Transparency of all the data processing stages for the concerned user is the prerequisite to exercising his rights.

If part of the available user data (be it single information such as the user name, or a combination of different facts such as the click history) enables the categorization of users into spammers and non-spammers, such pieces of information can be seen as relevant for the purpose of spam detection. If no other data is available with the same effect, it becomes necessary. According to Navarro Bullock et al. [2011b] "the operator, however, is not allowed to process any kind of data to reach 100% accuracy. Processing must be appropriate in every individual case, as it constitutes an interference with a fundamental right." Considering typical data mining experiments where different features and algorithms are available, a reasonable guideline is to select the alternative which performs good, but also requires the fewest personal data.

8.3 Privacy Aware Spam Experiments

8.3.1 Experimental Setup

This section describes the experimental setup considering performance and data privacy conditions and presents the experiment's results.
The feature engineering and classification experiments are conducted using the *SpamData08* dataset of the social bookmarking system BibSonomy, described in detail in 7.2.2 (see Chapter 7). The dataset contains about 2,500 active users who registered until mid-2008, and (despite the implementation of captchas for the registration of new users) more

than 25,000 spammers. It consists of all public posts with URL, title, date and its tags. Additionally, we added data from the system BibSonomy to include non-publicly available registration information such as the users' full names, e-Mail addresses or affiliation and log data. The features which can be derived from the public information include post and (implicit) social network information as described in Section 7.3.1 and Chapter 9. The additional non-publicly available data sources contain registration and log information.

The majority of features we derived from the data are introduced in Chapter 8. To get a better understanding of what kinds of features perform best, we categorized them into different groups (personal, behavioural, textual, network, location). Table 8.1 contains a more detailed description of the features including the feature groups, the data sources needed, a feature definition and short examples to illustrate them. Unfortunately, the features marked with * are not available in our dataset, as they were implemented in BibSonomy at a later time. Using the SpamData08 dataset, however, allows the comparison to experiments conducted by other researchers (described in Section 7.4.4).

Table 8.1: Description of feature groups

Group	Data used	Feature name	Description	Example
Personal	registration information	*maildigit, namedigit, realnamedigit, maillength, namelength, realnamelength*	digits contained in/length of user name, e-mail address, real name of user	"web123@yahoo.de" "krause@cs.uni-kassel.de" digits in e-mails: 3 versus 0; length of e-mails: 15 versus 23
		realname2, realname3	number of separate names in realname	"John Ferdinand Doe" versus "John123" : 3 versus 1
Behavioural	registration info, posts	*datediff*	time between registration date and first post date	A user registered on December 1st and submitted her first post a few minutes later. Some spammers, however, wait a few days or weeks until they get active.
		tasperpost, spammertag	min, max, avg. number of tags per post; specific keywords used in posts	A spam user adds about 4 tags to a resource while a legal user adds 3. Spam users tend to use typical tags such as "money, free or seo"
	logging information	*userlogins*, numclicks**	number of times a user logs into the system ; number of clicks on (spam / non-spam entries)	Spammers tend to click on their own entries more often than other users do.
Textual	posts	*tagposts, bibtexposts, bookmarkposts, allposts*	terms used in post (either bookmarks, bibtex or both) considering title, URL, tags, description	Bag of words of all terms contained in posts of the user
Network	posts	*co(no)spamr, co(no)spamt, co(no)spamtr*	implicit links: co-occurrences networks of tags, resources or both	Two users are linked, because they use the same tags or the same resources or they write the same tags to the same resources. The features count, how many users share tags and resources with other spam or non-spam users.
	social network information	*(no)spamfriends*, (no)spamfollowers**	explicit links: number of (spam/ legal) friends; number of (spam / legal) followers	Friends can be all other users in the system. Followers are users, who are interested in the content of the posts and like to watch new entries of the followed users regularly.
Location	log information + registration information	*spamip, tld*	location of IP address; number of spammers with the same provider in e-mail address	Users with an IP address of a specific country tend to be spammers. Users of the provider "webxxx.de" tend to be Spammers.

We use the toolkit Weka [1] to compute different classification algorithms such as (multinomial) Naive Bayes, logistic regression or decision trees. Following the challenge's task, the models are inferred from examples generated from the training data and tested on examples generated from the test data.

[1] `http://www.cs.waikato.ac.nz/ml/weka/`

Table 8.2: Order of feature groups in consideration of data privacy aspects

Rank	Data Category	Examples	Other
1	anonymised data	All user data that the operator cannot associate with a single user after having removed all features which allow an identification; generally impossible with posts, as they can easily be re-associated by comparison with the permanently saved and published information	
2	publicly available data	Posts marked as public by the user, including tags, keywords, resources, published contact and profile information, friend and follower links, even registration information such as e-mail address, real name, user name etc. if published	Preferably procession in pseudonymised form by department without access to the identification key
3	registration information	All registration data not explicitly published such as e-mail address, real name, user name	ditto
4	logging information	IP address, time information of registration and posts, number of times a user logs into the system, number of clicks on (spam / non-spam) entries	ditto
5	explicitly not published data	Posts, contact and profile information marked as private by the user	ditto

8.3.2 Evaluation

In order to evaluate the performance, we use the AUC (Area Under the ROC Curve) measure as described in Section 4.2. To evaluate the privacy-friendliness of the different features we briefly analyse the legal conditions and present a simple-to-use categorization of data privacy levels.

Privacy-friendliness of features [2]

> Service providers should always opt for the most privacy-friendly alternative when choosing features and methods for spam detection. First of all, the fewest possible personal data should be used. Non-personal or non-identifiable data are not subject to data protection laws and can be used without restrictions for any kind of data processing. This also applies to anonymised data which the operator can no longer associate with a user and thus with the natural person of the user. Such an anonymisation might be

[2] This section is the legal analysis and has been provided by the co-author Hana Lerch in Navarro Bullock et al. [2011b].

practically impossible, given the countless data combination possibilities as well as multiple ways to identify single persons – [which is] often desired by the user – through public posts and other linkable user information. [...] This way, only the classification as spam of single posts (and not of the entire user account) can be carried out.

In order to lower the risk of improper usage of data for other purposes, a separate in-house department could be assigned with the data processing for spam detection after the pseudonymisation of data to be used [takes place]. For that purpose, the operator [first of all] removes all attributes that render the identification possible and replaces them with a key by means of which he can later re-associate the data with a user account in [case it has] been positively identified as spam. This is[, however,] not an anonymisation, since the operator remains capable of identifying the data subject [...]. Alternatively, the data mining process for spam detection can be transferred to an external provider [...] [after the data has been pseudonymised so that] the processing entity cannot identify the data subject. The external provider only communicates the hits to the operator, who then re-assigns this data with the respective user account to take further measures. The weak point in the design might be, as in the above mentioned anonymisation case, the possibility of identification via data from public posts etc. which can hardly be excluded.

The distinction between data published by the user (public posts including tags, user name etc.) and non-public data generated by the user (registry information, utilisation data such as information on the activity of a user, her IP address, etc.) is another criteria for the selection of spam features. For the above mentioned reasons, the purpose of spam detection generally legitimates the use of both public and non-public data provided the respective data is [truly] indispensable for an efficient spam detection. Public data is often personal as well and thus not usable without any data protection restrictions. The [difficulty] caused by the usage of non-public data, however, is more severe. Users do not publish data [explicitely] for the purpose of spam detection but at the same time [they] indicate that this information shall be visible for everyone and thus abandon the higher protection they expect for the data they deliberately mark as private [...]. Similarly, the utilisation data[,] whose generation and storage the user might not even be aware of, can offer an equally high [amount of information] concerning her interests and personality as public data do, especially [when combined] with the latter.

Due to its lesser [restrictions], the use of public features is preferable. If the exclusive usage of public data delivers satisfying results in spam detection, non-public data should not be used. A minimum accuracy value cannot be given, as further developments in spam detection research and the adaptability of spammers will influence this value. However, a minor increase in precision cannot justify the usage of far more sensitive features. In any case, a success rate of 100% is hard to achieve due to technical reasons. [Table 8.2] offers a rough orientation for the selection of spam features by designers of spam detection systems. In specific, well-founded exceptions, variations may be justified. The categories are arranged according to [...] their data privacy

Table 8.3: Performance overview of AUC values for the best classifier of each feature group and feature computed using the SpamData08 dataset. Textual features perform best, followed by network features.

Feature Group	Feature Name	Classification Results		Privacy category
		AUC Value	Best Classifier	
Personal	*maildigit, namedigit,realnamedigit*	0.68	Naive Bayes	(3) registration information
	maillength, namelength, realnamelength	0.68		
	unimail	0.553		
	All	0.776		
Textual	*tagposts*	0.919	Multinomial Naive Bayes	(2) publicly available data
	bookmarkposts	0.951		
	bibtexposts	0.696		
	All	0.956		
Network (implicit)	*cospamr, cospamt, cospamtr*	0.718	Logistic Regression	(2) publicly available data
	conospamr, conospamt, conospamtr	0.653		
	All	0.903		
Behavioural	*datediff*	0.512	Logistic Regression	
	tasperpost	0.747		(2) publicly available data
	numbibtexposts	0.787		(4) logging information
	spammertags	0.754	J48 (pruned)	
	All	0.86		
Location	*domain*	0.709	J48 (pruned)	
	spamip	0.569		(3) registration information
	tld	0.753		(4) logging information
	All	0.798		

level, with the first being more privacy-friendly than the following etc..

8.3.3 Results and Discussion

Table 8.3 presents the AUC value of the best classifier for each specific feature and the feature groups. As can be seen, textual features perform best, followed by network features. Information derived from personal data, such as the registration category, do not perform as well. When combining different features of one category, better classification results can be achieved. For example, the single network features perform rather poor, while their combination achieves the second best result.

To get an insight into the performance of a combination of different feature categories, we combined all categories except the text features. Table 8.4 shows the classification results. The AUC value of just mixing all features (0.863) is lower than the top two feature categories. This can be due to some contradictory features diluting the classifier. By preselecting certain features (using the SVM attribute preselection method in Weka), a better performance can be achieved (0.941 – last line of Table 8.4). Only the text feature category with a performance of 0.956 (Table 8.3) attains better results.

Considering privacy conditions, the classification results show that features derived from the most privacy-friendly data sources in Table 8.2 perform better in general. Features in the first privacy category, anonymized data, are difficult to create in BibSonomy, as the data (for example the different posts) can be compared with the publicly available entries and the pseudomized name matched to the real user name. Textual and network features as computed in this study (without explicit relationship information such as friends) match the second optimal category: publicly-available data. Those features perform best in the case study. Critical features derived from registration or log information perform worse and - from a legal perspective - should not be used. Consciously marked private entries have not been used in any of the features, so the most critical privacy data category is not considered. A simple combination of all features does not automatically lead to better

Table 8.4: Performance overview of AUC values for the best classifier of all features computed using the SpamData08 dataset. By preselecting certain features (using the SVM attribute preselection method in Weka), the best combination of features can be achieved (0.941). Only the text feature category with a performance of 0.956 attains better results.

Feature Group	**Feature Name**	**Classification Results**		**Privacy category**
		AUC Value	**Best Classifier**	
All	features of above without textual features	0.863	Naive Bayes	(2), (3), (4)
All	feature combination (*bibtex, conospamr, cospamt,conospamt, conospamtr, unimail, cospamr, spamip*) without textual features	0.941	Naive Bayes	(2), (3), (4)

classification results. However, as could be shown with the combination of a selection of features, one may obtain better results by mixing different feature classes. Two features of this group, *unimail* and *spamip* (described in Table 8.1) have been generated from critical registration and log data sources. In this case, service providers need to consider privacy risks by either not using the specific, critical features or by informing users about the necessity and purpose of collecting this information.

8.4 Summary

In this chapter, we introduced a privacy-aware feature engineering approach for spam detection in social bookmarking systems. It consists of defining privacy categories for different data sources used to generate classification features. Features can then be evaluated not only by their classification performance, but also by their capability to protect the data privacy of users.

We evaluated our approach using a dataset of the social bookmarking system BibSonomy. It could be shown that textual features derived from posts perform best, followed by network features. Both features use publicly-available data, the second of five identified privacy levels. For this case study, we can conclude that effective spam detection can be conducted without significant performance losses. Features complying to the privacy policies introduced in this paper should therefore be preferred over those that do not. One issue needs to be considered when regarding practical spam detection applications: Spammers need to be identified as soon as possible. Often, only registration and personal information is available after a spammer's registration. Thus, one either has to wait until the first posts before spammers are marked, or privacy concerns need to be given less importance in favor of a "clean" spam system.

While the legal categorisation of spam features can be easily transferred to other systems,

the evaluation of the effectiveness of spam features is difficult to generalize. Spammers will react to the countermeasures introduced by the system provider by changing their behaviour. Therefore, other features and methods might be necessary in response to the adaption of spammers. Future work should therefore analyse the changing behaviour of spammers in social bookmarking systems to be able to further generalise the results. The different privacy categories, however, can be used as an orientation scheme for other data mining applications, such as recommendation and ranking services.

Part III

Applications

Chapter 9

BibSonomy Spam Framework

After a few years of running BibSonomy, due to the overwhelming amount of new registrations in BibSonomy, most of them spammers, the manual classification of spammers by the system's administrators was no longer possible. In 2009, system administrators were labeling about 200 spammers a day. At the end of 2009, the BibSonomy spam framework was introduced to reduce the manual labeling efforts of system administrators. This framework automatically classifies users based on a model learned regularly from BibSonomy training data. The features and classification methods of the framework were first evaluated by the experiments introduced in Section 7.3 and then slightly adjusted to fit a real life scenario. All user and classification data is collected in order to enable further research around spam classification.
In the following section we will first summarize the relevant facts about the framework and then describe the framework's processes and architecture in detail.

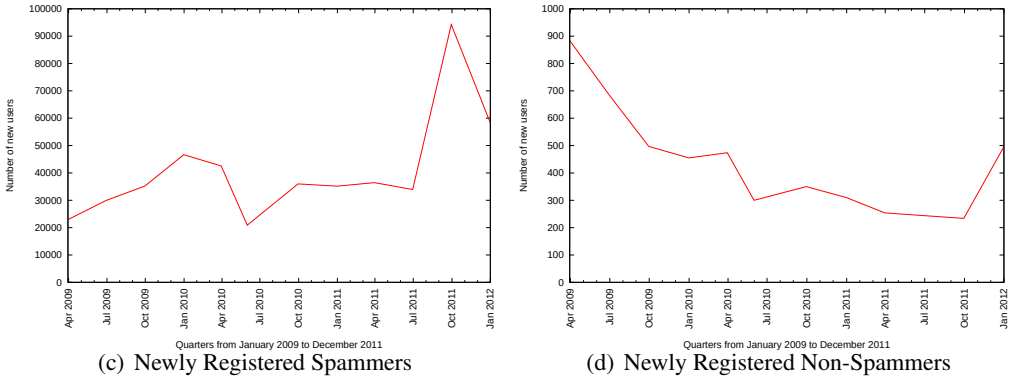
9.1 BibSonomy Spam Statistics

In order to get an overview of the framework's work load, we will present some figures about the number of users and spammers, their usage of BibSonomy and the basic performance of the framework. The time period ranges from 2009 until end of 2011 [1].
Table 9.1 presents basic statistics aggregating the three years. As can be seen in the first line, BibSonomy had nearly 730.000 registrations during that time period, whereby roughly two thirds were identified as spam. The rest are either non-spammers or "empty" registrations, i. e., users who never submitted a post and therefore have never been classified. When only considering users with at least one post (second line of the table) the sizes are smaller. The proportion of spam versus non-spam users is even more skewed: the system then contains about 500000 spammers and 5000 non-spammers.
Most of the registrations were handled by the spam framework: About 410000 users were classified by the framework without being changed by the administrators. About 80000 user flags have been touched by administrators, as the last line shows. Please note that the fourth and fifth lines do not sum up to the number of total registered spammers, as they only account for users who were updated during this period. Users updated at a later time have not been considered.

[1] The time I was an active researcher at the BibSonomy team.

Table 9.1: Basics figures of BibSonomy and the spam framework until end of 2011

	Spammer	Non-Spammer	All
Number of total registrations in BibSonomy	493234	229759	722993
Number of users with no posts	996	224564	225560
Number of users with at least one post	492238	5195	497433
Number of users classified by the framework and not being changed by admins	407254	1698	408952
Number of users classified manually	83527	2044	85571



(c) Newly Registered Spammers (d) Newly Registered Non-Spammers

Figure 9.1: Newly registered spammers and non-spammers having at least one post tracked over time. Please note that the scale of the y-axis is different as the amount of spammers exceeds the amount of non-spammers many times in BibSonomy.

Figure 9.1 depicts the number of spammers and non-spammers who registered during each quarter between 2009 and 2011. The number of new non-spam registrations is decreasing in general, which can be attributed to the fact that many potentially interested users have already registered or use another social bookmarking system. The peaks are mostly due to events where BibSonomy was used. For example, in October 2010, many users registered to BibSonomy during the KDML conference in Kassel [2]. The spam curve shows a first peak around June 2010. A second, more extreme peak appears in the fourth quarter of 2011. The actual cause for the different peaks are not clear. One possibility is that those peaks represent spam attacks where a few spammers register many times (under different names).

9.2 Framework Processes and Architecture

Figure 9.2 depicts the spam classification process by means of a BPMN collaboration diagram[3]. The collaboration represents the interactions between three participants involved: The user, the framework and the administrator. For each of them the activities they conduct

[2] http://www.kde.cs.uni-kassel.de/conf/lwa10/kdml
[3] http://www.omg.org/spec/BPMN/2.0.2/PDF/

along the spam classification process are clustered in an individual process, i. e., the user' one is "Initial Login", the administrators is "Check User" and the frameworks is "Classify new Users".
Each process consists of one or more starting events (the round circle at the beginning), several tasks and an ending event (the circle at the end). The different items are connected by directed solid lines showing the sequence flow. The tasks describe the specific activities conducted by the actor in the process. The data in- and outputs which will exist beyond the process are depicted by data stores. The focus of the figure, however, is on the description of the process, not the model of the data. The following list describes each participant and its process.

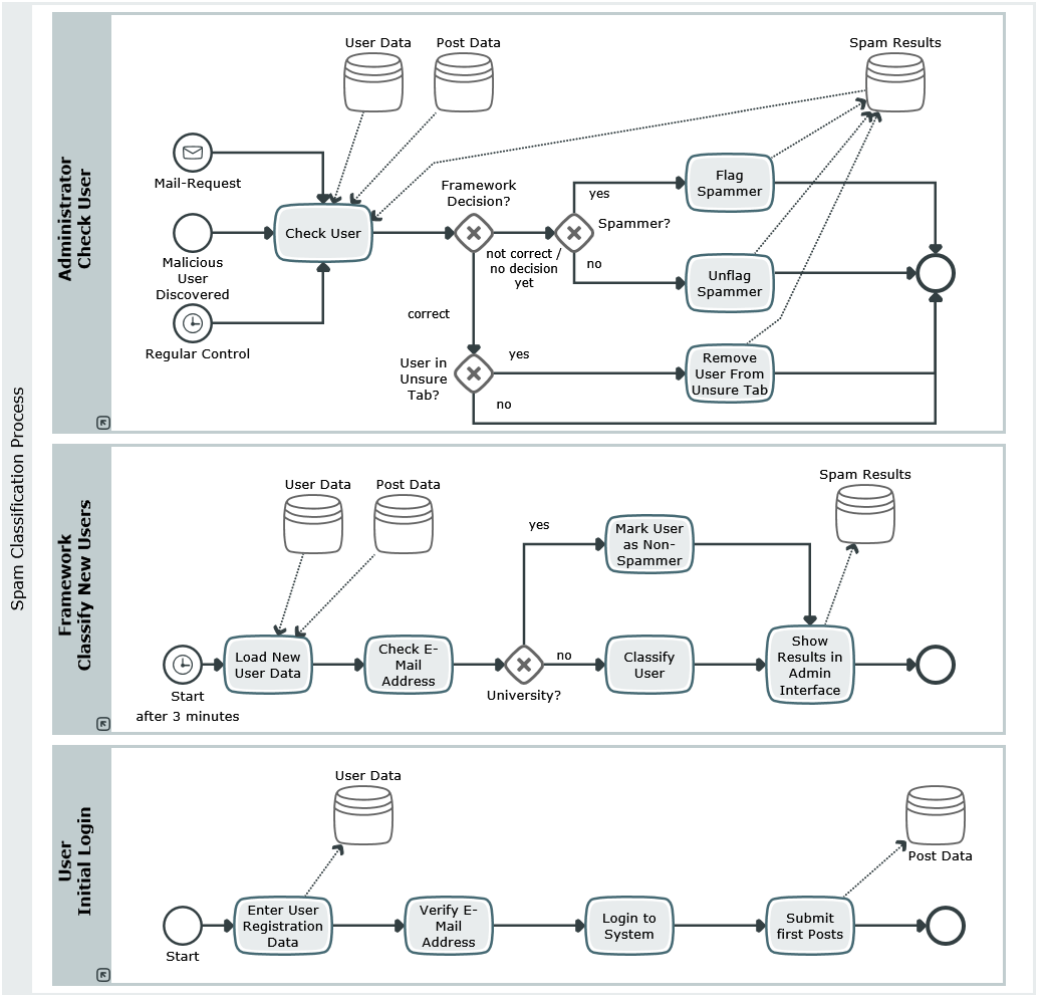


Figure 9.2: The three actors (user, framework and administrator) of the spam classification process and their activities relevant to spam detection.

- A *user* (process at the bottom of Figure 9.2) initially signs up by entering standard registration information such as a name, username, e-mail address and a password. After having verified the e-mail address, the participants are able to login to the system by entering their username, password and a captcha. Only after that can they

submit their first posts. User and post data are stored in a database.

- The *framework* (process in the middle of Figure 9.2) regularly (per default every 3 minutes) loads the data of all new users who have at least one post submitted. First, a user's e-mail address is checked. If it is a scholarly address, the user is marked as a non-spammer. Otherwhise, users are classified by means of a classification model built up from previous spam data. The classification itself is further described below (see Paragraph 9.2.2). The classification results are stored in the database. They are shown in the admin interface and can only be changed by the administrators.

- *Administrators* (process at the top of Figure 9.2) start checking registered users either after having identified a suspicious user themselves (for example by viewing the BibSonomy entry page [4]), by controlling the spam framework's classification or after being asked by a user (normally per mail request) who has been classified as a spammer but claims not to be. The user checking involves looking at the user's registration information in the administration interface and checking their posts (tags, titles, bookmarks). If – after having looked at those items – it is still not clear whether a user is a spammer or not, the administrator opens the original websites linked by the posted bookmarks and checks the content. A decision can normally then be taken. Depending on the classification's decision three actions can be identified. In case, the user is a spammer, but has been classified as a non-spammer, the user is flagged as a spammer. In case, the user has been classified as a spammer, but is not, the spam flag is removed. The framework also shows the degree of confidence. If the confidence value is below a certain threshold, the user is shown in a so-called "unsure" tab. The administrator then confirms the classifier decision so that the user is removed from the set of uncertain users. In case, the classifiers decision was right and the confidence value exceeds the threshold value, no action is required.

Figure 9.3 shows the framework classification in more detail in two lines from top to bottom. One can distinguish between two different "paths": the generation of the training model (left side) and the classification of new instances (right side). We will describe both parts in the following section.

9.2.1 Generation of the Training Model

In order to generate a model, training instances are loaded from the database into the system. Basically, the training instances consist of all instances classified so far. Not all instances are selected for training, however. Usually, training examples where non-spammers own an extensive amount of publication posts are excluded as the trained model becomes biased by such examples. Additionally, it is possible to select instances within a certain time period (for example all instances from the year 2011).
After the selection of instances, features are extracted for each instance. The features to be used for classification are marked in an option file. The use of a feature in the training process mainly depends on its contribution to accurate classification results, but also on its computation efforts. The features implemented correspond to the ones described in Section 7.3.1.

[4] `http://www.bibsonomy.org`

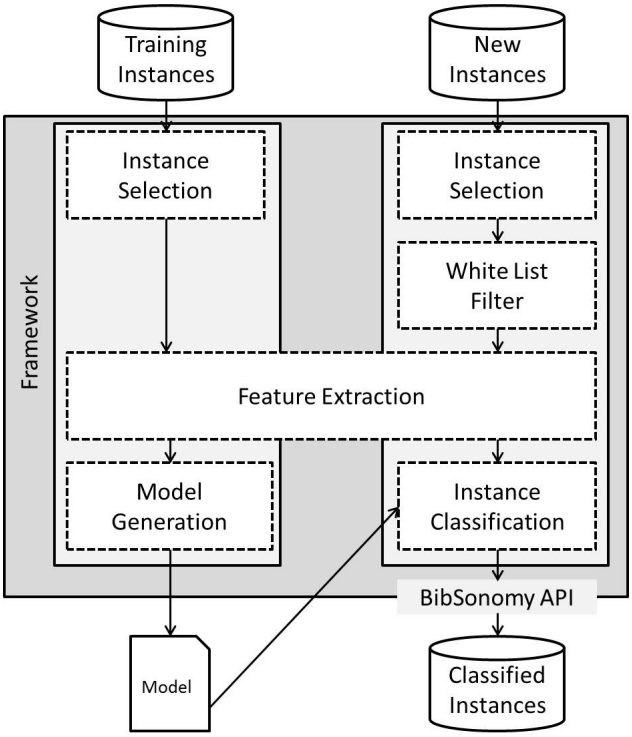


Figure 9.3: Basic components of the spam framework

For classification, one can select between different methods such as logistic regression, SVM, Naive Bayes and a decision-tree-learner. Logistic regression was used most of time, as it allows a fast model generation and classification results were similar to the results of other algorithms such as SVMs. The Weka framework [5] was integrated into the spam framework in order to use its standard classification algorithms. The output is a serialized model written to a file which is indicated by the arrow from the model to the instance classification in Figure 9.3.

9.2.2 Classifying New Instances

The input of the classification of new users are all instances which have not been classified before and which have at least one post. A white list filters users containing e-mail addresses from universities. From the remaining instances, the same features as the features selected for the model generation are extracted. Each instance is then classified using the model. As mentioned in the description of the spam classification process (see Figure 9.2), the classifier does not only decide between spammer and non-spammer, but computes a confidence score. With the help of this score it is possible to distinguish between certain and uncertain decisions. Overall, there are four options:

- *Secure Spammer*: The classifier has high confidence that the user is a spammer.
- *Unsure Spammer*: The classifier has flagged the user as a spammer but with a low

[5] `http://www.cs.waikato.ac.nz/ml/weka/downloading.html`

confidence score. An administrator needs to decide whether the user is truly a spammer or not.

- *Unsure Non-Spammer*: The classifier has flagged the user as a legitimate user but with a low confidence score. An administrator needs to decide whether the user is truly a spammer or not.

- *Secure Non-Spammer*: The classifier has high confidence that the user is not a spammer.

An unsure spammer or non-spammer classification needs to be reviewed by system administrators. The differentiation of secure and unsure classification results therefore helps administrators in their daily work. In order to check the automatic spam classification, they do not need to revise the entire classification lists, but concentrate on the unsure classification results. The classification results for different instances can be seen in Figure 9.4 (see Section 9.4).

9.3 Implementation Details

The framework, implemented in the programming language Java [6] runs independently from the BibSonomy application. It is distributed as a compressed JAR (Java ARchive) file.

The application accesses the BibSonomy database in reading mode and selects training and new, not yet classified users and their related information such as tags, bookmarks and resources. Framework related information such as specific settings, which can be submitted via the administration interface, is stored in a MySQL database[7]. The interaction between the databases and the application is managed by iBATIS [8] (which has been migrated to MyBatis [9]).

The classification algorithms applied are provided by the open source machine learning software Weka (Waikato Environment for Knowledge Analysis) [10]. The Weka library can be directly accessed in the Java application.

Classification results are transmitted to BibSonomy by using the system's API [11]. If a user is a non-spammer, the spammer field in the BibSonomy database is set to 0, if a user is a spammer it is set to 1. Additionally, each post has a specific number associated, indicating if it is a private or public post. If a user is identified as a spammer, this number is converted into a specific spam public or private number. If BibSonomy displays a specific page, such as showing the most recent entries, the application collects only those posts which do not have a specific spam number. Spam posts are therefore hidden from public view.

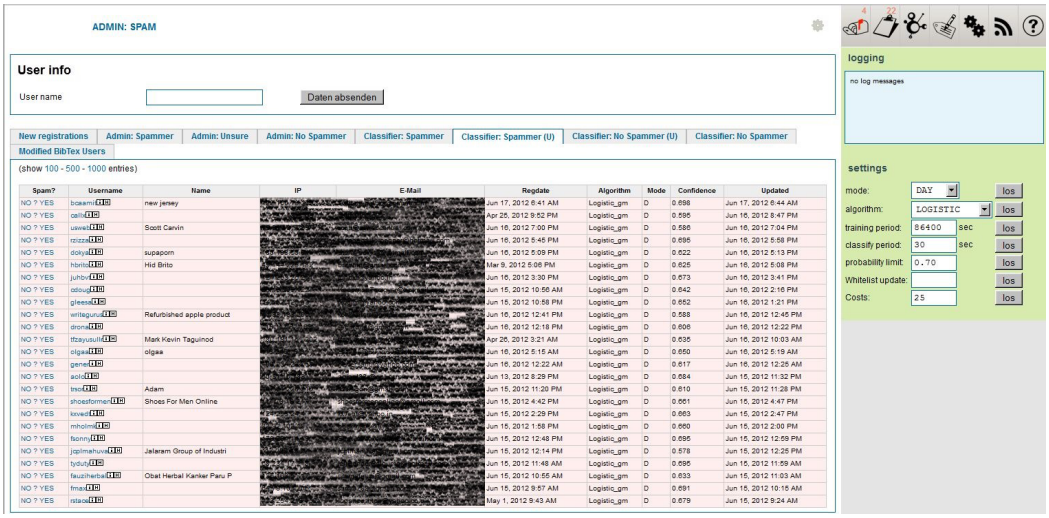


Figure 9.4: Administrator interface used to flag BibSonomy users as spammers or legitimate users. One can see the different tabs. Data is shown from the tab "Classifier: Spammer (U)", i. e., from the unsecure classification results of spammers.

9.4 Framework Interface

The main interface is shown in Figure 9.4. The interface offers different tabs summarizing information about different BibSonomy user groups. For example, instances of the four classification categories defined in Paragraph 9.2.2 can be viewed in individual tabs. Furthermore, users who registered recently or users who posted BIBTEX posts can be viewed separately. For each user appearing in one of the tabs, the username, name, ip, e-mail and registration date are shown. This information is normally sufficient to get a feeling for whether a user is a spammer or not. Additionally, information about the framework's classification is given. Administrators can view the type of classification algorithm, the classifier's confidence and the date when the status of the classifier was updated. Clicking on the small icons to the right of the username releases more details: A popup presents the last four posts of the particular user and the number of BIBTEX entries.

Sometimes, administrators require specific spam information about a user. They can use the *User info* box and type in the username. They will get the same user information as shown in the main interface. Finally, if administrators want to change specific settings, for example, the classification algorithm, they can specify the information in the *Settings box* on the right hand side of the user interface.

[6] http://www.java.com
[7] http://www.mysql.com
[8] http://ibatis.apache.org/
[9] http://code.google.com/p/mybatis/
[10] http://www.cs.waikato.ac.nz/ml/weka/
[11] http://www.bibsonomy.org/help/doc/api.html

9.5 Summary

The BibSonomy framework automatically classifies users by extracting different user features and applying a classification algorithm from the machine learning software Weka establishing a simple workflow to classify spam. The framework offers administrators a simple interface to flag or unflag users. Due to the amount of spam in BibSonomy, the framework is an essential part of the BibSonomy application.
Further development of the framework has been carried out based on the methods explored in Chapter 7, i. e., if better features and algorithms are explored the framework can be improved. However, there are a few aspects which need to be especially considered:

- Not only is the improvement of accuracy important, but the computation of features for new users and their classification need to perform at run time. Long classification times would lead to a slower flagging of new users. The BibSonomy home page showing the latest posts would be spammed heavily if the identification of spammers was not performed quickly enough.

- The amount of information about users available in the framework is not the same as the information comprised in the dataset. The decision whether a user is a spammer needs to take place as soon as possible in order to prevent the publication of spam posts. Therefore, user characteristics need to be based on features exploiting data from the beginning of a user's interaction with the system (such as registration data).

- Though most of the evaluation measures and experiments try to minimize the problem of false positives, the false classification of legitimate users remains a problem. There are, however, some mechanisms to cross-check users. For example, users who post BIBTEX entries after having been classified as spammers are again considered in the classification. However, a real time improvement of the false positive rate is critical to gain new system users.

Besides classification performance, future work needs to consider the implementation of further framework features. For example, a website evaluating the classifier online would help to track performance. Mechanisms to reclassify users (and how to select the ones which should be reclassified) could be better explored. Finally, the usability can be improved by transferring the framework's administration to the interface (currently an option file needs to be edited) or by exploring new GUI techniques to better present a user's characteristics.

Chapter 10

Case Study of Data Privacy in BibSonomy

In Chapter 8 we analysed the performance and privacy aspects of applications detecting spammers in social bookmarking systems considering the system BibSonomy.
In this chapter we will extend the discussion of handling private data to social tagging systems in general. The chapter is a slightly modified translation of a study published in German in [Krause et al., 2012] as a result of a research collaboration of Andreas Hotho, Gerd Stumme and the thesis' author (technical side) and Hana Lerch and Alexander Roßnagel (legal side).

10.1 Introduction

The nature of social tagging systems, namely the annotation and sharing of resources, generates huge amounts of user-generated data which is publicly available. Depending on the tagging system, such data includes profile data, where people publish their names, biographical information, opinions and interests. The main kind of data, however, are annotated resources such as bookmarks, photos or publications. These items often describe user interests, opinions and personal relationships. Because of the public sharing of such entries, they are available for anyone accessing the Internet.
Discussions as to which extent such data can be used for improving search results, personalized advertisements or profile creations of Internet users exist (see for example in [Heymann et al., 2008; Kolay and Dasdan, 2009a]). However, such analyses often omit a careful consideration of the personal sphere of users, providers and third parties involved. The development, operation and usage of social tagging systems touches the data privacy of humans and institutions. Such infringements lead to legal questions, especially in the area of data privacy, but also in the fields of copyright, competition regulations, protection of minors and criminal law. In many cases, a professional knowledge exchange between technical providers and legal practitioners resulting in a common design of new in our case social media applications is missing.
The following study is an example that can be used to fill this gap for social tagging applications by conducting a technical and legal examination of typical social tagging functionalities referencing the existing social bookmarking application BibSonomy (see Chapter 2.6). All fundamental aspects, from the registration of a user, over different possi-

bilities of using social tagging systems to the termination of membership, will be discussed with respect to legal data privacy issues which may arise by taking part in such services.

10.2 Data Privacy Analysis

The different functionalities and legal issues discussed consider the registration process, the storing of posts and publication metadata, the search of system content, forwarding data to third parties and the termination of a user's membership. As BibSonomy is operated by a research institution (see Chapter 2.6 for background information about the system), the specific legal situation of such systems is briefly discussed at the end of this chapter. The legal discussion is based on the privacy terms and the German privacy law as introduced in Chapter 5. Each of the following paragraphs first describes the functionality in question, then analyses the data used with respect to data privacy issues and ends with a recommendation on how to respect privacy regulations while implementing and offering the functionality.

10.2.1 Registration

System Description The first step towards being able to use BibSonomy is the creation of a user account. A registration form asks for the desired user name, the real name, an e-mail address, an optionally existing homepage and a password. Users also need to enter a captcha to successfully submit their information. Data fields which are required (e. g., user name, e-mail address and password) are marked with a star. Additionally, users have to acknowledge that they read the general terms and conditions of BibSonomy and its privacy statement. Once this is acknowledged the user's e-mail address is verified and access to the system is given.
The profile data collected during the registration process can be administered by means of a user's settings page. Users can decide which data is public and they can delete, change or add information. For instance, users can publish their real name and a link to their homepage so that they become visible to other users in form of a digital curriculum vitae (a functionality offered by BibSonomy especially for scientific users) or in their public profile form.
Legal Situation According to § 12 Abs. 1 TMG (see Section 5.2.2) a service provider is only allowed to collect and use private data to offer some kind of telemedia services if this is permitted by an explicit regulation or if the user has agreed to the collection and usage of this data beforehand.
The fact that users voluntarily submit information about themselves, for example via a registration form, cannot be considered as written consent. Without an explicit user agreement the usage of private data needs to conform to § 14 Abs. 1 TMG: "The service provider is only allowed to acquire and use the personal data of a user as far as it is required for the establishment, content-related design or change of a contractual relationship between the service provider and the recipient concerning the usage of telemedia".
Consequently, only user-related inventory data which is required for the creation, design and modification of the contractual relationship concerning the usage of telemedia is allowed to be collected by the service provider. The data is only allowed to be collected for this purpose. The decision whether personal data is required to establish the contractual

relationship depends on the situation. For instance, a real name and an account number needs to be collected in order to register at a non-free service.
In case a service is specialized in providing personalized features to a user, the information required to enable such personalization can be collected and used for this purpose. The usage for another purpose is only allowed in case the user explicitly allows it. If a system does not offer personalization, only the data required to create a password and user account are allowed to be acquired.
The collection and usage of a real name in order to publish it in the scope of an agreed upon functionality such as the digital curriculum vitae is therefore allowed. If a user has agreed to certain e-mail based services such as being informed about news in the system, technical disruptions or information about one's user account, the collection and use of a user's e-mail address is also allowed. However, the e-mail cannot be used for any other purpose without explicit permission.
Guidelines According to the principles of data minimization and avoidance, providers are supposed to collect and store nothing more than the absolutely essential details acquired to provide the service. Considering a particular case, one has to analyse which data is necessary to provide the service agreed upon between service provider and user. Only the required data can be collected and used and only for the specified purpose. Depending on the grade of personalization, a service may require more information about a user than another. Data which are not explicitly necessary for the provided service can only be collected and processed when an explicit user permission has been obtained.

10.2.2 Spam Detection

System Description Another reason why service providers may collect and process inventory data is the detection of user accounts created for the purpose of spamming (see Chapter 8 for an analysis of registration features supporting spam detection). Users posting spam entries need to be identified as soon as possible in order to remove their posts from publicly available web pages. The spam filter applied by BibSonomy (see Chapter 8 and Chapter 9) uses features created from data of the registration process such as the e-mail address or name. Names with numbers tend to point to a spammer while accounts where a full real name (surname, family name) has been submitted via the registration form are more often accounts of legitimate users.
Often, spammers add many tags to their posts in order to make them easily retrievable for other users. Furthermore, spam users use a specific vocabulary which differentiates them from legitimate users. Features inferred from such characteristics help to detect spammers automatically, especially if a new user submitted only a few posts. Furthermore, the probability that a user who registers with an e-mail address from a university is a legitimate user is very high, normally no further spam features need to be analysed in this case.
Legal Situation A service provider does not have to accept that the provided system is impeded or even disrupted by the destructive actions of spammers. In order to maintain the system, it is justifiable, that the provider conducts appropriate actions with user data with the purpose to identify disruptive users. However, the fight against spammers needs to take place within boundaries in order to protect the rights of legitimate users. Whether a specific kind of inventory data can be stored and used for spam detection depends on its contribution in the spam detection process. The more it helps distinguishing spammers

from legitimate users, the more necessary it becomes for the process. Service providers should rely on relevant research results where they exist. This thesis introduced an analysis of the contribution of features based on different kinds of data including inventory data in Chapter 8.

10.2.3 Storing Posts

System Description One of the first interactions of registered users with BibSonomy is the storing of resources in form of bookmarks from web pages or publication metadata. In the first case, the provider stores a bookmark for a web page and the user adds descriptive tags to it. Different possibilities exist to store information about a publication. For example, the relevant data can be automatically extracted from an external web page where the reference is marked or scrapers automatically extract publication metadata from a website with a predefined format (often the case with digital libraries). On top of the extracted metadata such as author, title, publication medium or the digital object identifier (DOI), additional user relevant information such as tags, descriptions or comments can be added. The service provider stores the submitted data in order to make them accessible for the posting user and for general public access on the Internet.

Legal Situation Since the storing of bookmarks and publication metadata (i. e., content data) can be considered as the main purpose of using the social bookmarking and publication sharing system BibSonomy, and the transmission of the required data by means of the Internet is necessary for the collection and storage of this data, the service provider is allowed to store and process the data (§ 28 Abs. 1 S. 1 Nr. 1 BDSG) for this purpose.

In social bookmarking systems, however, content data as described above is often used by data mining applications such as recommendation services or spam filtering. The analysis of content data in order to detect possible spammers can be seen as being acceptable by law. The detection of spam helps maintain a qualitative service for legitimate users. It preserves storage capacities which can in turn be used for non-spam posts. Hence, the processing of content data to build features for spam detection is necessary in order to offer the service a user expects from the provider and has agreed to when submitting his or her posts.

In order to decide whether content data can be processed for other data mining applications than the automatic detection of spam, one needs to determine if the usage of such data serves to establish the functionality the user has previously agreed to. If this is the case, the usage of content data for (internal) data mining is allowed. Depending on the contractual relationship, such data mining applications include recommendation algorithms for the suggestion of tags, resources or other interesting users.

The assumption, however, that data mining services based on content data are of interest for the concerned user is not enough. Instead, a user has to be explicitly aware of the specific functionality when selecting the service. Considering § 28 Abs. 1 S. 1 Nr. 2 and Nr. 3 BDSG, usage of data for another (not agreed upon) service can be allowed when there is no conflict with legitimate user interests. However, this depends on the particular situation and needs to be check on a case-by-case basis. One cannot globally justify the usage of content data for improving the functionality of online applications.

The creation of user profiles with the help of content data in order to provide personalized advertisements therefore normally requires the explicit agreement of the user (according to § 28 Abs. 3 BDSG).

Guidelines: Functionalities of a social bookmarking system requiring content-related data stored by users have to be explicitly agreed upon by the user when the contractual relationship between user and provider is formed. This could happen through a listwise enumeration of functionalities and data used during the registration process. Otherwise, users or providers might encounter uncertainties regarding non-specified functionalities. In the worst case, certain functions could be deemed illegitimate regarding the scope of the service, meaning that the providers have to obtain explicit permission in order to process user-related data for the functionality in question. This applies irrespective of the fact, that data usage has to be defined in the scope of the data privacy statement.

10.2.4 Storing and Processing Publication Metadata

System Description: Sometimes parties complain about posts of other users in which they are mentioned (for example as authors). Mostly, such complaints are submitted to the service provider via e-mail. Typical examples are

- misspelled author names
- wrongly mentioned or missing authors names
- wrong name of the publishing media (the journal or conference).

In most cases, the complaining party asks the service provider to correct the (in their opinion) wrong information. Though this is technically possible, it contradicts the nature of the self-administering social bookmarking systems: normally, only users, not the providers, are allowed to update user posts.
Legal Situation: Metadata can be considered content-related data of the system's user. At the same time it can be seen as a description of the personal situation of the referenced authors and possibly of further individuals such as co-authors or publishers. Provided that such individuals can be identified through the published data or in combination with further information the data needs to be considered as personal related data. Often, search engines help identify people using such metadata.
According to § 35 Abs. 1 BDSG, non-correct personal data need to be corrected. If metadata are incorrect, they have to be revised by the entity which is responsible. §3 Abs. 7 BDSG characterizes this entity as the person or institution that collects, processes or uses privacy related data for its own purpose or for third parties. In the context of "user related content" it is not yet clear which party is responsible: the content posting user or the system provider. There is good reason, however, to consider both responsible. The content posting user is responsible because he or she select the data themselves and expect the provider to publish and share this data. At the same time, the content provider has technical access and provides the infrastructure to make them publicly available to other people.
Providers cannot hold the user exclusively responsible for the erroneous data, but need to correct it in case they are aware of the problem. Thereby, they are not required to intervene if the effort to find the correct facts is disproportionally high. Often, a search engine request already helps to find the correct information. Incorrect names or missing authors might be clarified by checking an identity card of the author in question. If providers still have doubts about a correction request after having checked it with their available means,

they need to prevent the data in question from being publicly available until the problem is solved.
Guidelines: Since the responsibility of who needs to correct private data has not been clarified yet, a provider should try to correct such data in his or her own interests. A first step would be to inform the user that posted the incorrect data. If he or she does not react or refuses to help, providers should block the publication of the incorrect data or change the data themselves. If it is not clear whether a request to change certain data is correct, the provider needs to clarify the issue. Again, if the issue in question could not be clarified correctly, the provider should block the data.

10.2.5 Search in BibSonomy

System Description: Similar to search engines, social tagging systems help their users to find interesting and relevant information. The process of finding information is enabled by a specific navigation structure, which helps the user to easily browse through the data. Tags added to a certain post as well as the post's users are linked. Beginning with a post of one's own or with a general page (for example the most popular or the most recent posts), users can click on the tags and users related to the post and find other related information. In [Cattuto et al., 2007] the small world characteristics could be experimentally shown for folksonomies: A user can reach a totally different topic than the starting one in a few clicks. In BibSonomy, the number of steps to reach another topic is about three [Cattuto et al., 2007]. In addition, strongly connected points are mostly related in their content. As a consequence, users can find interesting information by following the links of a specific entry.
In order to find specific information, social tagging systems offer a search function which lists posts related to a search request. Often the list is ordered according to relevance.
By sharing information, browsing and searching the system, users leave traces. For example, users click on a specific link, enter a search term or copy a publicly available post to store the post in their own profile. Such interactions with the system can be recorded. Technical means for this are the protocols of the web server containing IP-addresses, request dates or the referrer, javascript functionality to keep track of user interactions with the system and cookies providing an identification of user sessions.
In contrast to traditional search engines, the information collected within a session (considered as usage data) cannot only be assigned to a specific cookie-id, but often to a registered user who was logged into the system when entering the usage data (see the description of click - and tagging data in Section 6.4). Such assignments between usage data and user names allow the construction of user profiles which can be in turn used to improve the system service (for example by providing a personalized ranking for the profile's owner). Provided the users agree to it, the data can also be forwarded to commercial companies, which provide personal advertisements based on such profiles. In BibSonomy, usage data is not made available to commercial companies.
Legal Situation: Usage data and -profiles can be seen as private data if a service provider can assign such data to a real person. For example, this is the case when a user's registration data contains a real name or an e-mail address with the full name. Another way of identifying users is to examine their posts (especially publication posts). Often those posts contain information which identify a user. If a provider does not technically prevent such identifications, all usage data of the provider need to be considered as private data.

Even if single user accounts do not allow the identification of the user, the data remains private data as it is impossible for the provider to distinguish between identifiable and non-identifiable accounts.
According to § 15 Abs. 1 S. 1 TMG, privacy-related usage data can only be collected and processed without explicit allowance of the user if it is required for using the service. Such data include the IP-address of the user's computer or the start- and target pages of navigation requests, as such data is technically necessary. A cookie can be considered necessary for certain functionalities, for example to maintain certain user settings within a session.
The creation of user profiles or the use of data mining techniques with usage data in order to realise individual recommendations or rankings is only permitted if such features have been agreed upon in the service contract. In particular cases, it can be difficult to decide whether a certain functionality is part of such a service contract or not. As long as a functionality is not part of the basic functions an average user would expect, the functionality need to be mentioned in the registration process so that users can decide for themselves if they are willing to provide data for it.
Data, that is not automatically accumulated through system usage or is not required for the interaction with the system (such as data of a clickdata analysis), is not included in the regulation just mentioned and therefore cannot be processed without explicit permission of the user. Only if the processor can technically assure that data cannot be allocated to a person or a user account can such data be processed. The usage data which is legally collected can be used for advertisement or for developing a user-tailored service according to § 15 Abs. 3 TMG as long as it is possible to pseudomize the data so that profile data and identification data can not be reunited.
Regardless of the pseudomization, providers creating user profiles still need to inform their users in advance about their intention and the user's right to object. If the concerned user disagrees, the provider is not allowed to create profiles.
Guidelines: The service provider is only allowed to collect and process private data if the data is required for successfully operating the service as agreed upon between user and provider in advance. Features which exceed basic system operations need to be presented to the user during registration. The data can be used for the creation of user profiles or for advertisement purposes as long as the data is pseudomized and the user does not contradict. The user needs to be informed about his or her right to disagree. Other data – such as clickdata to improve ranking algorithms – can only be used if a user agrees to such functionalities or if the provider can technically assure that the processing is anonymous, i. e., no data can be traced back to a particular user.

10.2.6 Forwarding Data to a Third Party

System Description: BibSonomy offers an application programming interface (API) which allows other systems to request BibSonomy data and process it individually. The API is based on the concept of the REST-API [Fielding and Taylor, 2002], providing the typical HTTP-verbs GET, PUT, POST and DELETE to conduct different actions with the concerned URLs. For example, one can request a list of all tags of the system by visiting the URL `http:\www.bibsonomy.org/api/tags`. The URL `http://www.bibsonomy.org/api/posts?resourcetype=bookmark&search=folksonomy` would deliver entries

which contain the term *folksonomy* in the title, in the list of tags or in the description of the post. The response in XML format possibly contain user related data, such as a user name, which can be assigned to a real person. As such information is publicly available, the API is not the only way to receive such information: One can simply get user names and other data opening the systems interface in an explorer (choosing an arbitrary format such as XML) and exporting the data.

Legal Situation: The release of system related data through (registered) third parties by means of an API, can be considered data transmission. Such a transfer of partly-personal-related data is acceptable if it is part of the agreed functionality between user and provider. This is the case if the transmission has either been explicitly agreed upon or if such functionality is typical for the specific type of system and an average user knows about it. Systems whose users do not naturally expect the transmission of their data via an API need to explicitly reach an agreement with the concerned users. APIs can be seen as a central part of Web 2.0 systems to which tagging systems can be counted. If such systems additionally include the provision of publicly available content in their scope of operation (for example, BibSonomy offers different export functionalities to create publication reference lists), the transmission of such data by means of an API is permitted.

Guidelines: If an API can be considered to be part of the agreed upon system functionality, it can be provided without legal restrictions. Apart from that, service providers need an explicit user agreement or apply techniques to anonymize the data to be transferred beforehand so that no link between a user account and the data transferred via the API can be made.

10.2.7 Membership Termination

System Description: If users want to terminate their system membership, they can delete their account via the settings interface. BibSonomy then disables the account so that posts cannot be viewed by other system members or by the registered user and the user can no longer log-in to post new entries.

Legal Situation: With the membership termination, the main purpose for storing and processing personal data does not apply anymore. Inventory data therefore need to be deleted.

Aside from individual cases where specific data (for example photos or videos) need to be deleted because of copyright reasons, the further usage of content related data has to be explicitly approved by the user. Otherwise a service provider has to demonstrate justifiable interests according to § 28 BDSG and that no contradicting prevailing interests of the user exist. However, the deletion of a user account can be seen as evidence that the user wants to end his or her "personal" relationship to the provider without leaving personal related data for free use. Unless it can be guaranteed organizationally and technically that a personal relation between data and user cannot be made, content data need to be deleted. Possible user profiles related to a pseudonym also have to be deleted or irrevocably anonymized, as the termination of membership can be considered as the execution of the user's right to contradict. In case of doubt, a user will not assume that a further statement is necessary.

Guidelines: After deleting a user account, no personal data is allowed to remain in the system: they have to be deleted or – if possible – anonymized. This means that identifying attributes such as the real name, the e-mail address or, because of their controversial classification, the IP address need to be deleted. The rest can be stored individually or under

a pseudonym which cannot be traced back to the user. However, if the remaining profile still offers the possibility of identifying a user, the data have to be deleted.

10.2.8 BibSonomy as a Research Project

System Description: The social bookmarking system BibSonomy is operated by a research institution in order to put scientific findings into practice. A major research interest of tagging systems is the exploration of the specific lightweight knowledge representation, the folksonomy structure: Each user describes resources individually with arbitrary tags. By means of overlapping resources and tags, new relationships evolve between tags, users and resources. The analysis and exploitation of such relationships is essential for a number of features such as spam detection (presented in Chapter 7) or ranking (presented in Section 6.5). Further research areas concern the structuring of tagging vocabulary (for example via the automatic extraction of synonym/hyponym relations), recommendation systems to recommend tags or resources and clustering to find user communities (an overview over research in this field is given in Chapter 2).
Algorithms for the above mentioned functionalities are developed in parallel to the further implementation of the system. Often it is not obvious beforehand which data will be useful for which functionalities. For example, only after a time did it become clear that the differences between spam and legitimate e-mail addresses contribute significantly to the fast exclusion of spammers. It is likely that commercial provider experiment with data and algorithms in a similar way.
Due to various requests from other research institutions, the BibSonomy team decided to publish a benchmark dataset consisting of the public posts (i. e., content data). Such a dataset enables external researchers to use BibSonomy data for their research purposes. The user data was pseudonmized by assigning an identification number to each user name. Consequently, a direct connection from the data to a user in BibSonomy is impossible. In some constellations, however, a system user can be identified by comparing the posts in the dataset with the publish posts in BibSonomy. This is especially simple if a user has very specific tags or posts. Without appropriate techniques which change the data insofar that data can not be related to system users without an disproportionate effort, all data in the dataset need to be considered as personal data.
Researchers, interested in the benchmark data need to sign a licence agreement which, among other things, ensures that the dataset is only used for the purpose of research and not further published by the licensee.
Legal Situation: The operation of BibSonomy as a research project of a Hessian university is subject to the Hessian privacy law (HDSG), especially § 33, HDSG. This law replaces the privacy legislations of the less specific TMG for usage and inventory data, because the operation of the telemedia (the BibSonomy system) is in itself a research object and the data required from its users are analysed and processed for scientific purposes.
The scientific purpose extends the scope for which inventory and usage data can be collected and processed in contrast to the given scope defined by the regulations of the TMG that non-scientific service providers need to follow. The public entries of users (content data) do not fall under any privacy related restrictions according to § 3 Abs. 4 HDSG and can be collected and processed without restrictions as long as they are published or planned to be published by the user. As a matter of principle, the data should if possible also be anonymized and pseudomyzed in the scientific field. The collection of data without

a specific purpose is prohibited in the scientific world as well. The reason for obtaining the data need to be determined before the data is collected. It is the nature of research, however, that single research questions might change or be extended over time.

10.3 Discussion

Citizens are protected by the right to informational self-determination. Concrete regulations for content data are defined in the BDSG and for inventory- and usage data in the TMG. Especially with new Internet developments (in our case the voluntary publication of partly-personal-related data in social tagging systems), it is not always clear whether and how to protect the data and their users. This chapter demonstrated by means of an interdisciplinary analysis of the social bookmarking system BibSonomy that data privacy considering basic legal conditions is possible in social tagging systems.
Our findings show that content data can be processed without problems if the processing is necessary for the system's service (the contractual relationship between service provider and service user). Since tagging systems are based on the social interaction of many users, the publication of user entries is an essential system function. It is questionable if functionalities such as the discovery of spam or the improvement of the search process are part of the main features of a social tagging system. Likewise, the handling of inventory and usage data needs to be considered carefully. Inventory data are used for the creation and design of a contractual relationship, while usage data enables the usage of the telemedia service. Generally, the collection, storage and further processing and usage of such data is only permitted for those purposes, and only in case they are required. This has to be verified for each of the considered data types.
Different options can be considered apart from the most simple method: abstaining from collecting the data. A service provider can provide transparency by informing the user about the different functionalities offered and the scope and purpose of data collection and usage. If data is used beyond the scope and purpose contained in the contractual relationship, a provider needs to obtain a written consent from the user. An idea could be to allow users to select the functionalities for which they are willing to provide their personal information. In any case, data should be anonymized and – if no longer required – deleted.
The contemporary legal and technical activities do not consider data voluntarily and systematically published by users in Web 2.0 applications. It is difficult to foresee the consequences of such expositions. In order to protect the right to self determination in Social Web applications, one needs to consider a long term technical and legal system design which hinders users from thoughtless data publications or which enables users to delete publicly available data. Different possibilities are already being discussed: The use of different privacy levels could allow data to only be selectively visible and not generally available to all users. An automatic "clean-up" function which deletes user entries after a certain time and in agreement with the user could help to remove or hide neglected data. The incentive to implement such features, however, depends on the demand of system users as well as legal guidelines which need to be aligned to today's Social Web.

Chapter 11

Conclusion and Outlook

This thesis presented three research topics, i. e., social search, spam detection and data privacy in social bookmarking systems. Regarding social search we compared tagging systems and search engines with respect to structure, content and search behaviour. Methods and applications to prevent spam in social bookmarking systems were analysed and evaluated. Finally, data privacy aspects in social bookmarking systems were analysed in detail and a solution considering privacy issues in a data mining application presented. Each of the three research fields offer various topics for further investigation. Summaries of each topic and the main ideas for further activities will be described in the following sections.

11.1 Social Search

Social tagging systems and search engines stem from different paradigms, namely the publishing and processing of user generated contents (social bookmarking systems) versus the publishing and processing of automatically generated indices (search engines). In this thesis we compared the two systems with respect to user behaviour, structure and semantics. It could be shown that similarities between both systems exist. However, due to the different process of searching and tagging, the systems are not equivalent. The main findings of the research are:

- Frequently used tags and queries are correlated over time. People's interests are therefore reflected in the tagged resources as well as in the searched resources. Current events and trends are considered in both systems.
- Though frequent terms overlap, one can still distinguish differences in the usage of tags and search terms. For example, search terms often include URLs, while tags reflect personal classification systems, which are difficult to use for formulating a more general information need (for instance tags such as "funny" or "myown" are often used).
- Prominent resources in both systems overlap. In which system the popular web resources show up first has not been analysed.
- Clickdata presents a folksonomy like structure (a "logsonomy"), which shows similar structural characteristics as folksonomies do. As a consequence, many of the

algorithms tuned to make use of folksonomy data can be applied to logsonomies as well. Search engine provider can use such information to improve search and other services. The study has also shown, however, that logsonomies are not as coherent as folksonomies. This can be explained by the fact that a user in the logsonomy is identified by a session ID, while users of a folksonomy have a registered account and therefore a unique username [Benz et al., 2010b]. Furthermore, no semantic analysis of query terms forming compound words or URLs has been conducted. Instead, query terms have been treated like tags in so far that they were split into single terms using whitespaces as the delimiter. Finally, click data is more prone to errors than folksonomy data. Users might click a resource but then find out that the information given is irrelevant regarding their information need.

- Logsonomies built from clickdata provide inherent semantics which can be captured by appropriate data mining techniques. However, the challenges are slightly different than from folksonomy data due to a different interaction with the search system. In search engines, the user first formulates an information need and afterwards receives a list of search results. In a folksonomy, a resource is described using specific tags best representing the resource's content and context.

As our study has been conducted a few years ago, a repetition with today's system features and present-day data could yield further answers to some questions: Do results change? Is the overlap of resources in rankings even bigger? Which of the two systems can better cover long-tail queries? Do certain features introduced in social bookmarking systems (for example recommender services) and search engines (for example the query expansion tool) influence user behaviour or system semantics in a similar way?
The last part of our analysis of search engines and social bookmarking systems (see Section 6.5) focused on integrative aspects. How can both systems benefit from the data produced in each system? One concrete scenario we considered was the usage of tagging data for implicit feedback generation in order to evaluate search algorithms. The preliminary results raise new questions which are to be explored in the future:

- It has been shown that other learning-to-rank algorithms outperform Ranking SVMs. One needs to apply other methods, such as listwise approaches, to see if the performance can be improved.

- The question concerning overlap between the two systems needs to be handled in more detail. In the experiments carried out in this thesis, a URL was classified as being of interest if it contained a tag corresponding to a query term. However, as tags and query terms show differences, those differences need to be integrated into computing overlap, for instance, by finding similar tags.

- Another interesting question is if folksonomy data can be used for improving search engine evaluations. Can we better detect errors in human labeled datasets when we use the feedback of folksonomy data?

- A user study manually comparing feedback derived from clickdata and feedback derived from folksonomy data would help reveal further qualitative differences between the two approaches. Such studies could include further methods for deriving

feedback data. For example, relevance feedback has been generated by crowdsourcing activities where unknown individuals generated labels mostly in exchange for micro-payments [Kazai et al., 2013].

- In Doerfel et al. [2014] the authors present an analysis of resource and tag sharing behaviour and system usage based on a log file derived from the social bookmarking system BibSonomy. In the light of logsonomies, a comparison of logdata from click files and from social bookmarking systems would be of interest. Do properties derived from log files of a folksonomy more reflect the properties of a logsonomy or of the folksonomy, the clickdata is based on. Can the clicks provided by the folksonomy logfile be of use for improving search algorithms?

Overall, one can conclude from the findings described above that both search engines and social bookmarking systems provide a helpful way to retrieve information. While search engines allow a focused search and filter relevant information based on many factors including social and personal interests, users searching and browsing folksonomies benefit from a social network reflecting short paths and a certain degree of serendipity. Considering the overlap of searched and tagged topics as well as relevant bookmarks in rankings, one can see that both system represent current and relevant information.
As both systems provide a platform for information retrieval and sharing, they can benefit from each other. Social bookmarking systems could be improved by integrating search aspects to better find information. This includes standard algorithms from information retrieval and adjusting them to the folksonomy structure, as has been done using the FolkRank algorithm (see Section 2.5.1).
Search engines on the other hand, could benefit from integrating user interactions into the search process. This has been realised by integrating social bookmarking resources into the process of creating indices and rankings or by personalizing search results based on similar users which liked or tagged similar resources. Google, for instance, realised the importance of social networks and user interaction for their search business and introduced the social network Google+ [1] in 2011. Among other things, they use the service to personalize search. For instance, they add social annotations (a picture and name) to search results showing that the query has been shared, reviewed or submitted by one of the searcher's online acquaintances [Maureen Heymans, 2009].
Nevertheless, social activities are not limited to the posting of bookmarks. Social search today includes friendship connections on social networks such as Facebook or user generated content such as Twitter messages or Amazon reviews. It is enhanced by the integration of mobile applications such as messaging or GPS data. The social component of information retrieval can no longer be ignored. With it, however, challenges such as the detection of spam or the protection of a user's privacy need to be taken into account.

11.2 Spam Detection

In order to find spam in social bookmarking systems we introduced different features which can be induced from social bookmarking data, such as registration information,

[1] https://plus.google.com

user posts or the social network connecting user, bookmarks and tags. To get a better understanding of these features, they were each assigned to a category (profile, location, activity and semantic) and their performance using standard classification algorithms (SVM, Naive Bayes, logistic regression) was compared. The best category on its own was the one containing semantic features based on the co-occurrence network of a folksonomy. The combination of all features, however, delivered the best results as it took all available information into account. Considering classification algorithms, no significant differences could be found: support vector machines and logistic regression delivered the best results. As the BibSonomy dataset was part of the ECML/PKDD discovery challenge 2008, the participant's ideas and the results obtained from using the BibSonomy dataset were introduced. The most successful idea considered all information around a post (bookmark, tags, description) as a text document and used a text classification setting to identify spam.
The understanding of typical spam and non-spam behaviour was deepened by means of an analysis of local patterns. What kind of descriptions can be found for the two groups - spammers and non-spammers? The patterns analysed confirmed many of our previous assumptions which administrators often unconsciously applied when manually flagging spammers. For instance, spammers do not submit compound names during registration, they tend to choose a long name and relatively long e-mail addresses.
The results of the experiments carried out have been implemented and used in the spam framework in BibSonomy. In order to improve spam detection and therefore facilitate the work of the BibSonomy administrators, various interesting data mining and social web ideas can be considered for the future:

- Spam detection on the post level instead of on the user level is an interesting option to improve the accuracy of spam detection algorithms. This would mean that only individual posts are marked as such, whereas currently all or none of the posts of a user are marked as spam. "The justification for detecting spammers on a post level is that users either have a malicious intention and use non-spam posts to hide their motivations or are legitimate users." [Krause et al., 2008c] First experiments considering this have been conducted by Yang and Lee [2011b]. They measure the similarity between tags of a post and the description of a referenced website. However, they use the BibSonomy ECML/PKDD dataset introduced in Section 7.2. Spam in this dataset has been marked on a user-level, so that differences in user posts indicating spam and non-spam are not considered.

- Using online-updateable algorithms can help improve performance and accuracy. Different variants of this approach exist. One idea is to treat spam data as a stream. As soon as a user is wrongly classified by the classification algorithm, the classification model is updated to consider the mistake. [Sculley, 2008], for example, introduced an SVM variant called Relaxed Online SVM to filter spam e-mail messages online.

- The set of features used for spam detection could be enhanced by using features derived from data of social networks and user participation. This includes the results of the user button to identify spam, the exploration of the "friends" and "followers" networks or the number of bookmark clicks from spammers or non-spammers.

- While we focused on analyzing different features, a deeper investigation of algorithmic details might be helpful to better tackle the problem. For example, the imple-

mentation of a kernel adjusted to the problem of spam might improve spam classification. Keeping in mind, that the best classifier of the ECML PKDD challenge used textual classification, string kernels [Amayri and Bouguila, 2010; Sonnenburg et al., 2007] could be of great interest.

- Considering the growing size of the BibSonomy data it will be necessary to preselect helpful training examples. Pre-selection techniques can be built on simple assumptions such as that the most current examples are the most informative ones or that wrongly classified examples should be considered. Another strategy can be to preselect training examples using machine learning techniques. For instance, active learning approaches (see for example in Settles [2009]) chose uncertain examples which are then manually labeled and used for building a classification model.

Beyond spam detection in social bookmarking systems, the identification and detection of spam in all kinds of Social Web and mobile activities will become an essential task in the future. The type of spam is likely to not change very much (we will still be offered loans, diet products or adult content), however, the way of distribution will be simplified: The wealth of social applications offer a variety of different distribution channels. As most of those applications obtain a profit when they reach a certain amount of (active) users, entry barriers to use them will remain low. This also holds true for spam users. Finally, as many of the Web 2.0 applications are connected, spammers benefit from spamming in one of the systems highlighting the visibility in other systems. For example, spammers build so called link farms in social networks by obtaining as many followers as possible to their spam site or tweet. This also helps improve the rank of their fraudulent page in search engines. Thus, the design of appropriate features and the further development of algorithms to detect spam will remain critical in order to preserve the most important aspect of Web 2.0 and future applications: Social user interaction based on trust.

11.3 Data Privacy

With the arrival of Web 2.0 applications, the volumes of stored information about internet users grew faster and faster. While this seems to be a big chance to improve data mining and profiling applications, many internet users and researchers worry about the consequences of publishing and collecting data, especially personal data. This concern was discussed in the third part of this thesis.
In Germany, the “right to determine in principle the disclosure and use of personal data” [Fischer-Hübner et al., 2011] has been established in the context of the German constitutional ruling as the right to informational self-determination. Based on this principle, regulations in Germany and Europe (other parts of the world are not discussed in this thesis) have been formulated to protect a user’s data privacy.
In the light of today’s digital world, where many free Web 2.0 services including search engines and social networks benefit from personal user information, an unbalance between data protection and data usage can be observed. The legal framework formulated in Germany and the EU does not fully consider today’s implementation, collection and usage of private information through online applications. Also, companies and other institutions collecting personal data in an international and digital environment ignore the basic principles of protecting their user’s data privacy.

This thesis analysed the collection, processing and further circulation of possible private data in the social bookmarking system BibSonomy. Different system features were assessed according to the German law (in 2010) and recommendations concerning how to improve and maintain user privacy protection were given (Chapter 10).
The design of features for data mining services respecting data privacy laws was investigated in Chapter 8. Features used for the detection of spam were assigned different privacy levels. The results showed that the identification of spammers using privacy - friendly levels is possible when the user account shared enough "public" information (such as public posts) in the social bookmarking system.
The discussion regarding how to preserve a user's right to informational self-determination and protect user data privacy in the Social Web needs to continue. Collecting and linking user data from many different sources such as the Web, mobile applications or radio-frequency identification is more and more common, so that the topic of data privacy has become even more relevant. Stock market-listed companies such as Amazon or Facebook rely on the profiles generated from their user databases for advertising purposes. They are keen on finding out as much as they can about their users' preferences and interests. However, not only commercial interests trigger the collection, processing and distribution of user data. The recently exposed actions by several nations' intelligence services show that private user data has become a target for use beyond marketing and sales activities.
In order to counter the digital monitoring of citizens by companies, governments and other interested parties, adjustments to the current legislation in Europe are being discussed. On January 25th 2012, the EU Commission proposed an overhaul of the Data Protection Directive 95/46/EC [Hornung, 2012]. Among other things, the reform could strengthen the rights of the individuals. For instance, the draft includes the "right to be forgotten" (Art. 17), whereby institutions collecting personal data need to delete a concerned person's private data after his or her request and inform third parties which process the data that the person requested the deletion of "any links to, or copy or replication of that personal data" [European Commission, 2012]. However, the example shows the difficulty with which such rights can be put into practice: The wording contains vague expressions such as "*reasonable steps*" concerning how third parties need to be advised of the data erasure: "To ensure this information, the controller should take all reasonable steps, including technical measures, in relation to data for the publication of which the controller is responsible [European Commission, 2012]." This allows room for interpretation. Considering today's decentralized digital infrastructure, it will be technically difficult to control the distribution of published data. Additionally, the erasure of such data itself leads to technical challenges such as dealing with database backups. In 2014, however, the European court confirmed the "right to be forgotten", stating that search engines have to remove data which is "inaccurate, inadequate, irrelevant or excessive for the purposes of the data processing [European Commission, 2015]" from their search result lists when requested by concerned users. The search engine Google has introduced a service which deletes links upon justified request [Google Support, 2015].
Another aspect of the discussion regarding privacy in digital applications is the usability of privacy preserving features. The emergence of the mobile application "WhatsApp" [2] is an example. In order to connect as many users as possible to the system and enable a simple start, the application collects phone entries from the mobile device's address

[2] `http://www.whatsapp.com/`

book and links those numbers to other user numbers already stored on the "WhatsApp' servers. The advantage is that a new user gets easily connected to acquaintances who have already registered. The disadvantage is that WhatsApp collects phone numbers even if their owners do not use the application. In contrast, similar applications such as Kik [3] do not require their users to transmit phone numbers. As users of such privacy-aware applications need to enter new contacts manually, the effort to become linked is much higher. Membership figures show[4] that the simple way still seems to attract more users – though they have to renounce to part of their privacy, namely their phone contacts.
As with every technical advancement of society, there are positive and negative aspects. Mostly, the benefits outweigh the downsides. This is definitely the case for Social Web applications. Those applications have become part of our daily lives. It is difficult to imagine a world without digital social networks, tagging systems or news from microblogging systems. Now it is time to educate users regarding the use of the Web, starting with school children. Only the users have the power to force the implementation of better data privacy laws and processes.

[3] `http://kik.com/about/`

[4] 100 million users registered with Kik, half of them based in the United States [Silcoff, 2013] versus 400 million active users each month in WhatsApp [Newton, 2013]

Bibliography

Fabian Abel. *Contextualization, user modeling and personalization in the social web: from social tagging via context to cross-system user modeling and personalization.* PhD thesis, University of Hanover, 2011. URL http://edok01.tib.uni-hannover.de/edoks/e01dh11/660718537.pdf. http://d-nb.info/1014252423.

Fabian Abel, Nicola Henze, and Daniel Krause. Ranking in Folksonomy Systems: can context help? In James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz, Key-Sun Choi, and Abdur Chowdhury, editors, *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 1429–1430. ACM, 2008. ISBN 978-1-59593-991-3.

Lada Adamic. Zipf, power-laws, and pareto – a ranking tutorial. http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html, 2002. URL http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html.

Eytan Adar. User 4xxxxx9: Anonymizing query logs. In *Query Logs Workshop at WWW2006*, 2007.

B. Thomas Adler, Luca de Alfaro, Santiago M. Mola-Velasco, Paolo Rosso, and Andrew G. West. Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features. In Alexander F. Gelbukh, editor, *CICLing (2)*, volume 6609 of *Lecture Notes in Computer Science*, pages 277–288, Tokyo, Japan, February 2011. Springer. ISBN 978-3-642-19436-8. URL http://repository.upenn.edu/cgi/viewcontent.cgi?article=1494&context=cis_papers.

Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17:734–749, June 2005. ISSN 1041-4347. doi: 10.1109/TKDE.2005.99. URL http://dx.doi.org/10.1109/TKDE.2005.99.

Eugene Agichtein, Eric Brill, Susan Dumais, and Robert Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: http://doi.acm.org/10.1145/1148170.1148175. URL http://portal.acm.org/citation.cfm?id=1148175.

Hend S. Al-Khalifa and Hugh C. Davis. Towards better understanding of folksonomic patterns. In *Proceedings of the eighteenth conference on Hypertext and hypermedia*, HT '07, pages 163–166, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-820-6. doi: 10.1145/1286240.1286288. URL http://doi.acm.org/10.1145/1286240.1286288.

Abdullah Almaatouq, Ahmad Alabdulkareem, Mariam Nouh, Erez Shmueli, Mansour Alsaleh, Vivek K. Singh, Abdulrahman Alarifi, Anas Alfaris, and Alex (Sandy) Pentland. Twitter: Who gets caught? observed trends in social micro-blogging spam. In *Proceedings of the 2014 ACM Conference on Web Science*, WebSci '14, pages 33–41, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2622-3. doi: 10.1145/2615569.2615688. URL http://doi.acm.org/10.1145/2615569.2615688.

Ola Amayri and Nizar Bouguila. A study of spam filtering using support vector machines. *Artificial Intelligence Review*, 34:73–108, 2010. ISSN 0269-2821. doi: 10.1007/s10462-010-9166-x. URL http://dx.doi.org/10.1007/s10462-010-9166-x.

Pierre Andrews, Ilya Zaihrayeu, and Juan Pane. A classification of semantic annotation systems. *Semant. web*, 3(3):223–248, August 2012. ISSN 1570-0844. doi: 10.3233/SW-2011-0056. URL http://dx.doi.org/10.3233/SW-2011-0056.

Karen Angel. *Inside Yahoo!: Reinvention and the Road Ahead*. John Wiley & Sons, Inc., Berlin – Heidelberg, 2002.

UN General Assembly. Guidelines for the regulation of computerized personal data files. Available at:http://www.unhcr.org/refworld/docid/3ddcafaac.html, December 1990. URL http://www.unhcr.org/refworld/docid/3ddcafaac.html.

Martin Atzmüller. *Knowledge-intensive subgroup mining : techniques for automatic and interactive discovery*. PhD thesis, Berlin; Amsterdam, 2007. URL http://www.amazon.com/Knowledge-Intensive-Subgroup-Mining-Dissertations-Diski-Dissertations/dp/1586037269.

Martin Atzmueller. *Knowledge-Intensive Subgroup Mining – Techniques for Automatic and Interactive Discovery*, volume 307 of *Dissertations in Artificial Intelligence-Infix (Diski)*. IOS Press, March 2007.

Martin Atzmueller, Florian Lemmerich, Beate Krause, and Andreas Hotho. Spammer discrimination: Discovering local patterns for concept characterization and description. In Johannes Fürnkranz and Arno Knobbe, editors, *Proc. LeGo-09: From Local Patterns to Global Models, Workshop at the 2009 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Bled, Slovenia, September 2009. accepted.

Ching Man Au Yeung, Nicholas Gibbins, and Nigel Shadbolt. Web search disambiguation by collaborative tagging. In *Proceedings of the Workshop on Exploiting Semantic Annotations in Information Retrieval (ESAIR 2008), co-located with ECIR 2008, Glasgow, United Kingdom, 31 March, 2008*, pages 48–61, 2008. URL http://eprints.ecs.soton.ac.uk/15393/.

Peter Bailey, Nick Craswell, Ian Soboroff, Paul Thomas, Arjen P. de Vries, and Emine Yilmaz. Relevance assessment: are judges exchangeable and does it matter. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 667–674, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390447. URL http://doi.acm.org/10.1145/1390334.1390447.

Shenghua Bao, Guirong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In *Proc. WWW '07*, pages 501–510, Banff, Canada, 2007.

Judit Bar-Ilan, Mazlita Mat-Hassan, and Mark Levene. Methods for comparing rankings of search engine results. *Comput. Networks*, 50(10):1448–1463, 2006. URL http://portal.acm.org/citation.cfm?id=1148375#.

Albert-Laesli Barabasi and Röka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

Albert-Laesli Barabasi and E. Bonabeau. Scale-free networks. *Scientific American*, 288 (60-69), 2003. URL http://www.nd.edu/~networks/PDF/Scale-Free%20Sci%20Amer%20May03.pdf.

Susan B. Barnes. A privacy paradox: Social networking in the united states. *First Monday*, 11(9), 2006. ISSN 13960466. URL http://firstmonday.org/ojs/index.php/fm/article/view/1394.

Fabiano M. Belém, Eder F. Martins, Jussara M. Almeida, and Marcos A. Gonçalves. Personalized and object-centered tag recommendation methods for web 2.0 applications. *Information Processing & Management*, 50(4):524 – 553, 2014. ISSN 0306-4573. doi: http://dx.doi.org/10.1016/j.ipm.2014.03.002. URL http://www.sciencedirect.com/science/article/pii/S0306457314000181.

Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. Detecting spammers on Twitter. In *Proceedings of the Seventh Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, July 2010.

Dominik Benz, Folke Eisterlehner, Andreas Hotho, Robert Jäschke, Beate Krause, and Gerd Stumme. Managing publications and bookmarks with bibsonomy. In Ciro Cattuto, Giancarlo Ruffo, and Filippo Menczer, editors, *HT '09: Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, pages 323–324, New York, NY, USA, June 2009a. ACM. ISBN 978-1-60558-486-7. doi: 10.1145/1557914.1557969. URL http://portal.acm.org/citation.cfm?doid=1557914.1557969#.

Dominik Benz, Beate Krause, G. Praveen Kumar, Andreas Hotho, and Gerd Stumme. Characterizing semantic relatedness of search query terms. In *Proceedings of the 1st Workshop on Explorative Analytics of Information Networks (EIN2009)*, Bled, Slovenia, September 2009b.

Dominik Benz, Andreas Hotho, Robert Jäschke, Beate Krause, Folke Mitzlaff, Christoph Schmitz, and Gerd Stumme. The social bookmark and publication management system

bibsonomy. *The VLDB Journal*, 19(6):849–875, December 2010a. ISSN 1066-8888. doi: 10.1007/s00778-010-0208-4. URL http://www.kde.cs.uni-kassel.de/pub/pdf/benz2010social.pdf.

Dominik Benz, Andreas Hotho, Robert Jäschke, Beate Krause, and Gerd Stumme. Query logs as folksonomies. *Datenbank-Spektrum*, 10(1):15–24, 2010b. ISSN 1618-2162. doi: 10.1007/s13222-010-0004-8. URL http://dx.doi.org/10.1007/s13222-010-0004-8.

Smriti Bhagat, Graham Cormode, Balachander Krishnamurthy, and Divesh Srivastava. Class-based graph anonymization for social network data. *Proc. VLDB Endow.*, 2: 766–777, August 2009. ISSN 2150-8097. URL http://portal.acm.org/citation.cfm?id=1687627.1687714.

Claudio Biancalana, Fabio Gasparetti, Alessandro Micarelli, and Giuseppe Sansonetti. Social semantic query expansion. *ACM Trans. Intell. Syst. Technol.*, 4(4):60:1–60:43, October 2013. ISSN 2157-6904. doi: 10.1145/2508037.2508041. URL http://doi.acm.org/10.1145/2508037.2508041.

Kerstin Bischoff, Claudiu S. Firan, Cristina Kadar, Wolfgang Nejdl, and Raluca Paiu. Automatically identifying tag types. In *Proceedings of the 5th International Conference on Advanced Data Mining and Applications*, ADMA '09, pages 31–42, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-03347-6. doi: 10.1007/978-3-642-03348-3_7. URL http://dx.doi.org/10.1007/978-3-642-03348-3_7.

Enrico Blanzieri and Anton Bryl. A survey of learning-based techniques of email spam filtering. *Artif. Intell. Rev.*, 29:63–92, March 2008. ISSN 0269-2821. doi: 10.1007/s10462-009-9109-6. URL http://portal.acm.org/citation.cfm?id=1612711.1612715.

Toine Bogers and Antal Van den Bosch. Using language modeling for spam detection in social reference manager websites. In Robin Aly, Claudia Hauff, I. den Hamer, Djoerd Hiemstra, Theo Huibers, and Franciska de Jong, editors, *Proceedings of the 9th Belgian-Dutch Information Retrieval Workshop (DIR 2009)*, pages 87–94, Enschede, February 2009.

Katrin Borchert. Active learning for spam detection in the social bookmarking system Bibsonomy. Master's thesis, University of Würzburg, 2011.

Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. Sopra: A new social personalized ranking function for improving web search. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 861–864, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484131. URL http://doi.acm.org/10.1145/2484028.2484131.

Karin. Breitman and Marco Antonio. Casanova. *Semantic Web:Concepts, Technologies and Applications*. Springer-Verlag London Limited, New York, 2007. ISBN 9781846285813 184628581X 9781846287107 1846287103. URL http://www.worldcat.org/search?qt=worldcat_org_all&q=9781846285813.

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. URL http://citeseer.ist.psu.edu/brin98anatomy.html.

Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36:3–10, September 2002. ISSN 0163-5840. doi: 10.1145/792550.792552. URL http://doi.acm.org/10.1145/792550.792552.

Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: http://doi.acm.org/10.1145/1102351.1102363. URL http://portal.acm.org/citation.cfm?id=1102351.1102363.

Yuanzhe Cai, Miao Zhang, Dijun Luo, Chris Ding, and Sharma Chakravarthy. Low-order tensor decompositions for social tagging recommendation. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 695–704, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0493-1. doi: 10.1145/1935826.1935920. URL http://doi.acm.org/10.1145/1935826.1935920.

Iván Cantador, Alejandro Bellogín, and David Vallet. Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 237–240, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-906-0. doi: 10.1145/1864708.1864756. URL http://doi.acm.org/10.1145/1864708.1864756.

Mark J. Carman, Mark Baillie, Robert Gwadera, and Fabio Crestani. A statistical comparison of tag and query logs. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 123–130, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: http://doi.acm.org/10.1145/1571941.1571965. URL http://portal.acm.org/citation.cfm?id=1571965&dl=GUIDE&coll=GUIDE&CFID=50799114&CFTOKEN=52331089.

Ciro Cattuto, Vittorio Loreto, and Luciano Pietronero. Collaborative tagging and semiotic dynamics. *CoRR*, abs/cs/0605015, 2006. URL http://dblp.uni-trier.de/db/journals/corr/corr0605.html#abs-cs-0605015.

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. ISSN 0921-7126. URL http://www.kde.cs.uni-kassel.de/stumme/papers/2007/cattuto2007network.pdf.

Ciro Cattuto, Dominik Benz, Andreas Hotho, and Gerd Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In Amit P. Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy W. Finin, and Krishnaprasad Thirunarayan, editors, *The Semantic Web – ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 615–631, Berlin/Heidelberg, 2008. Springer. ISBN 978-3-540-88563-4. doi: 10.1007/978-3-540-88564-1_39. URL http://cxnets.googlepages.com/cattuto_iswc2008.pdf.

Abhijnan Chakraborty, Saptarshi Ghosh, and Niloy Ganguly. Detecting overlapping communities in folksonomies. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media*, HT '12, pages 213–218, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1335-3. doi: 10.1145/2309996.2310032. URL http://doi.acm.org/10.1145/2309996.2310032.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chris Chatfield. *The analysis of time series: an introduction*. CRC Press, Florida, US, 6th edition, 2004.

Feilong Chen, Pang-Ning Tan, and Anil K. Jain. A co-classification framework for detecting web spam and spammers in social media web sites. In David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy J. Lin, editors, *CIKM*, pages 1807–1810. ACM, 2009a. ISBN 978-1-60558-512-3. URL http://dblp.uni-trier.de/db/conf/cikm/cikm2009.html#ChenTJ09.

Jilin Chen, Werner Geyer, Casey Dugan, Michael Muller, and Ido Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI '09, pages 201–210, New York, NY, USA, 2009b. ACM. ISBN 978-1-60558-246-7. doi: 10.1145/1518701.1518735. URL http://doi.acm.org/10.1145/1518701.1518735.

Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 137–144, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: http://doi.acm.org/10.1145/1102351.1102369. URL http://doi.acm.org/10.1145/1102351.1102369.

European Commission, Freedom Directorate General Justice, and Security. Comparative study on different approaches to new privacy challenges, in particular in the light of technological developments, 2010.

Oscar Corcho, Andrés García-Silva, and Iván Cantador. Enabling folksonomies for knowledge extraction: A semantic grounding approach. *Int. J. Semant. Web Inf. Syst.*, 8(3): 24–41, July 2012. ISSN 1552-6283. doi: 10.4018/jswis.2012070102. URL http://dx.doi.org/10.4018/jswis.2012070102.

Gordon V. Cormack. Email spam filtering: A systematic review. *Found. Trends Inf. Retr.*, 1:335–455, April 2008. URL http://portal.acm.org/citation.cfm?id=1454707.1454708.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3): 273–297, September 1995. ISSN 0885-6125. doi: 10.1023/A:1022627411411. URL http://dx.doi.org/10.1023/A:1022627411411.

David Cossock and Tong Zhang. Subset ranking using regression. In Gabor Lugosi and Hans Simon, editors, *Learning Theory*, volume 4005 of *Lecture Notes in Computer Science*, pages 605–619. Springer, Berlin / Heidelberg, 2006. ISBN 978-3-540-35294-5. doi: 10.1007/11776420_44. URL http://dx.doi.org/10.1007/11776420_44.

W.B. Croft, D. Metzler, and T. Strohmann. *Search Engines: Information Retrieval in Practice*. Pearson, London, England, 2010.

Cäline Van Damme, Martin Hepp, and Katharina Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. In *Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007)*, pages 57–70, 2007. URL http://www.kde.cs.uni-kassel.de/ws/eswc2007/proc/FolksOntology.pdf.

George Danezis. Inferring privacy policies for social networking services. In *Proceedings of the 2nd ACM workshop on Security and artificial intelligence*, AISec '09, pages 5–10, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-781-3. doi: 10.1145/1654988.1654991. URL http://doi.acm.org/10.1145/1654988.1654991.

Klaas Dellschaft and Steffen Staab. An epistemic dynamic model for tagging systems. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, HT '08, pages 71–80, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-985-2. doi: 10.1145/1379092.1379109. URL http://doi.acm.org/10.1145/1379092.1379109.

Klaas Dellschaft and Steffen Staab. On differences in the tagging behavior of spammers and regular users. In *Proceedings of the Web Science Conference 2010*, 2010.

E. U. Directive. 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of such Data. *Official Journal of the EC*, 23, 1995.

Endang Djuana, Yue Xu, Yuefeng Li, and Audun Jøsang. A combined method for mitigating sparsity problem in tag recommendation. In *47th Hawaii International Conference on System Sciences, HICSS 2014, Waikoloa, HI, USA, January 6-9, 2014*, pages 906–915, 2014. doi: 10.1109/HICSS.2014.120. URL http://dx.doi.org/10.1109/HICSS.2014.120.

Stephan Doerfel, Robert Jäschke, Andreas Hotho, and Gerd Stumme. Leveraging publication metadata and social data into folkrank for scientific publication recommendation. In *Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the*

Social Web, RSWeb '12, pages 9–16, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1638-5. doi: 10.1145/2365934.2365937. URL http://doi.acm.org/10.1145/2365934.2365937.

Stephan Doerfel, Andreas Hotho, Aliye Kartal-Aydemir, Alexander Roßnagel, and Gerd Stumme. *Informationelle Selbstbestimmung im Web 2.0 — Chancen Und Risiken sozialer Verschlagwortungssysteme*. Vieweg + Teubner Verlag, 2013. ISBN 9783642380556 3642380557. URL http://www.worldcat.org/search?qt=worldcat_org_all&q=9783642380556.

Stephan Doerfel, Daniel Zöller, Philipp Singer, Thomas Niebler, Andreas Hotho, and Markus Strohmaier. Of course we share! testing assumptions about social tagging systems. *CoRR*, abs/1401.0629, 2014. URL http://arxiv.org/abs/1401.0629.

Renato Domínguez García, Matthias Bender, Mojisola Anjorin, Christoph Rensing, and Ralf Steinmetz. Freset: an evaluation framework for folksonomy-based recommender systems. In *Proceedings of the 4th ACM RecSys workshop on Recommender systems and the social web*, RSWeb '12, pages 25–28, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1638-5. doi: 10.1145/2365934.2365939. URL http://doi.acm.org/10.1145/2365934.2365939.

Xishuang Dong, Xiaodong Chen, Yi Guan, Zhiming Yu, and Sheng Li. An overview of learning to rank for information retrieval. In Mark Burgin, Masud H. Chowdhury, Chan H. Ham, Simone A. Ludwig, Weilian Su, and Sumanth Yenduri, editors, *CSIE (3)*, pages 600–606. IEEE Computer Society, 2009. ISBN 978-0-7695-3507-4. URL http://dblp.uni-trier.de/db/conf/csie/csie2009-3.html#DongCGYL09.

Zhicheng Dou, Ruihua Song, Xiaojie Yuan, and Ji-Rong Wen. Are click-through data adequate for learning web search rankings? In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 73–82, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-991-3. doi: 10.1145/1458082.1458095.

Harris Drucker, Donghui Wu, and Vladimir Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

Georges Dupret, Vanessa Murdock, and Benjamin Piwowarski. Web search engine evaluation using clickthrough data and a user model. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, Banff, Canada, 2007.

Patrick Van Eecke and Maarten Truyens. Privacy and social networks. *Computer Law & Security Review*, 26(5):535 – 546, 2010. URL http://www.sciencedirect.com/science/article/B6VB3-5148KHG-7/2/3f622c9ebdda28400b2be86d8212d15c.

Folke Eisterlehner, Andreas Hotho, and Robert Jäschke, editors. *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497 of *CEUR-WS.org*, September 2009. URL http://ceur-ws.org/Vol-497.

European Commission. REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), 2012.

European Commission. Factsheet on the "right to be forgotten" ruling (c-131/12), 2015. Viewed: 08.04.2015, Available: http://ec.europa.eu/justice/data-protection/files/factsheets/factsheet_data_protection_en.pdf.

Brynn M. Evans and Ed H. Chi. An elaborated model of social search. *Inf. Process. Manage.*, 46:656–678, November 2010. ISSN 0306-4573. doi: http://dx.doi.org/10.1016/j.ipm.2009.10.012. URL http://dx.doi.org/10.1016/j.ipm.2009.10.012.

Lujun Fang and Kristen LeFevre. Privacy wizards for social networking sites. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 351–360, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772727. URL http://doi.acm.org/10.1145/1772690.1772727.

Reza Farahbakhsh, Xiao Han, Ángel Cuevas, and Noël Crespi. Analysis of publicly disclosed information in facebook profiles. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pages 699–705, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2240-9. doi: 10.1145/2492517.2492625. URL http://doi.acm.org/10.1145/2492517.2492625.

T. Fawcett. Roc graphs: Notes and practical considerations for researchers, 2004. URL citeseer.ist.psu.edu/fawcett04roc.html.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998a. ISBN 978-0-262-06197-1.

Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998b.

Roy T. Fielding and Richard N. Taylor. Principled design of the modern web architecture. *ACM Trans. Internet Technol.*, 2(2):115–150, May 2002. ISSN 1533-5399. doi: 10.1145/514183.514185. URL http://doi.acm.org/10.1145/514183.514185.

Simone Fischer-Hübner. *IT-Security and Privacy - Design and Use of Privacy-Enhancing Security Mechanisms*, volume 1958 of *Lecture Notes in Computer Science*. Springer, 2001. ISBN 3-540-42142-4.

Simone Fischer-Hübner, Chris Jay Hoofnagle, Ioannis Krontiris, Kai Rannenberg, and Michael Waidner. Online privacy: Towards informational self-determination on the internet (dagstuhl perspectives workshop 11061). *Dagstuhl Manifestos*, 1(1):1–20, 2011. URL http://dblp.uni-trier.de/db/journals/dagstuhl-manifestos/dagstuhl-manifestos1.html#Fischer-HubnerHKRW11.

Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23:147–168, April 2005. ISSN 1046-8188. doi: 10.1145/1059981.1059982. URL http://doi.acm.org/10.1145/1059981.1059982.

Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

LeylaJael Garcia-Castro, Martin Hepp, and Alexander Garcia. Tags4tags: Using tagging to consolidate tags. In SouravS. Bhowmick, Josef Küng, and Roland Wagner, editors, *Database and Expert Systems Applications*, volume 5690 of *Lecture Notes in Computer Science*, pages 619–628. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-03572-2. doi: 10.1007/978-3-642-03573-9_52. URL http://dx.doi.org/10.1007/978-3-642-03573-9_52.

R. Stuart Geiger and Aaron Halfaker. When the levee breaks: Without bots, what happens to wikipedia's quality control processes? In *Proceedings of the 9th International Symposium on Open Collaboration*, WikiSym '13, pages 6:1–6:6, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1852-5. doi: 10.1145/2491055.2491061. URL http://doi.acm.org/10.1145/2491055.2491061.

Jonathan Gemmell, Andriy Shepitsen, Bamshad Mobasher, and Robin Burke. Personalizing navigation in folksonomies using hierarchical tag clustering. In *Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery*, DaWaK '08, pages 196–205, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-85835-5. doi: 10.1007/978-3-540-85836-2_19. URL http://dx.doi.org/10.1007/978-3-540-85836-2_19.

Jonathan Gemmell, Thomas R. Schimoler, Laura Christiansen, and Bamshad Mobasher. Improving folkrank with item-based collaborative filtering. In *ACM RecSys'09 Workshop on Recommender Systems and the Social Web*, New York, USA, October 2009.

Jonathan Gemmell, Thomas Schimoler, Bamshad Mobasher, and Robin Burke. Resource recommendation in social annotation systems: A linear-weighted hybrid approach. *J. Comput. Syst. Sci.*, 78(4):1160–1174, July 2012. ISSN 0022-0000. doi: 10.1016/j.jcss.2011.10.006. URL http://dx.doi.org/10.1016/j.jcss.2011.10.006.

Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Resource recommendation in social annotation systems based on user partitioning. In Martin Hepp and Yigal Hoffner, editors, *E-Commerce and Web Technologies*, volume 188 of *Lecture Notes in Business Information Processing*, pages 101–112. Springer International Publishing, 2014. ISBN 978-3-319-10490-4. doi: 10.1007/978-3-319-10491-1_11. URL http://dx.doi.org/10.1007/978-3-319-10491-1_11.

Xiubo Geng, Tao Qin, Tie-Yan Liu, and Xue-Qi Cheng. A noise-tolerant graphical model for ranking. *Inf. Process. Manage.*, 48(2):374–383, March 2012. ISSN 0306-4573. doi: 10.1016/j.ipm.2011.11.003. URL http://dx.doi.org/10.1016/j.ipm.2011.11.003.

Fredric C. Gey. Inferring probability of relevance using the method of logistic regression. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 222–231, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X. URL http://dl.acm.org/citation.cfm?id=188490.188560.

Rumi Ghosh, Tawan Surachawala, and Kristina Lerman. Entropy-based classification of 'retweeting' activity on twitter. *CoRR*, abs/1106.0346, 2011a. URL http://dblp.uni-trier.de/db/journals/corr/corr1106.html#abs-1106-0346.

Saptarshi Ghosh, Pushkar Kane, and Niloy Ganguly. Identifying overlapping communities in folksonomies or tripartite hypergraphs. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 39–40, New York, NY, USA, 2011b. ACM. ISBN 978-1-4503-0637-9. doi: 10.1145/1963192.1963213. URL http://doi.acm.org/10.1145/1963192.1963213.

Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the twitter social network. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 61–70, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187846. URL http://doi.acm.org/10.1145/2187836.2187846.

Scott Golder and Bernardo A. Huberman. The structure of collaborative tagging systems, August 2005.

Scott Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. *Journal of Information Sciences*, 32(2):198–208, April 2006a. URL http://.hpl.hp.com/research/idl/papers/tags/index.html.

Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 32:198–208, April 2006b. ISSN 0165-5515. doi: 10.1177/0165551506062337. URL http://dl.acm.org/citation.cfm?id=1119738.1119747.

Google Support. Remove information from google, 2015. Viewed: 08.04.2015, Available: https://support.google.com/websearch/troubleshooter/3111061?hl=en.

Alan Gray and Mads Haahr. Personalised, collaborative spam filtering. In *IN PROCEEDINGS OF THE FIRST CONFERENCE ON EMAIL AND ANTI-SPAM (CEAS*, 2004.

Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, WPES '05, pages 71–80, New York, NY, USA, 2005. ACM. ISBN 1-59593-228-3. doi: 10.1145/1102199.1102214. URL http://doi.acm.org/10.1145/1102199.1102214.

Seda Gürses and Bettina Berendt. The social web and privacy: Practices, reciprocity and conflict detection in social networks. In Francesco Bonchi and Elena Ferrari, editors,

Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques. Chapman and Hall/CRC Press, 2010.

Ziyu Guan, Can Wang, Jiajun Bu, Chun Chen, Kun Yang, Deng Cai, and Xiaofei He. Document recommendation in social tagging services. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 391–400, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772731. URL http://doi.acm.org/10.1145/1772690.1772731.

Qing Guo, Wenfei Liu, Yuan Lin, and Hongfei Lin. Query expansion based on user quality in folksonomy. In Yuexian Hou, Jian-Yun Nie, Le Sun, Bo Wang, and Peng Zhang, editors, *Information Retrieval Technology*, volume 7675 of *Lecture Notes in Computer Science*, pages 396–405. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-35340-6. doi: 10.1007/978-3-642-35341-3_35. URL http://dx.doi.org/10.1007/978-3-642-35341-3_35.

Manish Gupta, Rui Li, Zhijun Yin, and Jiawei Han. Survey on social tagging techniques. *SIGKDD Explor. Newsl.*, 12:58–72, November 2010. ISSN 1931-0145. doi: 10.1145/1882471.1882480. URL http://doi.acm.org/10.1145/1882471.1882480.

Serge Gutwirth, Sjaak Nouwt, Yves Poullet, Paul Hert, and Cécile Terwangne. Reinventing data protection?, 2009. URL http://public.eblib.com/EBLPublic/PublicView.do?ptiID=450651.

Thiago S. Guzella and Walmir M. Caminhas. Review: A review of machine learning approaches to spam filtering. *Expert Syst. Appl.*, 36:10206–10222, September 2009. ISSN 0957-4174. doi: 10.1016/j.eswa.2009.02.037. URL http://dl.acm.org/citation.cfm?id=1539049.1539460.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11 (1):10–18, November 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278. URL http://doi.acm.org/10.1145/1656274.1656278.

Harry Halpin, Valentin Robu, and Hana Shepard. The dynamics and semantics of collaborative tagging. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW'06)*, pages 211–220, 2006.

Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 211–220, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242602. URL http://dx.doi.org/10.1145/1242572.1242602.

Seungyeop Han, Yong Y. Ahn, Sue Moon, and Hawoong Jeong. Collaborative blog spam filtering using adaptive percolation search. In *InWWW 2006 Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics*, 2006.

Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal regression. In *In International Conference on Artificial Neural Networks*, pages 97–102, 1999.

P. Heymann, Georgia Koutrika, and Hector Garcia-Molina. Fighting spam on social web sites: A survey of approaches and future challenges. *IEEE Internet Computing*, 11:36–45, November 2007. URL http://portal.acm.org/citation.cfm?id=1304062.1304547.

Paul Heymann, Georgia Koutrika, and Hector Molina. Can social bookmarking improve web search? In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 195–206, Palo Alto, California, USA, 2008. ACM. ISBN 978-1-59593-927-9. URL http://dx.doi.org/10.1145/1341531.1341558.

Maureen Heymans. Introducing google social search: I finally found my friend's new york blog!, 2009. URL http://googleblog.blogspot.de/2009/10/introducing-google-social-search-i.html.

Thomas Hoeren. Internetrecht, 2010. URL http://www.uni-muenster.de/Jura.itm/hoeren/materialien/Skript/Skript_Internetrecht_September202010.pdf. P. 419 et seq. Available at: http://www.uni-muenster.de/Jura.itm/hoeren/materialien/Skript/Skript_Internetrecht_September202010.pdf.

Gerrit Hornung. Eine Datenschutz-Grundverordnung für Europa? *Zeitschrift für Datenschutz*, 2(3):99–106, 2012.

Gerrit Hornung and Christoph Schnabel. Data protection in Germany I: The population census decision and the right to informational self-determination. *Computer Law amp; Security Review*, 25(1):84 – 88, 2009. ISSN 0267-3649. doi: 10.1016/j.clsr.2008.11.002. URL http://www.sciencedirect.com/science/article/pii/S0267364908001660.

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*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, 2006a. Springer. URL http://.kde.cs.uni-kassel.de/hotho.

Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. BibSonomy: A Social Bookmark and Publication Sharing System. In Aldo de Moor, Simon Polovina, and Harry Delugach, editors, *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, Aalborg, Denmark, July 2006b. Aalborg University Press.

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*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426, Heidelberg, June 2006c. Springer.

Andreas Hotho, Dominik Benz, Robert Jäschke, and Beate Krause, editors. *ECML PKDD Discovery Challenge 2008 (RSDC'08)*. Workshop at 18th Europ. Conf. on Machine Learning (ECML'08) / 11th Europ. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'08), 2008. URL http://www.kde.cs.uni-kassel.de/ws/rsdc08/pdf/all_rsdc_v2.pdf.

D.I. Ignatov, R. Zhuk, and N. Konstantinova. Learning hypotheses from triadic labeled data. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*, volume 2, pages 474–480, Aug 2014. doi: 10.1109/WI-IAT.2014.136.

Ivan Ivanov, Peter Vajda, Jong-Seok Lee, and Touradj Ebrahimi. In tags we trust: Trust modeling in social tagging of multimedia content. *IEEE Signal Process. Mag.*, 29(2): 98–107, 2012. doi: 10.1109/MSP.2011.942345. URL http://dx.doi.org/10.1109/MSP.2011.942345.

Ivorix. Sine-weighted moving average, 2007. URL http://www.ivorix.com/en/products/tech/smooth/smooth.html.

Bernard J. Jansen. Search log analysis: What it is, what's been done, how to do it. *Library & Information Science Research*, 28(3):407–432, 2006. URL http://dx.doi.org/10.1016/j.lisr.2006.06.005.

Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Process. Manage.*, 36:207–227, January 2000. ISSN 0306-4573. doi: 10.1016/S0306-4573(99)00056-4. URL http://dl.acm.org/citation.cfm?id=342495.342498.

Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. Determining the informational, navigational, and transactional intent of web queries. *Inf. Process. Manage.*, 44:1251–1266, May 2008. ISSN 0306-4573. doi: 10.1016/j.ipm.2007.07.015. URL http://portal.acm.org/citation.cfm?id=1351187.1351372.

Sara Javanmardi, David W. McDonald, and Cristina V. Lopes. Vandalism detection in Wikipedia: a high-performing, feature-rich model and its reduction through Lasso. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, WikiSym '11, pages 82–90, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0909-7. doi: 10.1145/2038558.2038573. URL http://doi.acm.org/10.1145/2038558.2038573.

Wooseob Jeong. Is tagging effective?: overlapping ratios with other metadata fields. In *Proceedings of the 2009 International Conference on Dublin Core and Metadata Applications*, pages 31–39. Dublin Core Metadata Initiative, 2009. URL http://dl.acm.org/citation.cfm?id=1670638.1670643.

Karel Jezek and Jiri Hynek. The fight against spam - a machine learning approach. In Leslie Chan and Bob Martens, editors, *ELPUB*, pages 381–392, 2007. ISBN 978-3-85437-292-9. URL http://dblp.uni-trier.de/db/conf/elpub/elpub2007.html#JezekH07.

Jay J. Jiang and David W. Conrath. Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics (ROCLING)*, pages 19–33. Taiwan, 1997.

Yan'an Jin, Ruixuan Li, Yi Cai, Qing Li, Ali Daud, and Yuhua Li. Semantic grounding of hybridization for tag recommendation. In *Proceedings of the 11th international conference on Web-age information management*, WAIM'10, pages 139–150, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-14245-1, 978-3-642-14245-1. URL http://dl.acm.org/citation.cfm?id=1884017.1884037.

T. Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, 2002.

Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, London, UK, UK, 1998a. Springer-Verlag. ISBN 3-540-64417-2. URL http://dl.acm.org/citation.cfm?id=645326.649721.

Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, number 1398, pages 137–142, Chemnitz, DE, 1998b. Springer Verlag, Heidelberg, DE. URL /brokenurl#joachims98.ps.

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 154–161, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. doi: 10.1145/1076034.1076063. URL http://doi.acm.org/10.1145/1076034.1076063.

Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25, April 2007. ISSN 1046-8188. doi: 10.1145/1229179.1229181. URL http://doi.acm.org/10.1145/1229179.1229181.

Robert Jäschke. *Formal concept analysis and tag recommendations in collaborative tagging systems*. PhD thesis, Heidelberg, 2011. URL http://d-nb.info/1010227467/04. Zugl.: Kassel, Univ., Diss. 2010.

Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in folksonomies. In Joost Kok, Jacek Koronacki, Ramon Lopez de Mantaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron, editors, *Knowledge Discovery in Databases: PKDD 2007*, volume 4702 of *Lecture Notes in Computer Science*, pages 506–514. Springer, Berlin / Heidelberg, 2007. ISBN 978-3-540-74975-2. doi: 10.1007/978-3-540-74976-9_52. URL http://dx.doi.org/10.1007/978-3-540-74976-9_52.

Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in social bookmarking systems. *AI Communications*, 21(4):231–247, 2008. ISSN 0921-7126. doi: 10.3233/AIC-2008-0438. URL http://dx.doi.org/10.3233/AIC-2008-0438.

Robert Jäschke, Folke Eisterlehner, Andreas Hotho, and Gerd Stumme. Testing and evaluating tag recommenders in a live system. In *RecSys '09: Proceedings of the third ACM Conference on Recommender Systems*, pages 369–372, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-435-5. doi: 10.1145/1639714.1639790. URL http://www.kde.cs.uni-kassel.de/pub/pdf/jaeschke2009testing.pdf.

Jaap Kamps, Marijn Koolen, and Andrew Trotman. Comparative analysis of clicks and judgments for ir evaluation. In *Proceedings of the 2009 workshop on Web Search Click Data*, WSCD '09, pages 80–87, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-434-8. doi: 10.1145/1507509.1507522. URL http://doi.acm.org/10.1145/1507509.1507522.

Cushla Kapitzke and Bertram C. Bruce, editors. *Libr@ries : changing information space and practice*. Lawrence Erlbaum Associates, Mahwah, New Jersey, 2006. URL http://eprints.qut.edu.au/33025/.

Gabriella Kazai, Jaap Kamps, and Natasa Milic-Frayling. An analysis of human factors and label accuracy in crowdsourcing relevance judgments. *Information Retrieval*, 16(2):138–178, 2013. ISSN 1386-4564. doi: 10.1007/s10791-012-9205-0. URL http://dx.doi.org/10.1007/s10791-012-9205-0.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46:604–632, September 1999a. ISSN 0004-5411. doi: 10.1145/324133.324140. URL http://doi.acm.org/10.1145/324133.324140.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999b. URL http://citeseer.ist.psu.edu/kleinberg99authoritative.html.

Thomas Knerr. Tagging ontology - towards a common ontology for folksonomies, 2006. URL http://code.google.com/p/tagont/. http://tagont.googlecode.com/files/TagOntPaper.pdf.

Pranam Kolari, Tim Finin, and Anupam Joshi. SVMs for the Blogosphere: Blog Identification and Splog Detection. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*. Computer Science and Electrical Engineering, University of Maryland, Baltimore County, March 2006a. Also available as technical report TR-CS-05-13.

Pranam Kolari, Akshay Java, Tim Finin, Tim Oates, and Anupam Joshi. Detecting spam blogs: a machine learning approach. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, AAAI'06, pages 1351–1356. AAAI Press, 2006b. ISBN 978-1-57735-281-5. URL http://dl.acm.org/citation.cfm?id=1597348.1597403.

Santanu Kolay and Ali Dasdan. The value of socially tagged urls for a search engine. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1203–1204, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526929. URL http://doi.acm.org/10.1145/1526709.1526929.

Santanu Kolay and Ali Dasdan. The value of socially tagged urls for a search engine. In Juan Quemada, Gonzalo Leon, Yoelle S. Maarek, and Wolfgang Nejdl, editors, *WWW*, pages 1203–1204. ACM, 2009b. ISBN 978-1-60558-487-4. URL http://dblp.uni-trier.de/db/conf/www/www2009.html#KolayD09.

Christian Körner, Dominik Benz, Andreas Hotho, Markus Strohmaier, and Gerd Stumme. Stop thinking, start tagging: tag semantics emerge from collaborative verbosity. In *Proceedings of the 19th international conference on World wide web*, pages 521–530. ACM, 2010.

Eleni Kosta, Aleksandra Kuczerawy, Ronald Leenes, and Jos Dumortier. Regulating identity management. In *Digital Privacy - PRIME*, pages 73–89. 2011.

Georgia Koutrika, Frans Adjie Effendi, Zoltán Gyöngyi, Paul Heymann, and Hector Garcia-Molina. Combating spam in tagging systems. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb '07, pages 57–64, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-732-2. doi: 10.1145/1244408.1244420. URL http://doi.acm.org/10.1145/1244408.1244420.

Dominik Kowald, Emanuel Lacic, and Christoph Trattner. Tagrec: Towards a standardized tag recommender benchmarking framework. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, HT '14, pages 305–307, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2954-5. doi: 10.1145/2631775.2631781. URL http://doi.acm.org/10.1145/2631775.2631781.

Dominik Kowald, Paul Seitlinger, Simone Kopeinik, Tobias Ley, and Christoph Trattner. Forgetting the words but remembering the meaning: Modeling forgetting in a verbal and semantic tag recommender. In Martin Atzmueller, Alvin Chin, Christoph Scholz, and Christoph Trattner, editors, *Mining, Modeling, and Recommending 'Things' in Social Media*, Lecture Notes in Computer Science, pages 75–95. Springer International Publishing, 2015. ISBN 978-3-319-14722-2. doi: 10.1007/978-3-319-14723-9_5. URL http://dx.doi.org/10.1007/978-3-319-14723-9_5.

Beate Krause, Andreas Hotho, and Gerd Stumme. A comparison of social bookmarking with traditional search. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen W. White, editors, *30th European Conference on IR Research, ECIR 2008*, volume 4956 of *Lecture Notes in Computer Science*, pages 101–113, Glasgow, UK, April 2008a. Springer.

Beate Krause, Robert Jäschke, Andreas Hotho, and Gerd Stumme. Logsonomy - social information retrieval with logdata. In *HT '08: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 157–166, New York, NY, USA, 2008b. ACM. ISBN 978-1-59593-985-2. doi: 10.1145/1379092.1379123.

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, 2008c. ACM. ISBN 978-1-60558-159-0. doi: http://doi.acm.org/10.1145/1451983.1451998.

Beate Krause, Hana Lerch, Andreas Hotho, Alexander Roßnagel, and Gerd Stumme. Datenschutz im Web 2.0 am Beispiel des sozialen Tagging-Systems BibSonomy. *Informatik-Spektrum*, pages 1–12, 2010. ISSN 0170-6012. doi: 10.1007/s00287-010-0485-8. URL http://dx.doi.org/10.1007/s00287-010-0485-8.

Beate Krause, Hana Lerch, Andreas Hotho, Alexander Roßnagel, and Gerd Stumme. Datenschutz im web 2.0 am beispiel des sozialen tagging-systems bibsonomy. *Informatik Spektrum*, 35(1):12–23, 2012. URL http://dblp.uni-trier.de/db/journals/insk/insk35.html#KrauseLHRS12.

Balachander Krishnamurthy and Craig E. Wills. Characterizing privacy in online social networks. In *Proceedings of the first workshop on Online social networks*, WOSP '08, pages 37–42, New York, NY, USA, 2008. ACM. URL http://doi.acm.org/10.1145/1397735.1397744.

Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. A few chirps about twitter. In *Proceedings of the first workshop on Online social networks*, WOSN '08, pages 19–24, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-182-8. doi: 10.1145/1397735.1397741. URL http://doi.acm.org/10.1145/1397735.1397741.

Christian Körner, Dominik Benz, Markus Strohmaier, Andreas Hotho, and Gerd Stumme. Stop thinking, start tagging - tag semantics emerge from collaborative verbosity. In *Proceedings of the 19th International World Wide Web Conference (WWW 2010)*, Raleigh, NC, USA, April 2010a. ACM. URL http://www.kde.cs.uni-kassel.de/benz/papers/2010/koerner2010thinking.pdf.

Christian Körner, Roman Kern, Hans-Peter Grahsl, and Markus Strohmaier. Of categorizers and describers: an evaluation of quantitative measures for tagging motivation. In *HT '10: Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pages 157–166, New York, NY, USA, 2010b. ACM. ISBN 978-1-4503-0041-4. doi: 10.1145/1810617.1810645. URL http://portal.acm.org/citation.cfm?id=1810617.1810645.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772751. URL http://doi.acm.org/10.1145/1772690.1772751.

Frederick Wilfrid Lancaster. *Indexing and abstracting in theory and practice*. Univ. of Illinois, Graduate School of Library and Information Science, 2003.

S. le Cessie and J.C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.

Kang-Pyo Lee, Hong-Gee Kim, and Hyoung-Joo Kim. A social inverted index for social-tagging-based information retrieval. *J. Inf. Sci.*, 38(4):313–332, August 2012. ISSN 0165-5515. doi: 10.1177/0165551512438357. URL http://dx.doi.org/10.1177/0165551512438357.

Sangho Lee and Jong Kim. Warningbird: A near real-time detection system for suspicious urls in twitter stream. *IEEE Trans. Dependable Secur. Comput.*, 10(3):183–195, May 2013. ISSN 1545-5971. doi: 10.1109/TDSC.2013.3. URL http://dx.doi.org/10.1109/TDSC.2013.3.

Hana Lerch, Beate Krause, Andreas Hotho, Alexander Roßnagel, and Gerd Stumme. Social Bookmarking-Systeme – die unerkannten Datensammler – Ungewollte personenbezogene Datenverabeitung? *MultiMedia und Recht*, 7:454–458, 2010.

Hang Li. *Learning to Rank for Information Retrieval and Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2011a.

Hang Li. A short introduction to learning to rank. *IEICE Transactions*, 94-D(10):1854–1862, 2011b. URL http://dblp.uni-trier.de/db/journals/ieicet/ieicet94d.html#Li11.

Peng Li, Jian-Yun Nie, Bin Wang, and Jing He. Document re-ranking using partial social tagging. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on*, volume 1, pages 274–281, December 2012. doi: 10.1109/WI-IAT.2012.124. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6511896.

Ping Li, Christopher J. C. Burges, and Qiang Wu. Mcrank: Learning to rank using multiple classification and gradient boosting. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *NIPS*. MIT Press, 2007. URL http://dblp.uni-trier.de/db/conf/nips/nips2007.html#LiBW07.

Xin Li, Lei Guo, and Yihong Eric Zhao. Tag-based social interest discovery. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 675–684, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2. doi: 10.1145/1367497.1367589. URL http://doi.acm.org/10.1145/1367497.1367589.

Freddy Limpens, Fabien Gandon, and Michel Buffa. Helping online communities to semantically enrich folksonomies. *Web Science Conference 2010*, 2010.

Yu-Ru Lin, Hari Sundaram, Yun Chi, Junichi Tatemura, and Belle L. Tseng. Splog detection using self-similarity analysis on blog temporal dynamics. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, AIRWeb '07, pages 1–8, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-732-2. doi: 10.1145/1244408.1244410. URL http://doi.acm.org/10.1145/1244408.1244410.

Yuan Lin, Hongfei Lin, Song Jin, and Zheng Ye. Social annotation in query expansion: A machine learning approach. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 405–414, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2009972. URL http://doi.acm.org/10.1145/2009916.2009972.

G. Linden, B. Smith, and J. York. Amazon.com recommendations - item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

Marek Lipczak and Evangelos Milios. The impact of resource title on tags in collaborative tagging systems. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, HT '10, pages 179–188, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0041-4. doi: 10.1145/1810617.1810648. URL http://doi.acm.org/10.1145/1810617.1810648.

Kaipeng Liu, Binxing Fang, and Yu Zhang. Detecting tag spam in social tagging systems with collaborative knowledge. In *FSKD'09: Proceedings of the 6th international conference on Fuzzy systems and knowledge discovery*, pages 427–431, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-4545-5. URL http://portal.acm.org/citation.cfm?id=1802229&dl=GUIDE&coll=GUIDE&CFID=102532134&CFTOKEN=53193656.

Tie Y. Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *SIGIR '07: Proceedings of the Learning to Rank workshop in the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007. URL http://research.microsoft.com/~junxu/papers/SGIR2007-LR4IR%20workshop-LETOR.pdf.

Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3:225–331, March 2009. ISSN 1554-0669. doi: 10.1561/1500000016. URL http://portal.acm.org/citation.cfm?id=1618303.1618304.

Tie-Yan Liu. *Learning to Rank for Information Retrieval.* Springer, 2011. ISBN 978-3-642-14266-6.

Xin Liu and Tsuyoshi Murata. Detecting communities in k-partite k-uniform (hyper)networks. *J. Comput. Sci. Technol.*, 26(5):778–791, September 2011. ISSN 1000-9000. doi: 10.1007/s11390-011-0177-0. URL http://dx.doi.org/10.1007/s11390-011-0177-0.

Craig Macdonald and Iadh Ounis. Usefulness of quality click-through data for training. In *WSCD '09: Proceedings of the 2009 workshop on Web Search Click Data*, pages 75–79, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-434-8. doi: 10.1145/1507509.1507521.

Craig Macdonald, Rodrygo L. Santos, and Iadh Ounis. The whens and hows of learning to rank for web search. *Inf. Retr.*, 16(5):584–628, October 2013. ISSN 1386-4564. doi: 10.1007/s10791-012-9209-9. URL http://dx.doi.org/10.1007/s10791-012-9209-9.

Matteo Manca, Ludovico Boratto, and Salvatore Carta. Friend recommendation in a social bookmarking system: Design and architecture guidelines. In Kohei Arai, Supriya Kapoor, and Rahul Bhatia, editors, *Intelligent Systems in Science and Information 2014*, volume 591 of *Studies in Computational Intelligence*, pages 227–242. Springer International Publishing, 2015. ISBN 978-3-319-14653-9. doi: 10.1007/978-3-319-14654-6_14. URL http://dx.doi.org/10.1007/978-3-319-14654-6_14.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, USA, 2008.

Holger Stark Marcel Rosenbach. *Der NSA-Komplex: Edward Snowden und der Weg in die totale Überwachung*. Deutsche Verlags-Anstalt, München, 2014. ISBN 978-3-641-14150-9.

Leandro Balby Marinho, Alexandros Nanopoulos, Lars Schmidt-Thieme, Robert Jäschke, Andreas Hotho, Gerd Stumme, and Panagiotis Symeonidis. Social tagging recommender systems. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 615–644. Springer US, 2011. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3_19. URL http://dx.doi.org/10.1007/978-0-387-85820-3_19.

Benjamin Markines, Ciro Cattuto, and Filippo Menczer. Social spam detection. In Dennis Fetterly and Zoltän Gyöngyi, editors, *Proc. 5th International Workshop on Adversarial Information Retrieval on the Web AIRWeb*, ACM International Conference Proceeding Series, pages 41–48, 2009a. ISBN 978-1-60558-438-6. URL http://dblp.uni-trier.de/db/conf/airweb/airweb2009.html#MarkinesCM09.

Benjamin Markines, Ciro Cattuto, Filippo Menczer, Dominik Benz, Andreas Hotho, and Gerd Stumme. Evaluating similarity measures for emergent semantics of social tagging. In *18th International World Wide Web Conference*, pages 641–641, April 2009b. URL http://www.kde.cs.uni-kassel.de/pub/pdf/markines2009evaluating.pdf.

Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Position Paper, Tagging, Taxonomy, Flickr, Article, ToRead. In *Proceedings of the Collaborative Web Tagging Workshop at the WWW 2006*, Edinburgh, Scotland, May 2006. URL http://.rawsugar.com/www2006/cfp.html.

Murali Viswanathan Maureen Heymans. Introducing Google Social Search: I finally found my friend's New York blog!, 2009. URL http://googleblog.blogspot.de/2009/10/introducing-google-social-search-i.html. Accessed: 2014-02-17.

Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *Learning for Text Categorization: Papers from the 1998 AAAI Workshop*, pages 41–48. AAAI Press, 1998.

Soghra M.Gargari and Sule Gunduz Oguducu. A novel framework for spammer detection in social bookmarking systems. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, ASONAM '12,

pages 827–834, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4799-2. doi: 10.1109/ASONAM.2012.150. URL http://dx.doi.org/10.1109/ASONAM.2012.150.

Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *Proceedings of the Fourth International Semantic Web Conference (ISWC 2005)*, LNCS, pages 522–536. Springer, 2005. URL http://www.cs.vu.nl/~pmika/research/papers/ISWC-folksonomy.pdf.

Gilad Mishne, David Carmel, and Ronny Lempel. Blocking blog spam with language model disagreement. In *AIRWeb*, pages 1–6, 2005. URL http://dblp.uni-trier.de/db/conf/airweb/airweb2005.html#MishneCL05.

Thomas Mitchell. *Machine Learning*. McGraw-Hill Education (ISE Editions), October 1997. ISBN 0071154671. URL http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0071154671.

Folke Mitzlaff, Martin Atzmueller, Dominik Benz, Andreas Hotho, and Gerd Stumme. Community assessment using evidence networks. In Martin Atzmueller, Andreas Hotho, Markus Strohmaier, and Alvin Chin, editors, *Analysis of Social Media and Ubiquitous Data*, volume 6904 of *Lecture Notes in Computer Science*, pages 79–98. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23598-6. doi: 10.1007/978-3-642-23599-3_5. URL http://dx.doi.org/10.1007/978-3-642-23599-3_5.

Folke Mitzlaff, Martin Atzmueller, Andreas Hotho, and Gerd Stumme. The social distributional hypothesis: a pragmatic proxy for homophily in online social networks. *Social Network Analysis and Mining*, 4(1):216, 2014. ISSN 1869-5450. doi: 10.1007/s13278-014-0216-2. URL http://dx.doi.org/10.1007/s13278-014-0216-2.

Paola Monachesi and Thomas Markus. Using social media for ontology enrichment. In *Proceedings of the 7th International Conference on The Semantic Web: Research and Applications - Volume Part II*, ESWC'10, pages 166–180, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-13488-2, 978-3-642-13488-3. doi: 10.1007/978-3-642-13489-0_12. URL http://dx.doi.org/10.1007/978-3-642-13489-0_12.

Meredith Ringel Morris and Eric Horvitz. Searchtogether: an interface for collaborative web search. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 3–12, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-679-0. doi: 10.1145/1294211.1294215. URL http://doi.acm.org/10.1145/1294211.1294215.

P. Jason Morrison. Tagging and searching: Search retrieval effectiveness of folksonomies on the world wide web. *Inf. Process. Manage.*, 44:1562–1579, July 2008. ISSN 0306-4573. doi: 10.1016/j.ipm.2007.12.010. URL http://portal.acm.org/citation.cfm?id=1377061.1377474.

Vasanth Nair and Sumeet Dua. Folksonomy-based ad hoc community detection in online social networks. *Social Network Analysis and Mining*, 2(4):305–328, 2012. ISSN 1869-

5450. doi: 10.1007/s13278-012-0081-9. URL http://dx.doi.org/10.1007/s13278-012-0081-9.

Ramesh Nallapati. Discriminative models for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 64–71, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009006.

Beate Navarro Bullock, Robert Jäschke, and Andreas Hotho. Tagging data as implicit feedback for learning-to-rank. In *Proceedings of the ACM WebSci'11*, June 2011a. URL http://journal.webscience.org/463/.

Beate Navarro Bullock, Hana Lerch, Alexander Rossnagel, Andreas Hotho, and Gerd Stumme. Privacy-aware spam detection in social bookmarking systems. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '11, pages 15:1–15:8, New York, NY, USA, 2011b. ACM. ISBN 978-1-4503-0732-1. doi: 10.1145/2024288.2024306. URL http://doi.acm.org/10.1145/2024288.2024306.

Nicolas Neubauer and Klaus Obermayer. Hyperincident connected components of tagging networks. *SIGWEB Newsl.*, pages 4:1–4:10, September 2009. ISSN 1931-1745. doi: 10.1145/1592394.1592398. URL http://doi.acm.org/10.1145/1592394.1592398.

Nicolas Neubauer and Klaus Obermayer. Tripartite community structure in social bookmarking data. *New Rev. Hypermedia Multimedia*, 17(3):267–294, December 2011. ISSN 1361-4568. doi: 10.1080/13614568.2011.598952. URL http://dx.doi.org/10.1080/13614568.2011.598952.

Nicolas Neubauer, Robert Wetzker, and Klaus Obermayer. Tag spam creates large non-giant connected components. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '09, pages 49–52, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-438-6. doi: 10.1145/1531914.1531925. URL http://doi.acm.org/10.1145/1531914.1531925.

MEJ Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):323–351, 2005. doi: 10.1080/00107510500052444. URL http://www.tandfonline.com/doi/abs/10.1080/00107510500052444.

Casey Newton. WhatsApp now has over 400 million monthly users, 2013. URL http://www.theverge.com/2013/12/19/5228656/whatsapp-now-has-over-400-million-monthly-users. Accessed: 2014-02-16.

Satoshi Niwa, Takuo Doi, and Shinichi Honiden. Web page recommender system based on folksonomy mining for itng '06 submissions. In *Proceedings of the Third International Conference on Information Technology: New Generations*, pages 388–393, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2497-4. doi: 10.1109/ITNG.2006.140. URL http://dl.acm.org/citation.cfm?id=1128011.1128138.

Michael G. Noll. *Understanding and Leveraging the Social Web for Information Retrieval*. PhD thesis, Universität Potsdam, April 2010.

Michael G. Noll and Christoph Meinel. Exploring social annotations for web document classification. In *Proceedings of the 2008 ACM symposium on Applied computing*, SAC '08, pages 2315–2320, New York, NY, USA, 2008a. ACM. ISBN 978-1-59593-753-7. doi: 10.1145/1363686.1364235. URL http://doi.acm.org/10.1145/1363686.1364235.

Michael G. Noll and Christoph Meinel. The metadata triumvirate: Social annotations, anchor texts and search queries. In *Web Intelligence*, pages 640–647. IEEE, 2008b. URL http://dblp.uni-trier.de/db/conf/webi/webi2008.html#NollM08.

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 international ACM SIGIR conference on Research and development in information retrieval*, pages 612–619, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: http://doi.acm.org/10.1145/1571941.1572046. URL http://portal.acm.org/citation.cfm?id=1571941.1572046.

Noorda, C. Noorda, and S. Hanloser. *E-Discovery and Data Privacy: A Practical Guide*. Kluwer Law International, 2010. ISBN 9789041133458. URL http://books.google.de/books?id=o95_SqirU5wC.

OECD. Guidelines on the protection of privacy and transborder flows of personal data., 1980. http://www.oecd.org/document/18/0,2340,en_2649_34255_1815186_1_1_1_1,00.html.

Tim O'Reilly. What is web 2.0. design patterns and business models for the next generation of software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, September 2005. URL http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html. Stand 12.5.2009.

Simon Overell, Börkur Sigurbjörnsson, and Roelof van Zwol. Classifying tags using open content resources. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 64–73, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-390-7. doi: 10.1145/1498759.1498810. URL http://doi.acm.org/10.1145/1498759.1498810.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998. URL http://citeseer.ist.psu.edu/page98pagerank.html.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

Symeon Papadopoulos, Yiannis Kompatsiaris, and Athena Vakali. A graph-based clustering scheme for identifying related tags in folksonomies. In *Proceedings of the 12th international conference on Data warehousing and knowledge discovery*, DaWaK'10, pages 65–76, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15104-3, 978-3-642-15104-0. URL http://dl.acm.org/citation.cfm?id=1881923.1881931.

Symeon Papadopoulos, Athena Vakali, and Yiannis Kompatsiaris. Community detection in collaborative tagging systems. In Eric Pardede, editor, *Community-Built Databases*, pages 107–131. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-19046-9. doi: 10.1007/978-3-642-19047-6_5. URL http://dx.doi.org/10.1007/978-3-642-19047-6_5.

G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *Proc. 1st Intl. Conf. on Scalable Information Systems*, page 1. ACM Press New York, NY, USA, 2006.

Jing Peng, Daniel Dajun Zeng, Huimin Zhao, and Fei-yue Wang. Collaborative filtering in social tagging systems based on joint item-tag recommendations. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 809–818, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. doi: 10.1145/1871437.1871541. URL http://doi.acm.org/10.1145/1871437.1871541.

Martin Potthast, Benno Stein, and Robert Gerling. Automatic vandalism detection in wikipedia. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and RyenW. White, editors, *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 663–668. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78645-0. doi: 10.1007/978-3-540-78646-7_75. URL http://dx.doi.org/10.1007/978-3-540-78646-7_75.

Martin Potthast, Benno Stein, and Teresa Holfeld. Overview of the 1st International Competition on Wikipedia Vandalism Detection. In Martin Braschler, Donna Harman, and Emanuele Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*, 2010. ISBN 978-88-904810-0-0. URL http://dblp.uni-trier.de/db/conf/clef/clef2010w.html#PotthastSH10.

J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986. ISSN 0885-6125. doi: 10.1023/A:1022643204877. URL http://dx.doi.org/10.1023/A:1022643204877.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

Emanuele Quintarelli. Folksonomies: power to the people, June 2005. URL http://www-dimat.unipv.it/biblio/isko/doc/folksonomies.htm.

Steffen Rendle and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In Brian D. Davison, Torsten Suel, Nick Craswell, and Bing Liu, editors, *WSDM*, pages 81–90. ACM, 2010. ISBN 978-1-60558-889-6. URL http://dblp.uni-trier.de/db/conf/wsdm/wsdm2010.html#RendleS10.

Alexander Roßnagel. *Datenschutz in einem informatisierten Alltag, Studie für die Friedrich Ebert-Stiftung*. Studie für die Friedrich Ebert-Stiftung, Berlin 2007, 2007.

Giovanni Maria Sacco and Yannis Tzitzikas. *Dynamic Taxonomies and Faceted Search Theory, Practice, and Experience.* Springer, New York, 2009. ISBN 9783642023583 3642023584. URL http://www.worldcat.org/search?qt=worldcat_org_all&q=9783642023583.

Owen Sacco and Cécile Bothorel. Exploiting semantic web techniques for representing and utilising folksonomies. In *Proceedings of the International Workshop on Modeling Social Media*, MSM '10, pages 9:1–9:8, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0229-6. doi: 10.1145/1835980.1835989. URL http://doi.acm.org/10.1145/1835980.1835989.

Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

Yuta Sakakura, Toshiyuki Amagasa, and Hiroyuki Kitagawa. Detecting social bookmark spams using multiple user accounts. In *ASONAM*, pages 1153–1158. IEEE Computer Society, 2012. ISBN 978-0-7695-4799-2. URL http://dblp.uni-trier.de/db/conf/asunam/asonam2012.html#SakakuraAK12.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, August 1988. ISSN 0306-4573. doi: 10.1016/0306-4573(88)90021-0. URL http://dx.doi.org/10.1016/0306-4573(88)90021-0.

Rossano Schifanella, Alain Barrat, Ciro Cattuto, Benjamin Markines, and Filippo Menczer. Folks in folksonomies: Social link prediction from shared metadata. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 271–280, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-889-6. doi: 10.1145/1718487.1718521. URL http://doi.acm.org/10.1145/1718487.1718521.

Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme. Mining association rules in folksonomies. In V. Batagelj, H.-H. Bock, A. Ferligoj, and A. Žiberna, editors, *Data Science and Classification. Proceedings of the 10th IFCS Conf.*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–270, Heidelberg, July 2006. Springer.

Patrick Schmitz. Inducing ontology from flickr tags. In *Collaborative Web Tagging Workshop at WWW 2006*, Edinburgh, Scotland, May 2006.

Johann Schrammel, Christina Köffel, and Manfred Tscheligi. How much do you tell?: information disclosure behaviour indifferent types of online communities. In *Proceedings of the fourth international conference on Communities and technologies*, C&T '09, pages 275–284, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-713-4. doi: 10.1145/1556460.1556500. URL http://doi.acm.org/10.1145/1556460.1556500.

Johann Schrammel, Christina Köffel, and Manfred Tscheligi. Personality traits, usage patterns and information disclosure in online communities. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, BCS-HCI '09, pages 169–174, Swinton, UK, UK, 2009b. British Computer Society. URL http://portal.acm.org/citation.cfm?id=1671011.1671031.

D. Sculley. *Advances in Online Learning-based Spam Filtering*. PhD thesis, Tufts University, Boston, MA, August 2008. URL http://www.ceas.cc/2007/papers/paper-61.pdf.

David Sculley and Gabriel M. Wachman. Relaxed online svms for spam filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 415–422, New York, NY, USA, 2007. ACM. URL http://doi.acm.org/10.1145/1277741.1277813.

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 2006 20th anniversary conference on Computer supported cooperative work*, pages 181–190, New York, NY, USA, 2006. ACM. ISBN 1-59593-249-6. doi: 10.1145/1180875.1180904. URL http://portal.acm.org/citation.cfm?id=1180904.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL http://axon.cs.byu.edu/~martinez/classes/778/Papers/settles.activelearning.pdf.

Clay Shirky. Ontology is overrated: Categories, links, and tags, 2005. URL http://www.shirky.com/writings/ontology_overrated.html.

Alexander B. Sideridis and Charalampos Z. Patrikakis, editors. *Next Generation Society. Technological and Legal Issues - Third International Conference, e-Democracy 2009, Athens, Greece, September 23-25, 2009, Revised Selected Papers*, volume 26 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 2010. Springer. ISBN 978-3-642-11629-2.

Seam Silcoff. Upstart Canadian chat service Kik logs 100 million users, 2013. URL http://www.theglobeandmail.com/technology/tech-news/upstart-canadian-chat-service-kik-logs-100-million-users/article15940135/. Accessed: 2014-02-16.

Fabrizio Silvestri. Mining query logs: Turning search usage data into knowledge. *Found. Trends Inf. Retr.*, 4:1–174, January 2010. ISSN 1554-0669. doi: 10.1561/1500000013. URL http://dx.doi.org/10.1561/1500000013.

Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24 (4):35–43, 2001. URL http://dblp.uni-trier.de/db/journals/debu/debu24.html#Singhal01.

K. Smets, B. Goethals, and B. Verdonk. Automatic vandalism detection in wikipedia: Towards a machine learning approach. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy (WikiAI08)*, pages 43–48. AAAI Press, 2008.

Sören Sonnenburg, Gunnar Rätsch, and Konrad Rieck. Large scale learning with string kernels. In Léon Bottou, Olivier Chapelle, Dennis DeCoste, and Jason Weston, editors, *Large Scale Kernel Machines*, pages 73–103. MIT Press, Cambridge, MA., 2007.

Lucia Specia and Enrico Motta. Integrating folksonomies with the semantic web. In *Proceedings of the 4th European conference on The Semantic Web: Research and Applications*, ESWC '07, pages 624–639, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-72666-1. doi: 10.1007/978-3-540-72667-8_44. URL http://dx.doi.org/10.1007/978-3-540-72667-8_44.

Louise Spiteri. Structure and form of folksonomy tags: The road to the public library catalogue. *Webology*, 4(2), 2007.

Tobias Stadler. Schutz vor Spam durch Greylisting - Eine rechtsadäquate Handlungsoption? *Datenschutz und Datensicherheit*, 6:433–438, 2005.

Julia Stoyanovich, Sihem Amer-Yahia, Cameron Marlow, and Cong Yu. Leveraging tagging to model user interests in del.icio.us. In *AAAI'08: Proceedings of the 2008 AAAI Social Information Spring Symposium*, 2008.

Markus Strohmaier, Denis Helic, Dominik Benz, Christian Körner, and Roman Kern. Evaluation of folksonomy induction algorithms. *ACM Trans. Intell. Syst. Technol.*, 3(4): 74:1–74:22, September 2012. ISSN 2157-6904. doi: 10.1145/2337542.2337559. URL http://doi.acm.org/10.1145/2337542.2337559.

Frederic Stutzman. An evaluation of identity-sharing behavior in social network communities. *iDMAa Journal*, 3(1), 2006. URL http://www.ibiblio.org/fred/pubs/stutzman_pub4.pdf.

Dany Sullivan. Msn search gets neural netranknet technology & (potentially) awesome new search commands. http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html, 2005. URL http://searchenginewatch.com/article/2061776/. Accessed: 2014-02-16.

Kyoung-Jun Sung, Soo-Cheol Kim, and Sung Kwon Kim. Tag quantification for spam detection in social bookmarking system. In *Advanced Information Management and Service (IMS), 2010 6th International Conference on*, pages 297–303, Nov 2010.

Panagiotis Symeonidis. User recommendations based on tensor dimensionality reduction. In Lazaros S. Iliadis, Ilias Maglogiannis, Grigorios Tsoumakas, Ioannis P. Vlahavas, and Max Bramer, editors, *AIAI*, volume 296 of *IFIP Advances in Information and Communication Technology*, pages 331–340. Springer, 2009. ISBN 978-1-4419-0220-7. URL http://dblp.uni-trier.de/db/conf/ifip12/aiai2009.html#Symeonidis09.

Monika Taddicken. The 'privacy paradox' in the social web: The impact of privacy concerns, individual characteristics, and the perceived social relevance on different forms of self-disclosure. *Journal of Computer-Mediated Communication*, 19(2):248–273, 2014. ISSN 1083-6101. doi: 10.1111/jcc4.12052. URL http://dx.doi.org/10.1111/jcc4.12052.

Kurt Thomas, Chris Grier, Dawn Song, and Vern Paxson. Suspended accounts in retrospect: An analysis of twitter spam. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 243–258, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1013-0. doi: 10.1145/2068816.2068840. URL http://doi.acm.org/10.1145/2068816.2068840.

Kurt Thomas, Frank Li, Chris Grier, and Vern Paxson. Consequences of connectivity: Characterizing account hijacking on twitter. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 489–500, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2957-6. doi: 10.1145/2660267.2660282. URL http://doi.acm.org/10.1145/2660267.2660282.

Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 319–326, New York, NY, USA, 2004. ACM. ISBN 1-58113-702-8. doi: 10.1145/985692.985733. URL http://doi.acm.org/10.1145/985692.985733.

Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 315–323, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: 10.1145/290941.291017. URL http://doi.acm.org/10.1145/290941.291017.

Thomas Vander Wal. Explaining and showing broad and narrow folksonomies. Blog post, February 2005. URL http://www.personalinfocloud.com/2005/02/explaining_and_.html.

A.H. Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10. IEEE, 2010.

Xufei Wang, Lei Tang, Huiji Gao, and Huan Liu. Discovering overlapping groups in social media. In Geoffrey I. Webb, Bing Liu 0001, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu, editors, *ICDM*, pages 569–578. IEEE Computer Society, 2010. URL http://dblp.uni-trier.de/db/conf/icdm/icdm2010.html#WangTGL10.

Christian Wartena. Automatic classification of social tags. In Mounia Lalmas, Joemon Jose, Andreas Rauber, Fabrizio Sebastiani, and Ingo Frommholz, editors, *Research and Advanced Technology for Digital Libraries*, volume 6273 of *Lecture Notes in Computer Science*, pages 176–183. Springer, Berlin / Heidelberg, 2010. ISBN 978-3-642-15463-8. doi: 10.1007/978-3-642-15464-5_19. URL http://dx.doi.org/10.1007/978-3-642-15464-5_19.

Duncan J. Watts and Steven Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, June 1998.

Andrew G. West, Jian Chang, Krishna Venkatasubramanian, Oleg Sokolsky, and Insup Lee. Link spamming wikipedia for profit. In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, CEAS '11, pages 152–161, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0788-8. doi: 10.1145/2030376.2030394. URL http://doi.acm.org/10.1145/2030376.2030394.

Alan F. Westin. *Privacy and freedom*. Atheneum, New York, 1970. URL http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+267571232&sourceid=fbw_bibsonomy.

Robert Wetzker, Carsten Zimmermann, and Christian Bauckhage. Analyzing social bookmarking systems: A del.icio.us cookbook. In *Mining Social Data (MSoDa) Workshop Proceedings*, pages 26–30. ECAI 2008, July 2008.

Robert Wetzker, Winfried Umbrath, and Alan Said. A hybrid approach to item recommendation in folksonomies. In *Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in Information Retrieval*, ESAIR '09, pages 25–29, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-430-0. doi: 10.1145/1506250.1506255. URL http://doi.acm.org/10.1145/1506250.1506255.

Willinge, Alderson, and Doyle. Mathematics and the internet: A source of enormous confusion and great potential. Technical Report 5, May 2009. URL /brokenurl#.

Microsoft Windows. Messenger has moved to Skype, 2013. URL http://windows.microsoft.com/en-us/messenger/messenger-to-skype. Accessed: 2014-05-04.

Harris Wu, Mohammad Zubair, and Kurt Maly. Harvesting social knowledge from folksonomies. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, HYPERTEXT '06, pages 111–114, New York, NY, USA, 2006a. ACM. ISBN 1-59593-417-0. doi: 10.1145/1149941.1149962. URL http://doi.acm.org/10.1145/1149941.1149962.

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, 2006b. ACM Press. URL http://doi.acm.org/10.1145/1135777.1135839.

Jingfang Xu, Chuanliang Chen, Gu Xu, Hang Li, and Elbio Renato Torres Abib. Improving quality of training data for learning to rank using click-through data. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 171–180, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-889-6. doi: 10.1145/1718487.1718509.

Hsin-Chang Yang and Chung-Hong Lee. Post-level spam detection for social bookmarking web sites. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 180 –185, july

2011a. doi: 10.1109/ASONAM.2011.81. URL http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5992578&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F5992553%2F5992565%2F05992578.pdf%3Farnumber%3D5992578.

Hsin-Chang Yang and Chung-Hong Lee. Post-level spam detection for social bookmarking web sites. In *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '11, pages 180–185, Washington, DC, USA, 2011b. IEEE Computer Society. ISBN 978-0-7695-4375-8. doi: 10.1109/ASONAM.2011.81. URL http://dx.doi.org/10.1109/ASONAM.2011.81.

Yuhao Yang, Jonathan Lutes, Fengjun Li, Bo Luo, and Peng Liu. Stalking online: On user privacy in social networks. In *Proceedings of the Second ACM Conference on Data and Application Security and Privacy*, CODASPY '12, pages 37–48, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1091-8. doi: 10.1145/2133601.2133607. URL http://doi.acm.org/10.1145/2133601.2133607.

Sasan Yazdani, Ivan Ivanov, Morteza AnaLoui, Reza Berangi, and Touradj Ebrahimi. Spam fighting in social tagging systems. In Karl Aberer, Andreas Flache, Wander Jager, Ling Liu, Jie Tang, and Christophe Guéret, editors, *Social Informatics*, volume 7710 of *Lecture Notes in Computer Science*, pages 448–461. Springer Berlin Heidelberg, 2012a. ISBN 978-3-642-35385-7. doi: 10.1007/978-3-642-35386-4_33. URL http://dx.doi.org/10.1007/978-3-642-35386-4_33.

Sasan Yazdani, Ivan Ivanov, Morteza AnaLoui, Reza Berangi, and Touradj Ebrahimi. Spam fighting in social tagging systems. In Karl Aberer, Andreas Flache, Wander Jager, Ling Liu, Jie Tang, and Christophe Guéret, editors, *SocInfo*, volume 7710 of *Lecture Notes in Computer Science*, pages 448–461. Springer, 2012b. ISBN 978-3-642-35385-7. URL http://dblp.uni-trier.de/db/conf/socinfo/socinfo2012.html#YazdaniIABE12.

Ching Man Au Yeung. *From User Behaviours to Collective Semantics*. PhD thesis, University of Southampton, 2009.

Dawei Yin, Liangjie Hong, and Brian D. Davison. Exploiting session-like behaviors in tag prediction. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 167–168, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0637-9. doi: 10.1145/1963192.1963277. URL http://doi.acm.org/10.1145/1963192.1963277.

Takayuki Yoshinaka, Soichi Ishii, Tomohiro Fukuhara, Hidetaka Masuda, and Hiroshi Nakagawa. A user-oriented splog filtering based on a machine learning. In *Proceedings of the 2008/2009 international conference on Social software: recent trends and developments in social software*, BlogTalk'08/09, pages 88–99, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-16580-X, 978-3-642-16580-1. URL http://dl.acm.org/citation.cfm?id=1929285.1929294.

Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*,

SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6. doi: 10.1145/383952.384019. URL http://doi.acm.org/10.1145/383952.384019.

Zi-Ke Zhang, Tao Zhou, and Yi-Cheng Zhang. Tag-aware recommender systems: A state-of-the-art survey. *Journal of Computer Science and Technology*, 26(5):767–777, 2011. ISSN 1000-9000. doi: 10.1007/s11390-011-0176-1. URL http://dx.doi.org/10.1007/s11390-011-0176-1.

Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explor. Newsl.*, 10:12–22, December 2008a. ISSN 1931-0145. doi: 10.1145/1540276.1540279. URL http://doi.acm.org/10.1145/1540276.1540279.

Dong Zhou, Séamus Lawless, and Vincent Wade. Improving search via personalized query expansion using social media. *Inf. Retr.*, 15(3-4):218–242, June 2012. ISSN 1386-4564. doi: 10.1007/s10791-012-9191-2. URL http://dx.doi.org/10.1007/s10791-012-9191-2.

Mianwei Zhou, Shenghua Bao, Xian Wu, and Yong Yu. An unsupervised model for exploring hierarchical semantics from social annotations. pages 680–693, 2008b. URL http://dx.doi.org/10.1007/978-3-540-76298-0_49.

Tom Chao Zhou, Hao Ma, Michael R. Lyu, and Irwin King. Userrec: A user recommendation framework in social tagging systems. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010.

Linhong Zhu, Aixin Sun, and Byron Choi. Detecting spam blogs from blog search results. *Inf. Process. Manage.*, 47(2):246–262, March 2011. ISSN 0306-4573. doi: 10.1016/j.ipm.2010.03.006. URL http://dx.doi.org/10.1016/j.ipm.2010.03.006.

Arkaitz Zubiaga, Raquel Martinez, and Victor Fresno. Getting the most out of social annotations for web page classification. In Uwe M. Borghoff and Boris Chidlovskii, editors, *ACM Symposium on Document Engineering*, pages 74–83. ACM, 2009. ISBN 978-1-60558-575-8. URL http://dblp.uni-trier.de/db/conf/doceng/doceng2009.html#ZubiagaMF09.

Appendix A

Appendix

Table A.1: Features used for computing rankings described in Section 6.5.3. C is the document collection, q the query term or tag, d represents a part of the document, which can be a title, summary, URL or the concatenation of all three. The document frequency $df(q_i)$ is the number of documents containing q_i. $count(q_i, d)$ counts the number of occurrences of q_i in document d.

Feature	Description	Formula / References
tftitle tfsummary tfurltok tfall	term frequency (tf) of title term frequency (tf) of summary term frequency (tf) of URL term frequency (tf) of all text available	$tf(q,d) = \sum_{q_i \in q \cap d} count(q_i, d)$
idfsummary idftitle idfurltok idfall	idf of summary idf of title idf of URL idf of all text available	$idf = \sum_{q_i \in q \cap d} log(\frac{\|C\| - df(q_i) + 0.5}{df(q_i) + 0.5})$
tf-idfsummary tf-idftitle tf-idfurltok tf-idfall	tf*idf of summary tf*idf of title tf*idf of URL tf*idf of all text available	
bm25urltok bm25title bm25summary bm25all	BM25 of URL BM25 of title BM25 of summary BM25 of all	$bm25 = idf * \frac{tf*(k_1+1)}{(k_1*(1-b+b*(\frac{docLength}{averageLength}))+tf))}$
lmirjmtitle lmirjmsummary lmirjmurltok lmirjmall	Language model with Jelinek-Mercer smoothing	Zhai and Lafferty [2001]
lmirdirtitle lmirdirsummary lmirdirurltok lmirdirall	Language model with Bayesian smoothing using Dirichlet priors	Zhai and Lafferty [2001]
lmirabstitle lmirabssummary lmirabsurltok lmirabsall	Language model with absolute discounting smoothing	Zhai and Lafferty [2001]
queryinhost queryishost pathexists pathlength fileexists filelength urllength queryinurl urltilde titlelength homeintitle homeinsummary homeinurltok homeinall dlsummary dltitle dlurl dlall	query is contained in host query is the host number of "/" in the path length of the path true if file is contained in URL length of the file length of the URL number of times query is in url tilde is contained in URL length of the document title title contains keyword *home* title contains keyword *home* title contains keyword *home* title contains keyword *home* document length of summary document length of title document length of URL document length of all	